

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL



Facultad de Ingeniería en Electricidad y Computación

“MEJORAMIENTO DEL PROCESO DE APROVISIONAMIENTO, CONFIGURACIÓN
Y ENTREGA DE SERVICIOS DE HOSTING DE UNA EMPRESA DE
TELECOMUNICACIONES USANDO BPMN”

TRABAJO DE TITULACIÓN

PREVIA A LA OBTENCIÓN DEL TÍTULO DE:

MAGISTER EN SISTEMAS DE INFORMACIÓN GERENCIAL

Presentado por:

ALLAN ROBERTO SUÁREZ CARVAJAL

GUAYAQUIL - ECUADOR

AÑO 2023

AGRADECIMIENTO

Agradezco a mi familia por el apoyo y la paciencia, a mi empresa por la oportunidad de crecimiento que me brindan y sobre todo a Dios, sin el, ningún esfuerzo es posible y ningún sacrificio tiene sentido.

DEDICATORIA

Dedico este trabajo a mi familia, a mi hija
y sobre todo a Dios.

TRIBUNAL DE SUSTENTACIÓN

M.Sc. Juan Carlos García Plúa

TUTOR

M.Sc. Lenin Eduardo Freire Cobo

REVISOR

DECLARACIÓN EXPRESA

“La responsabilidad del contenido de esta Tesis de Grado, me corresponde exclusivamente, y el patrimonio intelectual de la misma a la ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL.”

(Reglamento de graduación de la ESPOL)

Ing. Allan Roberto Suárez Carvajal

RESUMEN

El poder realizar una propuesta de mejora enfocado en la automatización del aprovisionamiento de servicios de Hosting, específicamente, en la gestión de máquinas virtuales, fue el enfoque de este trabajo, en donde, como primera instancia se debió analizar aquellos puntos de dolor que terminan causando inconformidad respecto a la calidad de servicios de cara a los clientes. Este problema básicamente se deriva de un trabajo manual el cual, muchas veces resulta de un proceso bastante democrático con tiempos de espera y gestión elevados acorde a como las áreas involucradas vayan realizando sus tareas en contraste del “mal” uso de la tecnología, esto, al no aprovechar todas las capacidades que las herramientas usadas por la operación pueden proveer. Esto puede darse por la falta de conocimiento o planificación para llevar a cabo un esquema de transformación digital progresiva.

Para poder cumplir con el objetivo, se realizó el levantamiento del proceso actual para que, a partir de ahí, podamos ser capaces de entender el estatus e identificar dónde atacar el problema, esta información pudo ser obtenida analizando del cómo opera el sistema en este flujo, en contraste con los modelos provistos por el área de Procesos de la compañía, el modelo se levanto utilizando el Modeler BPMN de Camunda.

Una vez levantado el proceso y los puntos de dolor, se analizó la data recopilada en el tiempo que tiene operando este flujo en el sistema, con esto, se pudo identificar además, los indicadores (en función del tiempo) que permitieron explicar de manera

cuantitativa el problema y sobre todo sirvió como guía para la propuesta de mejora planteada.

Con la información del modelo y los indicadores, se pudo establecer luego, un modelo BPM mejorado, en donde, el enfoque principal básicamente estaba orientado en la integración del sistema con las herramientas o APIs disponibles por aquellas plataformas, en donde manualmente se venían realizando estas tareas de aprovisionamiento. Para llegar a este objetivo fue importante la revisión de literatura relacionada a aspectos de virtualización, integración, APIs y arquitectura orientada a microservicios, que al final, sobre esta última, se implementaron mediante la prueba de concepto, los procesos y servicios automáticos que serían ejecutados por el task manager de Camunda, pudiendo así, simular la interacción modelo, microservicio y plataforma y en esta última, con un mockup de las definiciones del API provista a partir de la documentación.

Como resultado final de esta propuesta de implementación, se puede anticipar el cómo el proceso de mejora automatizado, reduciría considerablemente aquellos tiempos elevados y poco productivos en contraste con lo que una infraestructura de alta disponibilidad puede respondernos en términos de milisegundos, al ejecutar cada definición de las APIs que reemplace aquellas tareas manuales de formas más eficiente.

ÍNDICE GENERAL

AGRADECIMIENTO	II
DEDICATORIA.....	III
TRIBUNAL DE SUSTENTACIÓN	IV
DECLARACIÓN EXPRESA.....	V
RESUMEN	VI
ÍNDICE GENERAL.....	VIII
ÍNDICE DE FIGURAS.....	XI
ÍNDICE DE TABLAS	XIV
INTRODUCCIÓN	XV
CAPÍTULO 1	1
GENERALIDADES.....	1
1.1 Antecedentes	1
1.2 Descripción del problema	2
1.3 Solución propuesta	3
1.4 Objetivo general.....	5
1.5 Objetivos específicos	5
1.6 Metodología.....	6

CAPÍTULO 2	9
MARCO TEÓRICO	9
2.1 Modelamiento BPMN.....	10
2.1.1 Modelado de Procesos.....	10
2.1.2 Notación BPMN	11
2.1.3 Mapeo de procesos AS-IS/TO-BE.....	14
2.2 Camunda como plataforma BPMN.....	17
2.2.1 Motivación.....	17
2.2.2 Arquitectura	18
2.3 Arquitectura orientada a microservicios	24
2.3.1 Descripción general.....	24
2.3.2 Orquestación de Microservicios.....	27
2.3.3 Comunicación API REST	29
2.4 Virtualización	31
2.4.1 Terminología.....	31
2.4.2 Cloud Computing.....	37
2.5 VMware Cloud Directory	40
2.5.1 Overview	40
2.5.2 Arquitectura vCloud Director.....	45
2.5.3 Integración vía vCloud Directory APIs	47
CAPÍTULO 3	52

LEVANTAMIENTO Y ANÁLISIS DEL PROCESO DE HOSTING ACTUAL	52
3.1 Características de requerimientos de Hosting	52
3.2 Etapas de implementación de servicios de Hosting	57
3.2.1 Matriz RACI	58
3.2.2 Análisis de las etapas de implementación.....	60
3.2.3 Análisis de Tiempos de gestión de las etapas	62
3.3 Modelo AS-IS del proceso actual	67
CAPÍTULO 4	70
DISEÑO DE LA SOLUCIÓN	70
4.1 Modelo TO-BE para el flujo de aprovisionamiento y configuración	71
4.2 Diseño arquitectónico de componentes de las solución.....	74
4.3 Diagramas de secuencias de la solución	77
CAPÍTULO 5	84
ANÁLISIS DE RESULTADOS	84
5.1 Prueba de concepto.....	84
5.2 Análisis de resultados	111
CONCLUSIONES Y RECOMENDACIONES.....	113
BIBLIOGRAFÍA	115

ÍNDICE DE FIGURAS

Imagen 2.1 : Modelado de Procesos	11
Imagen 2.2 : Elementos básicos de notación BPMN	12
Imagen 2.3 : Mapeo de procesos	16
Imagen 2.4 : Ciclo de vida de los procesos	17
Imagen 2.5 : Overview de la arquitectura de Camunda	19
Imagen 2.6 : Arquitectura del Engine de Camunda.....	20
Imagen 2.7 : Camunda Modeler Interface.....	21
Imagen 2.8 : Camunda Cockpit	22
Imagen 2.9 : Camunda Tasklist	23
Imagen 2.10 : Tipos de arquitectura de aplicaciones – Arquitectura monolítica	26
Imagen 2.11 : Tipos de arquitectura de aplicaciones – Arquitectura de microservicios	26
Imagen 2.12 : Arquitectura de un clúster de contenedores.....	28
Imagen 2.13 : Hypervisor sobre Hardware	32
Imagen 2.14 : Arquitectura hospedada	33
Imagen 2.15 : Arquitectura nativa.....	33
Imagen 2.16 : vCloud Director - Arquitectura general.....	46
Imagen 2.17 : Operaciones básicas vCloud API	49
Imagen 3.1 : Segmentación de recursos del POOL	57
Imagen 3.2 : Matriz RACI flujo de Cloud IAAS.....	59
Imagen 3.3 : Gráfico de barras de tiempos de flujo por años	65

Imagen 3.4 : Modelo AS-IS Flujo Aprovisionamiento.....	68
Imagen 4.1 : Modelo TO-BE de la Solución.....	73
Imagen 4.2 : Diagrama de componentes de Software.....	76
Imagen 4.3 : Vista general de casos de usos y actores	78
Imagen 4.4 : Diagrama de secuencia general.....	79
Imagen 5.1 : Definición del Mock con HOST virtual	85
Imagen 5.2 : Folder Request.....	86
Imagen 5.3 : Definición de folder response en base a documentación	86
Imagen 5.4 : Cluster API Request.....	87
Imagen 5.5 : Definición de cluster response en base a documentación.....	87
Imagen 5.6 : Datastore API Request	88
Imagen 5.7 : Definición de datastore response en base a documentación.....	88
Imagen 5.8 : Resource Pool API Request	89
Imagen 5.9 : Definición de Resource-pool response en base a documentación	89
Imagen 5.10 : Host API Request	90
Imagen 5.11 : Definición de Host response en base a documentación	90
Imagen 5.12 : Máquina Virtual API Request.....	91
Imagen 5.13 : Definición de MV response en base a documentación	91
Imagen 5.14 : Máquina Virtual Relocate API Request.....	92
Imagen 5.15 : Estructura del microservicio de aprovisionamiento.....	93
Imagen 5.16 : Configuraciones del servicio	95
Imagen 5.17 : Segmento del Modelo TO-BE automatizado.....	96

Imagen 5.18 : Inicialización del Microservicio de aprovisionamiento	96
Imagen 5.19 : Login plataforma de Camunda	97
Imagen 5.20: Home page de Camunda	97
Imagen 5.21: Formulario de requerimientos técnicos	98
Imagen 5.22: Despliegue del formulario sobre Camunda	99
Imagen 5.23: Despliegue del modelo TO-BE sobre Camunda	99
Imagen 5.24 : Proceso cargado a la plataforma	100
Imagen 5.25: Proceso con sus valores de definción.....	100
Imagen 5.26: Rest API para inicializar flujo el flujo del modelo.....	101
Imagen 5.27 : Ejecución del API Simulador de vmware (datastore)	102
Imagen 5.28: Inicio del flujo BPMN via Camunda API REST	103
Imagen 5.29 : Inicio de proceso BPMN de aprovisionamiento	104
Imagen 5.30 : Ejecución del Task - Ingresar anexo técnico	104
Imagen 5.31 : Ingreso de valores en formulario para el Task de anexo técnico	105
Imagen 5.32: Ejecución de flujo manual de Solicitar Factibilidad técnica	106
Imagen 5.33: Proceso de validación de capacidades dentro del flujo	107
Imagen 5.34 : Validación de capacidad de disco.....	107
Imagen 5.35 : Validación de capacidades de memoria y CPU	108
Imagen 5.36 : Proceso de creación de máquinas virtuales	108
Imagen 5.37: Creación de máquina virtual	109
Imagen 5.38: Máquina Virtual con información para creación.....	109
Imagen 5.39: User Task para planificación de entrega	110

ÍNDICE DE TABLAS

Tabla 1 : Variables de medición.....	7
Tabla 2 : Ejemplos de Métodos HTTP	30
Tabla 3 : Definición de API básica provista por vSphere	50
Tabla 4 : Solución de servicios para Cliente ABC	54
Tabla 5 : Asignación de Recursos al POOL	56
Tabla 6 : Roles o actores del flujo	58
Tabla 7 : Actividades para implementación de Servicio de Hosting	60
Tabla 8 : Definición de la descripción de intervalos de tiempos de análisis.....	64
Tabla 9 : Tabulación del tiempo promedio en minutos de cada etapa (2018-2022)...	64
Tabla 10 : Consolidado (5 años) de tiempo promedio en minutos y días.....	66
Tabla 11 : Detalle de Actividades para el proceso de mejora.....	71
Tabla 12 : Componentes de Software de la solución.....	75
Tabla 13 : CU001 - Generación de Anexo Técnico.....	80
Tabla 14 : CU002 - Generación de Orden de Servicio	81
Tabla 15 : CU003 - Solicitar Factibilidad Técnica	81
Tabla 16 : CU004 - Validar disponibilidad y capacidades	82
Tabla 17 : CU005 - Crear máquinas virtuales	82
Tabla 18 : CU006 - Planificar entrega y activación de servicio	83

INTRODUCCIÓN

En la actualidad, poder abarcar más mercado que permitan generar mayor ingreso a las empresas, termina siendo una de las principales estrategias a aplicar. Sin embargo, muchas veces, aunque se haya encontrado la forma en la que esto funcione, puede darse el caso en el cual, las empresas no se encuentran alineadas internamente de una manera eficiente, sea esto, que las áreas no trabajen de manera colaborativa y fluida o que los sistemas de información estén creados de tal manera que necesiten en algún punto interacción humana incluso en aquellos procesos que pueden ser realizados de forma más rápida y eficiente a partir de una automatización.

La empresa de análisis de esta tesis, tiene algunas líneas de negocio enfocadas en servicios de Internet, Datos y de Centro de datos, sobre todo, en este último, es en donde se enfoca toda la estrategia de mejora en aquellos flujos ligados a sus productos core, por ejemplo, los servicios de Hosting, que como se puede entender, tiene mucha relación con plataformas de virtualización, sea, en la nube o centros de datos propios.

La estrategia más adecuada siempre será saber integrar los sistemas internos con las herramientas que sirven a la operación, esto es, entender como camina día a día la tecnología, los procesos automáticos e integraciones entre ellos y así poder reducir trabajo operativo y sobre todo costos.

CAPÍTULO 1

GENERALIDADES

En el siguiente capítulo se darán a conocer los antecedentes de la empresa y los problemas que actualmente se enfrenta en uno de sus procesos de aprovisionamiento, así como el planteamiento de la solución y la metodología a ser usada. Además, se define tanto el objetivo general como específico del alcance de este proyecto.

1.1 Antecedentes

Dentro de una empresa de Telecomunicaciones cuyo Core de negocio está enfocado en entregar servicios de Internet, Datos y de Centro de Datos, al momento, el flujo de ingreso de órdenes de servicio, aprovisionamiento y entrega de servicios de HOSTING específicamente aquellos referentes a la entrega de Máquinas Virtuales, resulta un proceso bastante burocrático y

demorado lleno de pasos lo cual se evidencia por los tiempos elevados del mismo, lo cual trae inconformidad de parte del cliente en la percepción de nivel de Servicio que tiene de la empresa para este tipo de requerimientos.

1.2 Descripción del problema

Gran parte de ese conjunto de procesos está ligado a una serie de validaciones, autorizaciones y gestión manual que fluctúan entre el manejo de múltiples correos de control y de evidencia o en la realización de demasiados pasos dentro del ERP empresarial por donde convergen los procesos, esto realizado por varias áreas técnicas y comerciales que actúan durante todo el ciclo de entrega del requerimiento. Además, dentro el Sistema se realizan validación manual de la disponibilidad de recursos lógicos para aprovisionar cada máquina virtual requerida, además de las configuraciones de las mismas por parte del área de TI, por tanto, esto dificultad la celeridad que se debe tener.

Otro problema que surge a partir de este procedimiento es que, al no tener conectados los sistemas de forma adecuada, se pierde el control de inventario de cuantos recursos lógicos han sido asignados, por tanto, cuando una máquina llega a ser retirada, es complicado poder tener visibilidad de la disponibilidad que se tiene para futuros aprovisionamientos, que, en su mayoría, deriva en conflictos de espacio o configuraciones al no tener un control más inmediato y sobre todo automático. Dado todo lo mencionado, el mayor riesgo que resulta del problema es el impacto directo sobre la cartera

de clientes en la percepción del servicio ya que al no tener resultados inmediatos en un proceso que debería resultar bastante ágil y adicional a las inconformidades que de esto resulte, podría causar mala percepción respecto a las herramientas tecnológicas sobre las cuales se apalanca la operación pudiendo causar que desistan en el momento de solicitar nuevos servicios y que los lleve a evaluar ofertas de la competencia.

1.3 Solución propuesta

Es clave resolver estas dificultades en el aprovisionamiento porque nos permitirá poder responder de forma más rápida ante las necesidades de los clientes. Una forma necesaria para poder plantear esta mejora es empezar determinando y entendiendo en que se basa cada etapa o subproceso que se ejecuta en el flujo en mención. Entonces, identificar claramente el cómo está planteado el proceso actual ayudaría a entender e identificar los puntos más débiles que se deben mejorar, para esto, es importante enfocarnos en lo que nos indica el dominio de la Arquitectura de Negocio pudiendo identificar claramente procesos, roles, actividades y funciones que estén definidos por la organización para el flujo a analizar [1], poder esquematizarlo y plantear las oportunidades de mejora [2] para tener el Blue print de la transformación digital a aplicar [3].

Para poder cumplir con lo mencionado, primero se requiere poder esquematizar tanto el modelo actual como la propuesta de mejora, nos ayudaremos del modelamiento de estos procesos mencionados utilizando

herramientas BPMN [4] . Dado que de acuerdo a proyectos enfocados en automatización y transformación digital, una vez identificado cada proceso interviniente es mucho más sencillo poder implementar automatizaciones, es decir, como una estrategia de negocio, la descomposición de aquellos procesos importantes permitirá liberar la carga operativa [5].

Una parte importante, es la selección de los insumos y herramientas tecnológicas que nos ayuden con el objetivo, en este caso, para el modelamiento de los procesos necesarios se hará uso de Camunda como plataforma BPM [6]-[7], en donde podremos no sólo implementar los flujos requeridos (AS-IS y TO-BE), sino también, establecer las diferentes reglas DMN y entradas/salidas necesarias que permitan integrar los componentes requeridos que estarán basados en la Arquitectura de Software orientada en Microservicios [8], los cuales estarán contruidos con la capacidad necesaria que ayuden a ir empujando cada fase del aprovisionamiento de forma más ágil, eficiente y con alta disponibilidad.

El diseño de mejora a plantear considerará la integración entre: el ERP de la empresa, la arquitectura de Microservicios y la plataforma de virtualización VMware VCloud Directory por medio de los servicios disponibles vía API (Application Programming Interface) para ejecutar cada etapa del aprovisionamiento que la plataforma provee [9]-[10] y todo esto orquestado por BPMN. ¿En qué nos ayuda esta solución? Básicamente, permite darle una gestión de aprovisionamiento automatizada a partir de un disparador inicial desde el ERP en la sección del flujo en la que interviene actualmente el área

Comercial, es decir, una vez ingresado el requerimiento e ingresadas las características técnicas del Servicio, se activará el flujo en Camunda que irá empujando a cada microservicio encargado de la comunicación vía API con VCloud Directory acorde a la necesidad del cliente. Estas características deberán ser enviadas al API la cual irá determinando que tipo de virtualización aplicar y en qué espacio deben ser creadas las máquinas virtuales requeridas, todo esto, determinado de forma automática por la plataforma validando disponibilidad de recursos.

1.4 Objetivo general

Automatizar el flujo de aprovisionamiento y configuración de servicios de Hosting relacionados con la generación de máquinas virtuales para agilizar el cumplimiento de tiempos de entrega utilizando BPMN.

1.5 Objetivos específicos

Analizar y establecer el estado inicial del proceso de aprovisionamiento y configuración de servicios de Hosting.

Diseñar un proceso automático de aprovisionamiento y configuración respecto a la asignación de recursos lógicos necesarios para poder desplegar las máquinas virtuales.

Esquematizar un diagrama de arquitectura de alto nivel (componentes) de la solución resultante dado el modelamiento de mejora.

Evaluar el resultado de la prueba de concepto a ser realizada en comparación con el estado actual del proceso.

1.6 Metodología

Este trabajo de titulación está basado en un estudio cuantitativo con un alcance descriptivo dado que en función de determinar las diferentes causas que producen los tiempos elevados de cada etapa del proceso, permite determinar aquellos puntos de fallos que necesitan ser mejorados y qué efectos producen sobre los clientes y sus requerimientos.

Para este estudio se determina un diseño No Experimental de tipo Ex post facto, dado que la mejora está enfocada en optimizar los tiempos que dura cada proceso planteado, se tomará como referencia la información registrada por los sistemas de la empresa, que refleja los tiempos que toma cada etapa realizada de forma manual actualmente, estos nos ayudarán a validar el problema planteado en función de estas medidas derivadas de la operación. A partir de eso, el objetivo está alineado a realizar un diseño usando BPMN que ayuden a reducir estos tiempos de trabajos manuales por procesos que se encarguen de este trabajo de forma más rápida y eficiente.

Este trabajo estará basado en tres etapas, las cuales están descritas a continuación:

En la primera etapa se debe establecer las características de requerimientos de Hosting levantados por los clientes y explicación de las diferentes etapas sobre las cuales el flujo funciona.

En la segunda se determina del modelo actual del proceso a ser mejorado y se realiza el análisis de los datos históricos para evaluar los tiempos que toma el proceso como punto de partida de cara a la mejora a ser planteada. En este punto las variables que se utilizarán para esta medición son las siguientes:

- Tiempo de aprovisionamiento
- Tiempo de configuración

En la siguiente tabla se describe cada variable propuesta

Variable	Definición Conceptual	Definición Operacional
Tiempo de aprovisionamiento	Variable que mide la duración de un proceso o acontecimiento	Tiempo medido entre la cual una orden de servicios ingresa al área de TI, esta lo aprovisiona y es entregado para su configuración.
Tiempo de implementación y entrega	Variable que mide la duración de un proceso o acontecimiento	Tiempo medido entre que el recurso se entrega aprovisionado, este es configurado por el área de Soporte de Operaciones y este es entregado al cliente

Tabla 1 : Variables de medición

La información para recopilar estará limitada a el último año de gestión de los clientes con este tipo de requerimientos. La medida que se tendrá en cuenta será en función de horas, determinada entre cada etapa del flujo.

En la tercera etapa, se establecerá un diseño estimado del cómo debería funcionar la solución, la cual conjuga todas las tecnologías planteadas para llegar a cumplir con tiempos de entrega eficientes a partir de un diseño de arquitectura de alto nivel.

En la última etapa, se realizará el desarrollo y evaluación de la Prueba de concepto del diseño de mejora planteada y la validación de los resultados con los stakeholders.

CAPÍTULO 2

MARCO TEÓRICO

El objetivo del siguiente capítulo es poder contextualizar las diferentes herramientas tecnológicas, conceptos y definiciones que servirán como referencia para el desarrollo de la propuesta y poder cumplir con el alcance de este proyecto.

En primer lugar analizaremos la estrategia del modelamiento BPMN y cómo ésta aporta a la implementación de flujos de procesos adecuados para asentar una base y sus posibles mejoras, estas, de mira a los modelos AS-IS y TO-BE. Luego analizaremos una herramienta la cual nos permita hacer despliegue, configuración y ejecución de los flujos establecidos, para este efecto, veremos a fondo lo que Camunda nos ofrece y entendiendo el por qué de su elección y una overview de su arquitectura. Una vez revisado todo lo referente a procesos, nos orientaremos hacia el cómo implementar estas integraciones de los procesos, para esto, entender cómo

la arquitectura orientada a microservicios nos ofrece una potente solución y ecosistema consistente para la creación de soluciones y por último entenderemos conceptualmente a qué nos referimos con la virtualización, componentes y cómo podemos integrar estos conceptos a nuestro ecosistema por medio del consumo de APIs REST disponibilizados tanto por los componentes internos como los virtualizadores que VMWare ofrece con su administrador.

2.1 Modelamiento BPMN

2.1.1 Modelado de Procesos

Para poder tener orden y entender cómo funcionan cada uno de los procesos de la operación de las empresas, es importante poder tener tener visibilidad de los mismos y sus diferentes actores, para esto, nos ayudamos de una herramienta potente que, mediante el modelamiento de estos, nos ayuda a darle mayor fuerza al negocio ya que permite ver y medir cuan eficientes son y además identificar puntos de fallo y sobre estos, poder realizar mejoras potenciales, esto, agregando o eliminando pasos según sea el caso y estableciendo los actores con mayor facilidad [11].

Este modelamiento se llama Procesos de Negocio y que son un flujo progresivo de actividades que se siguen para llegar a un determinado objetivo y sobre todo con el objetivo de brindar valor a los clientes, como se mencionó, cada uno de estos procesos debe tener un dueño

ejecutante, estos pueden ser, humanos o procesos autónomos. Estos procesos de negocio poseen las siguientes partes:

- Entradas
- Salida o productos resultantes
- Recursos útiles para estas salidas (humanos o de algún otro tipo)

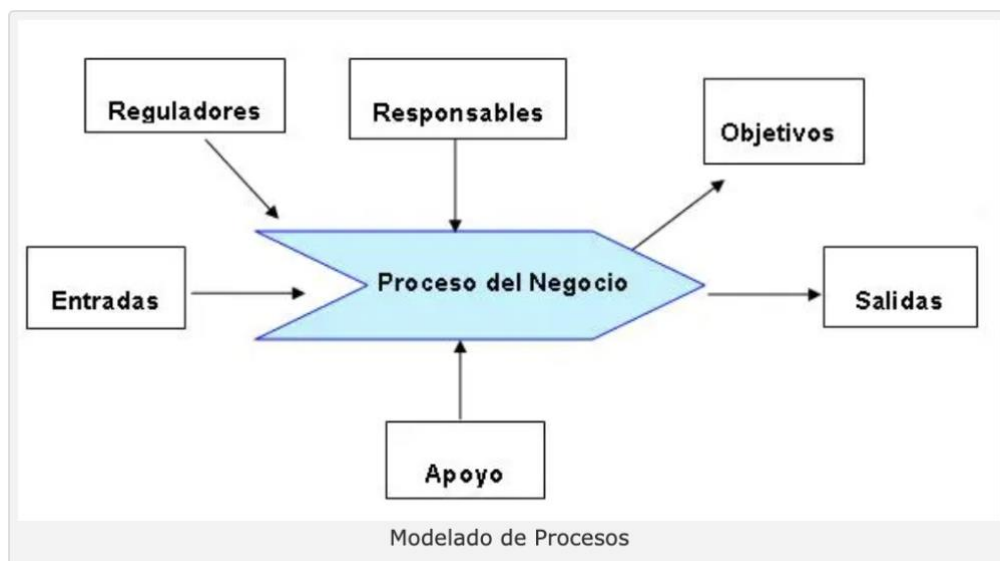


Imagen 2.1 : Modelado de Procesos

Fuente : Modelado de procesos [11]

2.1.2 Notación BPMN

El Business Process Modeling Notation (BPMN), es la notación estandarizada para representar de forma gráfica todos los procesos de negocio. Es un enfoque sistemático que permite levantar, documentar, diseñar, ejecutar, medir y controlar los procesos tanto manuales como automatizados para poder obtener resultados consistentes en pro de

los objetivos del negocio [12]. Tiene el propósito de servir como lenguaje común para poder cerrar la brecha que existe entre el diseño de los procesos del negocio y su implementación.

El modelamiento en BPMN se realiza mediante un conjunto de diagramas sencillos y varios elementos gráficos que incluyen símbolos, relaciones y atributos, esto, facilita que tanto los usuarios dueños de los procesos de negocio como los desarrolladores entiendan en un mismo nivel los diferentes flujos modelados.

A continuación se muestra una gráfica con los diferentes elementos básicos usados en el modelamiento:

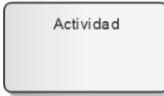

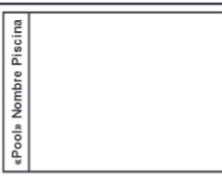
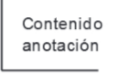



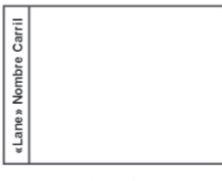




Objetos de Flujo	Objetos de Conexión	Canales (Swimlanes)	Artefactos	Datos
 Actividad Actividades	 Flujo de Secuencia	 «Pool» Nombre Piscina Piscina	 Contenido anotación Comentario	 Objeto de datos Objeto de datos
 Eventos	 Flujo de Mensaje	 «Lane» Nombre Carril Carril	 Agrupación	 Almacén de datos
 Compuertas	 Asociación			

Imagen 2.2 : Elementos básicos de notación BPMN

Fuente: Elementos de notación BPMN [12]

Objetos de flujo: Son objetos principales dentro del flujo BPMN:

- **Actividades:** Describen el trabajo desarrollado durante el proceso

- **Eventos:** Describe sucesos durante el desarrollo de un proceso y que pueden afectar a su flujo, estos pueden ser de inicio, intermedios y de fin.

- **Compuertas:** Controles de secuencia dentro del flujo, determina bifurcaciones o combinaciones de las rutas de los flujos dependiendo de alguna condición establecida.

Objetos de conexión: Relacionan todos los objetos dentro del flujo, estos son:

- **Flujo de secuencia:** Enlaza dos elementos en flujos normales, condicionales o por defecto.

- **Flujo de mensaje:** Indica el envío de mensajes entre dos elementos ubicados en pool distintos.

- **Asociación:** Se utiliza para conectar artefactos a objetos de flujo.

Canales: Representa a los responsables de las actividades en un proceso.

- **Piscina:** Puede contener uno o más carriles que pueden representar los diferentes stakeholders, usuarios o participantes del proceso.

- **Carril:** Muestra un rol o área funcional dentro de la piscina, sirve para categorizar las actividades en función de los roles o áreas.

Artefactos: Elementos de documentación para hacer más comprensible los diagramas.

- **Comentario**: Nota descriptiva de un documento específico del diagrama

- **Agrupación**: Agrupa diferentes actividades sin afectación del flujo.

Datos: Representa archivos de datos, objetos de datos o documentos que son producidos o consultados por un proceso o actividad.

2.1.3 Mapeo de procesos AS-IS/TO-BE

El mapeo de procesos tanto AS-IS como TO-BE es una herramienta de gestión que ayuda con la descripción y mejora de los procesos internos de una organización [2].

Mapeo AS-IS

Es la definición de la situación actual de un proceso con sus actores y actividades, por lo general existen procesos que no tiene su debido detalle especificado en el respectivo detalle o documentación complementaria, para poder resolver esta posible problemática, es necesario solicitar a los usuarios claves, dueños de cada de uno de estos procesos, las definiciones de los mismos en una etapa de levantamiento de información.

Mapeo TO-BE

Es la definición del futuro, de la mejora de un determinado proceso o hacia donde se quiere llegar, para llegar a esto, se requiere de la ayuda de herramientas que sirvan de apoyo además de que los participantes, dueños de los procesos o usuarios tengan la experiencia con estos y que contribuyan a la optimización de los mismos.

Es clave poder tener el mapeo de procesos internos, dado que se convierte en un punto diferencial dentro de la organización, nos ayuda a tener más claro cada uno de ellos y permite además tomar mejores decisiones dado que se tiene visibilidad de actividades y sus responsables y además, va a permitir aumentar la productividad y sobre todo mejorar la calidad de servicio de cara a los clientes.

Una buena estrategia para poder cumplir con los lineamientos del modelamiento es, como se mencionó, detectar aquellos usuarios claves, dado que son los que poseen el conocimiento acerca de las reglas de negocio y que sobre todo realizan el proceso diariamente, esto es clave para recopilar toda la información necesaria [2]. Para efecto de este proyecto es super importante identificar cual es el estatus actual del proceso a ser mejorado y luego analizar el cómo se va a resolver el problema y para esto es importante tanto la visión del usuario cómo de la tecnología a ser utilizada.

Posteriormente, como paso resultante de este modelado, es el análisis de efectividad midiendo la productividad tomando como referencia las salidas generadas y los tiempos de gestión de cada actividad y validando en conjunto con los dueños de los procesos para determinar si el objetivo es cumplido y cuan eficiente es.

A continuación, se muestra una referencia del cómo se representa el mapeo de procesos:



Imagen 2.3 : Mapeo de procesos

Fuente : Mapeo de procesos [13]

Por último, es importante recordar que la gestión de estos procesos tiene un ciclo de vida que puede resultar cambiante y que está centrado en la mejora continua, es decir, el descubrimiento de nuevos puntos de mejora, hace que el proceso se encuentre en constante evolución, como se detalle en la siguiente imagen:

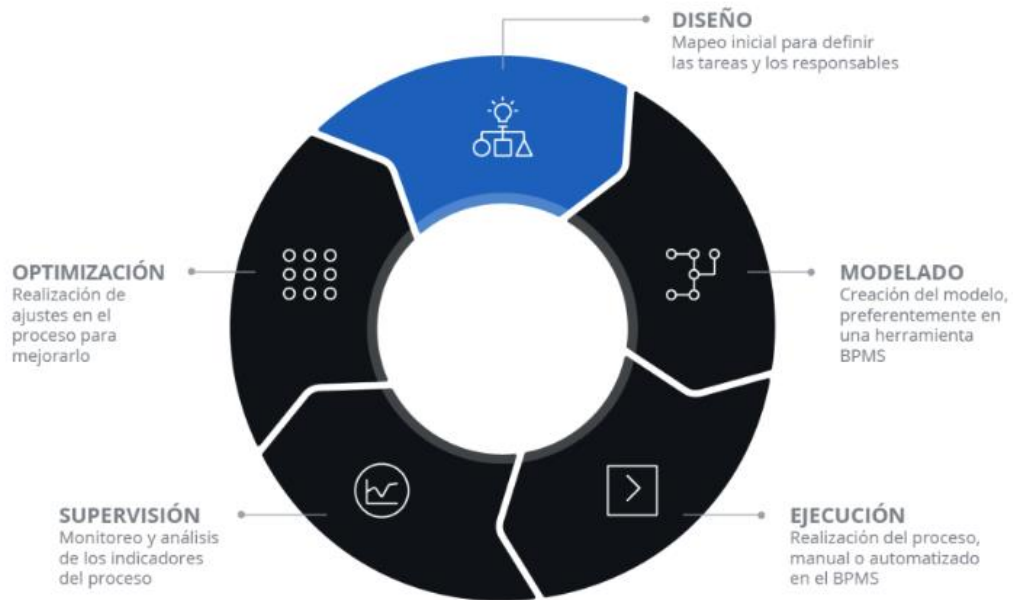


Imagen 2.4 : Ciclo de vida de los procesos

Fuente : Mapeo de procesos [13]

2.2 Camunda como plataforma BPMN

Camunda es un Framework basado en Java para realizar flujos de automatización de procesos, complementario como herramienta que ayuda a los modelos de negocio ser llevados a ejecución en conjunto con otras tecnologías complementarias.

2.2.1 Motivación

Una de las razones clave en la elección de esta herramienta BPMN es en primer lugar por tener capacidad de ser Open Source, dado que es parte de la filosofía de la empresa el manejo de este tipo de

herramientas y por otro lado Camunda ya es un Software que se viene utilizando en la operación.

Por otro lado, aterrizando aspectos técnicos, la razón detrás de haber elegido Camunda es por ser un lenguaje de modelado conocido, fácil de aprender y bastante popular para la descripción de procesos. Además, que los usuarios puede desplegar sus propios modelos en el Sistema y al estar desarrollado en Java, estos, pueden escribir sus propias clases para añadir comportamientos específicos a los servicios. Otro aspecto es su capacidad de integrarse fácilmente con modelos BPMN independientes del dominio. Por último, tiene una amplia comunidad de aporte como recursos y guías disponibles y una extensiva documentación de apoyo [14].

Camunda tiene una ventaja que al ser Open Source, todo su código y paquetes más importantes están disponibles abiertamente desde los repositorios oficiales y la guía de cómo hacer uso de ellos con las herramientas complementarias, como el Camunda Modeler y el Camunda Engine que proveen una capa de API para poder tener acceso, ejecución y consulta a los procesos implementados [6].

2.2.2 Arquitectura

A continuación se presenta una visión general de la arquitectura de Camunda, así como los roles típicos de los usuarios que interactúan con los diferentes componentes del sistema:

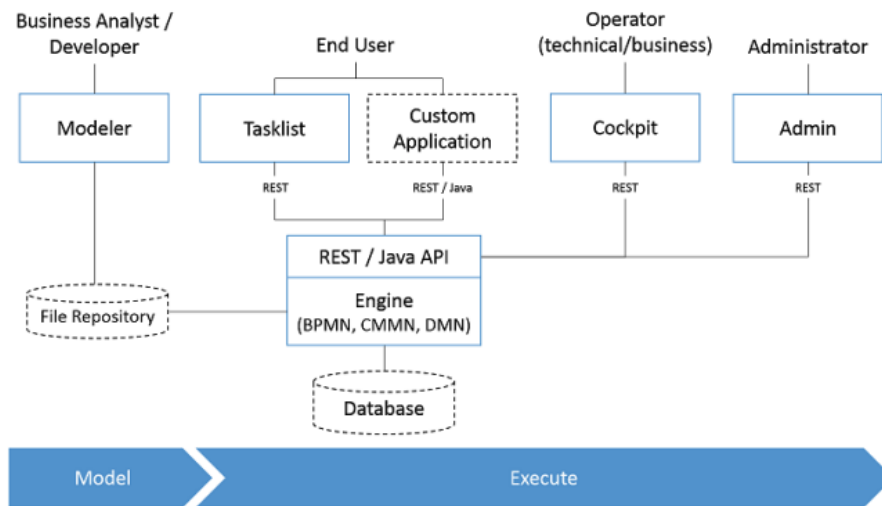


Imagen 2.5 : Overview de la arquitectura de Camunda

Fuente: Overview de la arquitectura de Camunda [14]

Observando la Imagen 2.5, vamos a mencionar brevemente los componentes mostrados en la misma, que son claves para el funcionamiento de todo el ecosistema de Camunda [14]:

The Engine: Este componente se encuentra en el núcleo de Camunda, se encarga de traducir el modelo BPMN a objetos Java y almacenará su representación internamente, básicamente gestiona operaciones como la creación, activación, suspensión o eliminación de una instancia de proceso, el despliegue o la definición del mismo. El motor hace la navegación a través del modelo gráfico de una actividad a otra, ejecuta sus comportamientos y se comunica con participantes externos del proceso (como servicios Web u otros motores), esta última capacidad es clave para el efecto de cumplir con el alcance de este proyecto.

A continuación se muestra una representación de la arquitectura del Process Engine:

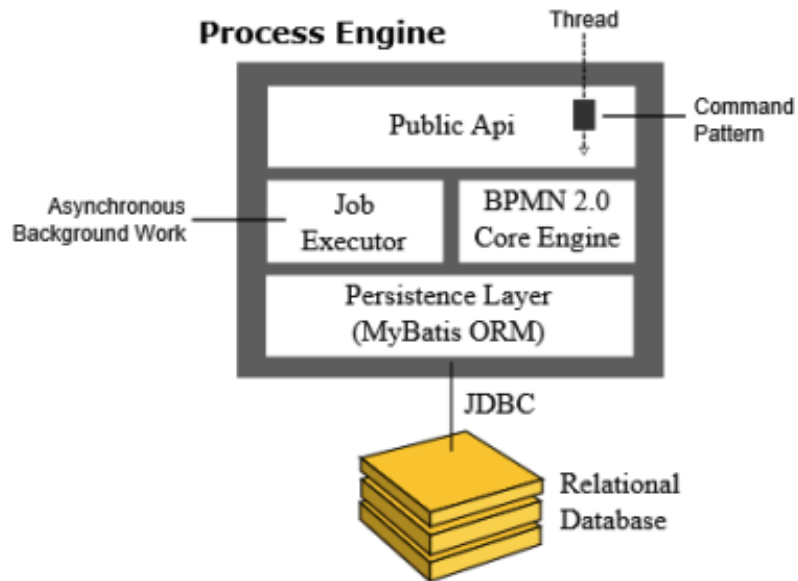


Imagen 2.6 : Arquitectura del Engine de Camunda

Fuente: Conceptos, diseño e implementación de Camunda [14]

Modeler: Es una aplicación Standalone, un editor visual que es usado para realizar el modelamiento BPMN, esta aplicación puede ser descargada e instalada en el sistema del usuario. Ofrece un set completo de herramientas como el modelamiento de procesos, crear tareas y asignación a usuarios o procesos externos, estos modelos luego podrán ser desplegados en el Camunda Engine explicado anteriormente directamente desde la interface del usuario.

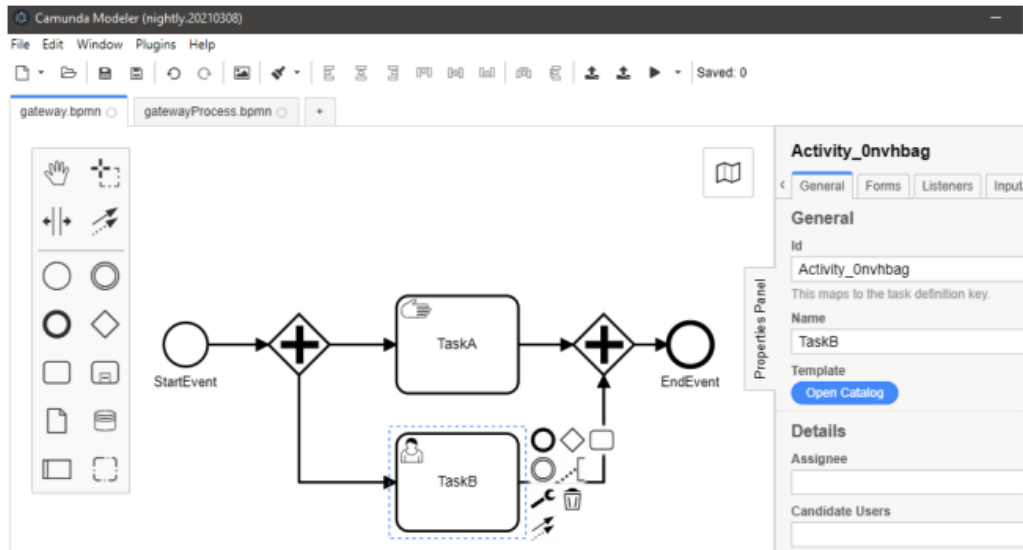


Imagen 2.7 : Camunda Modeler Interface

Fuente: Conceptos, diseño e implementación de Camunda [14]

Cockpit: Dentro de las aplicaciones front-end que Camunda tiene, este es el más importante desde el punto de vista del desarrollador por las múltiples funcionalidades que posee y la información que provee, puede ser usado para administración como para monitoreo de los modelos de procesos y sus despliegues, sus instancias y actividades. En esta, se muestra información específica de los procesos, como su versionamiento, el número de instancias corriendo, etc. El Cockpit también permite la inspección particular de una instancia de un proceso, como variables y asignaciones.

The screenshot displays the Camunda Cockpit interface. On the left, the 'Definition Version' is set to 1, and the 'Definition Name' is 'Gateway'. The main area shows a BPMN diagram of a process starting with a 'StartEvent', followed by an XOR gateway, then two parallel tasks 'TaskA' and 'TaskB', another XOR gateway, and finally an 'EndEvent'. Below the diagram, the 'Process Instances' tab is active, showing a table with three instances.

State	ID	Start Time	Business Key
✓	20f7106b-76b6-11eb-ab24-00d861fc144c	2021-02-24T16:36:57	
✓	1fbe4242-76b6-11eb-ab24-00d861fc144c	2021-02-24T16:36:55	
✓	45bb1003-76b4-11eb-bbf3-00d861fc144c	2021-02-24T16:23:40	

Imagen 2.8 : Camunda Cockpit

Fuente: Conceptos, diseño e implementación de Camunda [14]

Tasklist: Esta aplicación permite el manejo de las actividades de usuarios, como la búsqueda, filtros, crear nuevas tareas o completar alguna ya existente, también permite reasignar tareas y establecer fechas de cumplimiento. Después de que una tarea fue asignada a un usuario dentro de Cockpit, este aparecerá dentro de la lista de tareas y el usuario puede tener acceso a estas.

Otra funcionalidad importante es la posibilidad de iniciar procesos nuevos, mediante el botón Start Process en la parte superior derecha de la pantalla de la aplicación. Una vez, presionado el inicio, se mostrará al usuario un formulario genérico en el cual se pueden

especificar variables (de ser el caso) y luego se dará inicio al proceso seleccionado.

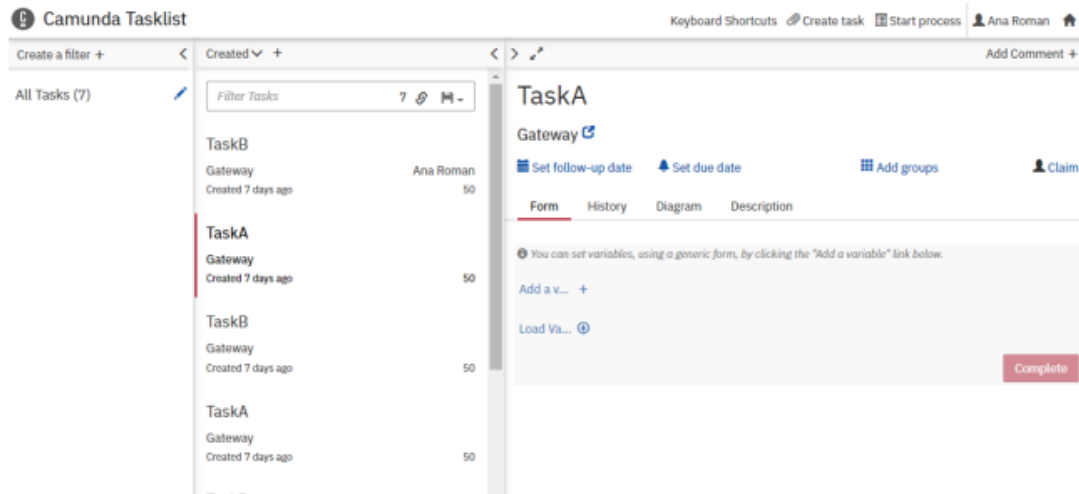


Imagen 2.9 : Camunda Tasklist

Fuente: Conceptos, diseño e implementación de Camunda [14]

Admin: Esta aplicación es responsable para la administración de derechos de usuario, identidad y autorización, se puede tener visibilidad general de todos los usuarios registrados, sus funciones, tareas y pertenencias a grupos.

Tomar en cuenta que, para poder extender la explicación y ver más detalle de las especificaciones mencionadas, se pueden encontrar dentro de la documentación oficial [6].

2.3 Arquitectura orientada a microservicios

2.3.1 Descripción general

Las arquitecturas de microservicios han venido siendo un patrón para el diseño de aplicaciones que van dejando atrás el desarrollo el esquema de arquitecturas monolíticas, ofrecen un desempeño mas resiliente, distribuido, robusto, ligero, desacoplado, mantenimiento rápido, ágil, entre otras características que a hoy, ofrecen crear aplicaciones potentes de distintas índoles. En el sector de telecomunicaciones, entendiendo cómo se mueve cada proceso de la operación, se vuelve fundamental poder aplicar este patrón para poder armar bloques que contienen funcionalidades específicas, permitiendo, poder mantener estable todo un core funcional al no tener dependencia entre un módulo y otro, esto mejora significativamente lo que aplicaciones monolíticas pueden brindar. La capacidad de escalabilidad es otro factor primordial que nos brinda este patrón de arquitectura de aplicaciones y sobre todo tener estrategias de integración de estos pequeños modulos o microservicios que permitan poder conectar funcionalidades acorde las necesidades de la operación.

Otra ventaja que se tiene es que, cada microservicio es un código que puede estar en un lenguaje de programación diferente y como se mencionó, desempeña una función específica, pueden ser servicios

únicos para una solución o puede ser servicios transversales a toda la operación.

Sin embargo, también posee ciertas desventajas, dado que pueden llegar a consumir memoria, depende del conocimiento del desarrollador y sobre todo de una infraestructura y complementos tecnológicos en donde se pueda apalancar estas implementaciones de forma más eficiente. Pero, aun así, siempre va a depender de la necesidad de la operación sobre qué estrategia será la mejor a ser utilizada y no siempre por lo mencionado, los microservicios serán un primera alternativa [15].

A continuación se muestra una diferenciación entre ambos tipos de arquitectura, la tradicional o monolítica y la orientada a microservicios:

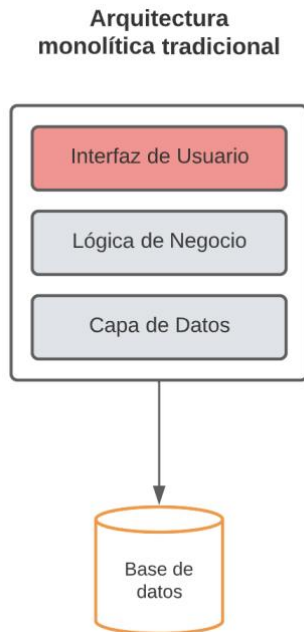


Imagen 2.10 : Tipos de arquitectura de aplicaciones – Arquitectura monolítica

Fuente: Autor

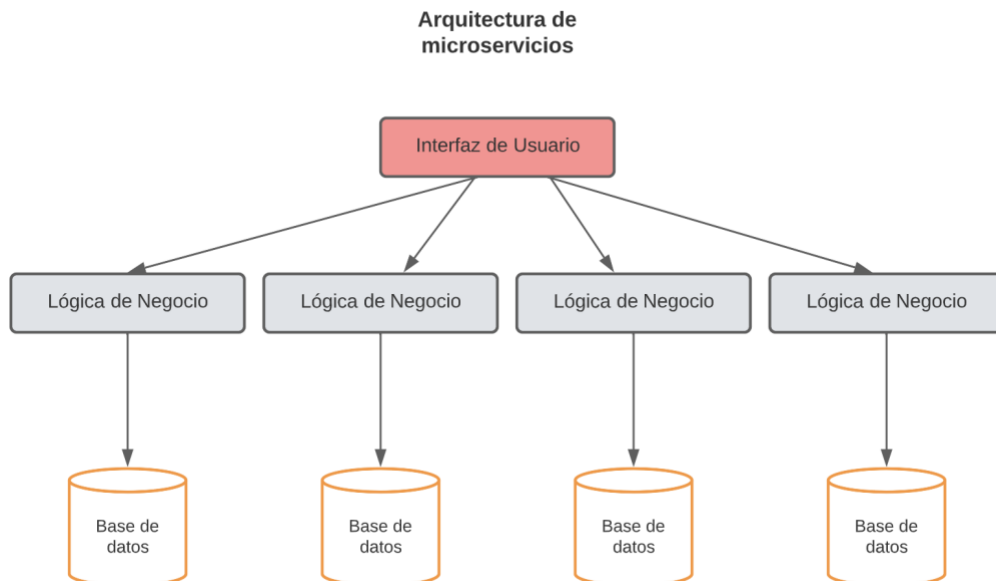


Imagen 2.11 : Tipos de arquitectura de aplicaciones – Arquitectura de microservicios

Fuente: Autor

2.3.2 Orquestación de Microservicios

Entendiendo los tipo de arquitectura para aplicaciones, otro aspecto que se debe resolver es justamente el cómo se puede o debe utilizar este patrón. Como se mencione la arquitectura orientada a microservicios permite construir aplicaciones distribuidas como conjunto de pequeños servicios desacoplados que trabajan de forma independiente pero pueden llevar un flujo coordinado u orquestado.

Una característica de estos artefactos es que pueden exponer sus funcionalidades con mecanismos tan ligeros como APIs REST (Representational State Transfer) para comunicarse vía protocolo HTTP (Hypertext Transfer Protocol) o mediante mensajería.

Ahora bien, actualmente, los microservicios no se disponibilizan por sí solos de manera pura, existen herramientas potentes que apalancan su despliegue de manera más eficiente y mantenible, esto con el uso de Contenedores, estos, son procesos que contienen las aplicaciones internamente que se desean ejecutar con todas sus dependencias con ayuda del kernel del Sistema Operativo que se esté usando, esto los hace 100% portables, dado que, son independientes al software o hardware sobre el cual fueron creados [16]. Adicional, una característica de estos contenedores que permite desplegar aplicaciones de forma muy rápida y escalable.

Entonces, ya al trabajar con datos masivos, datos distribuidos, es necesario realizar escalamiento horizontal de las aplicaciones que transaccionan con ellos, esto es factible con el uso de múltiples nodos (clusters) logrando un ambiente distribuido, a continuación se muestra un ejemplo de una arquitectura de cluster de contenedores:

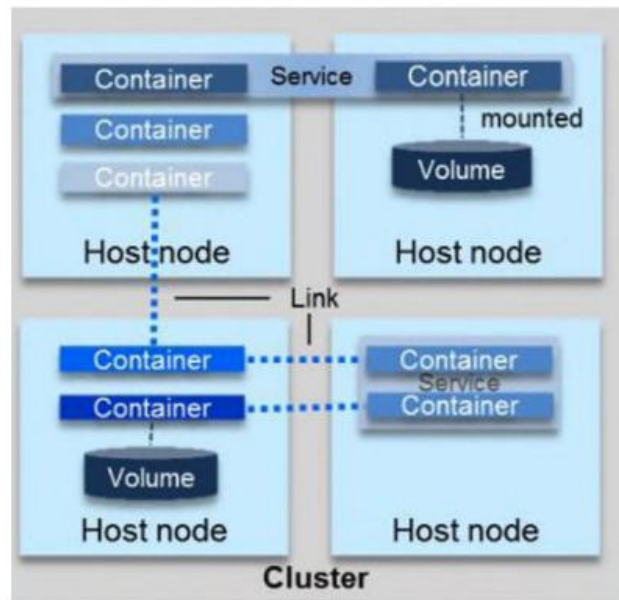


Imagen 2.12 : Arquitectura de un clúster de contenedores

Fuente: Orquestación de microservicios [16]

Cada clúster es un servidor virtual y puede tener múltiples contenedores, entonces para poder trabajar con este tipo de esquemas, se requiere un estrategia de orquestación o de coreografía.

Orquestación, quiere decir en términos generales que existe una entidad central (pudiese ser un mismo microservicio) que se encargue de la comunicación entre cada componente, permitiendo así que entre

cada uno de ellos no haya comprensión de quién debe consumir o ser consumido.

Coreografía, quiere decir que cada servicio o componente conoce en que momento y con quién debe interactuar.

Va a depender siempre el escenario sobre el cual se desee trabajar y el objetivo para elegir una estrategia u otra dentro de la arquitectura.

2.3.3 Comunicación API REST

Un API REST (Representational State Transfer) describe un conjunto de recursos y operaciones que pueden ser llamados desde cualquier cliente HTTP, incluido desde un código JavaScript de lado del cliente ejecutado en un navegador web [18].

Pero también se puede usar este mecanismo para poder comunicar microservicios mediante el mismo protocolo HTTP y no necesariamente orientado a un navegador web, este último caso, es el que generalmente se usa dentro del consumo de recursos que son disponibilizados por cada microservicio dentro del ecosistema o solución, estos pueden exponer un punto de entrada vía API hacia una aplicación web como puede ser usado para consumo entre aplicaciones de manera interna y automática acorde como se establezca las integraciones.

Las APIs pueden manejar versiones acorde a la necesidad de la operación y se tipifican en los path que se definan, por ejemplo, la ruta base para la primera versión podría ser /student/v1 mientras que para la segunda versión sería /student/v2.

Las operaciones en un API REST tienen un nombre y un método HTTP, estos son: GET, POST, PUT o DELETE, este nombre debe ser único en todos los recursos de esta API, dando así, la definición del API de forma más completa y de mejor comprensión, como se ve en el siguiente ejemplo:

Métodos HTTP	Operación	Descripción
GET	getStudent/	Obtiene información del estudiante desde una fuente de datos
PUT	updateStudent/	Actualiza el detalle de un determinado estudiante
DELETE	deleteStudent/	Elimina la referencia de un estudiante

Tabla 2 : Ejemplos de Métodos HTTP

2.4 Virtualización

Como concepto general, podemos decir que, la virtualización es una estrategia para desplegar los recursos del ordenador en diferentes capas aisladas como hardware, software, datos, red, almacenamiento [19]. La virtualización fue una manera lógica de particionar equipos mainframe en máquinas virtuales independientes, permitiendo realizar múltiples tareas simultáneas, es decir, procesos y aplicaciones al mismo tiempo.

2.4.1 Terminología

Empezaremos analizando la capa de virtualización, esta es una capa de software que se adiciona entre el hardware y los sistemas operativos. Esta capa le permite a múltiples instancias de sistemas operativos correr de manera concurrente dentro de máquinas virtuales en un único servidor físico, garantizando el particionado y la compartición de los recursos físicos disponibles como el CPU, la memoria y los dispositivos de almacenamiento y de entrada y salida [19].

El primer termino y quizá más importante que analizaremos es el *hypervisor*, que es quien asigna los recursos de hardware. Este es el software que crea un ambiente de máquinas virtuales en una computadora. En este ambiente, el hypervisor es el programa de control maestro, con el más alto nivel de privilegios, y administra uno o más sistemas operativos, a los que se refiere como sistemas

operativos huéspedes. Este se inicia durante el arranque, antes que las máquinas virtuales, y que normalmente corre justo sobre el hardware [20].

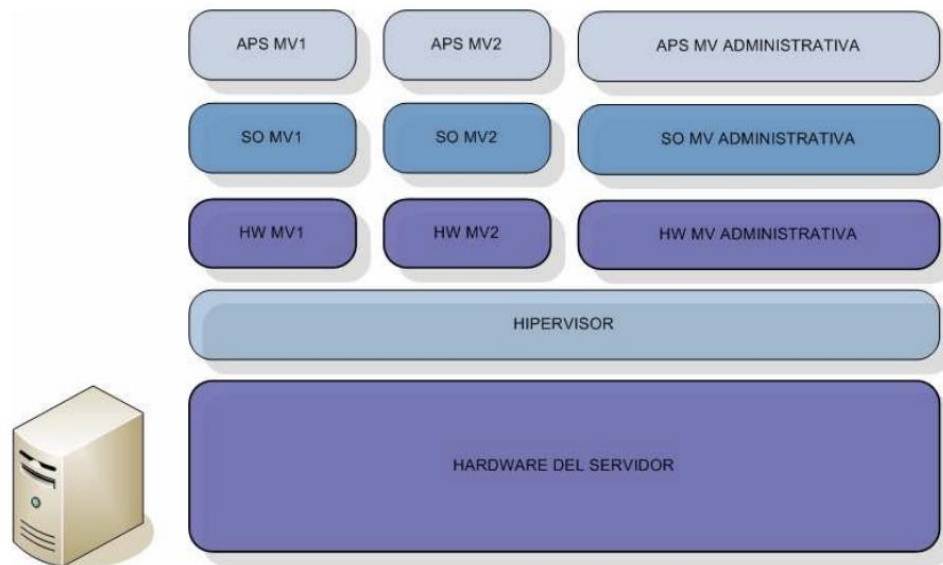


Imagen 2.13 : Hypervisor sobre Hardware

Fuente: Virtualización [20]

Para sistemas de arquitectura x86, los hypervisores se clasifican en 2 arquitecturas principales:

- Arquitectura hospedada
- Arquitectura nativa

La primera instala y corre una capa de virtualización encima de un Sistema operativo y la segunda instala una capa de virtualización directamente en el hardware de sistemas basado en x86, esta última, es más eficiente.

A continuación, se muestra dos imágenes que detallan lo mencionado:

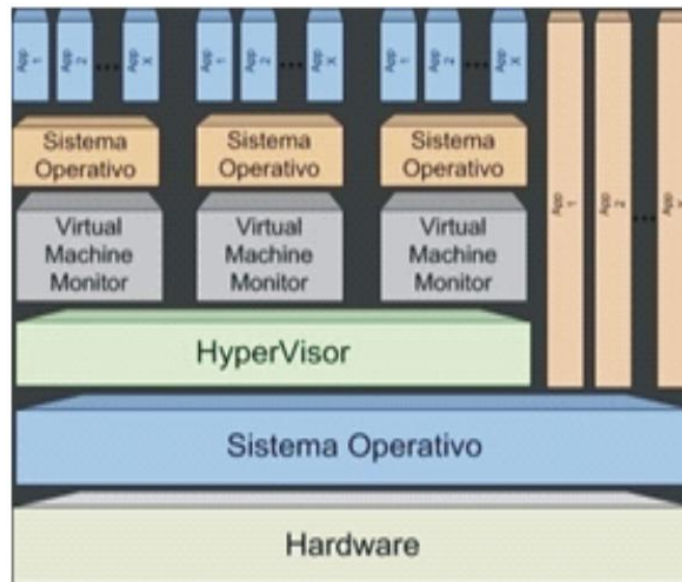


Imagen 2.14 : Arquitectura hospedada

Fuente: Virtualización [19]

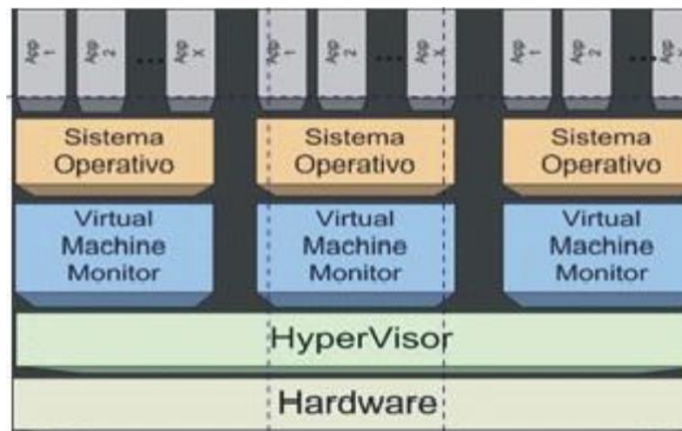


Imagen 2.15 : Arquitectura nativa

Fuente: Virtualización [19]

Modelos de Virtualización

Podemos distinguir cuatro modelos principales de virtualización [20]:

Virtualización de plataforma: En términos generales consiste en la abstracción de todo el hardware subyacente de una plataforma de manera que múltiples instancias de sistemas operativos puedan ejecutarse de manera independiente, con la ilusión de que los recursos abstraídos les pertenecen en exclusiva. Esto es muy importante, ya que cada máquina virtual no ve a otra máquina virtual como tal, sino como otra máquina independiente de la que desconoce que comparte con ella ciertos recursos.

Sobre este modelo podemos también apreciar y diferenciar algunos tipos de paradigmas de virtualización de plataforma existentes:

Sistemas operativos invitados: Sobre una aplicación para virtualización –no hace uso de hipervisor u otra capa de virtualización- que corre sobre la instancia de un sistema operativo –sistema operativo host- se permite la ejecución de servidores virtuales con sistemas operativos independientes. Algunos ejemplos de soluciones de este tipo son VMware Workstation, Parallels Desktop, Sun xVM VirtualBox, VMware Player, y Microsoft Virtual PC.

Emulación: Un emulador que replica una arquitectura hardware al completo – procesador, juego de instrucciones, periféricos hardware- permite que se ejecuten sobre él máquinas virtuales. Por lo tanto se permite la ejecución de sistemas operativos y aplicaciones distintos al instalado físicamente en la máquina que ejecuta el emulador. Los

emuladores más importantes actualmente son Bochs, MAME, DOSBox, Hercules, MESS, VirtualPC, y Qemu.

Virtualización completa: También llamada nativa. La capa de virtualización, un hipervisor, media entre los sistemas invitados y el anfitrión, la cual incluye código que emula el hardware subyacente –si es necesario- para las máquinas virtuales, por lo que es posible ejecutar cualquier sistema operativo sin modificar, siempre que soporte el hardware subyacente. El código de emulación puede provocar pérdida en el rendimiento. Dentro de esta categoría encontramos soluciones como: VMware Server, XenServer, z/VM, Oracle VM, Sun xVM Server, Virtual Server, VMware ESX Server, VMware Fusion, Xen, Hyper-V.

Paravirtualización: Similar a la virtualización completa porque introduce hipervisor como capa de virtualización, pero además de no incluir emulación del hardware, introduce modificaciones en los sistemas operativos invitados que por consiguiente están al tanto del proceso. Las soluciones más extendidas e importantes dentro del paradigma de la paravirtualización son Xen, Logical Domains, Oracle VM, y Sun xVM Server.

Virtualización a nivel de sistema operativo: Virtualiza los servidores sobre el propio sistema operativo, sin introducir una capa intermedia de virtualización. Por lo tanto, simplemente aísla los servidores independientes, que comparten el mismo sistema operativo. Aunque

requiere cambios en el núcleo del sistema operativo, ofrece rendimientos próximos al sistema sin virtualizar. Como ejemplos representativos de este modelo podemos citar OpenVZ, Linux V-Server, Virtuozzo, FreeBSD's chroot jails, Free VPS, Solaris Containers y Solaris Zones.

Virtualización a nivel de kernel: Convierte el núcleo Linux en hipervisor utilizando un módulo, el cual permite ejecutar máquinas virtuales y otras instancias de sistemas operativos en el espacio de usuario del núcleo Linux anfitrión. Dos soluciones destacan en esta categoría: KVM y User-mode Linux.

Virtualización de recursos: En este segundo caso el recurso que se abstrae es un recurso individual de un computador, como puede ser la conexión a red, el almacenamiento principal y secundario, o la entrada y salida.

Virtualización de aplicaciones: Las aplicaciones son ejecutadas encapsuladas sobre el sistema operativo -recurso usado en este tipo de virtualización- de manera que aunque creen que interactúan con él –y con el hardware- de la manera habitual, en realidad no lo hacen, sino que lo hacen bien con una máquina virtual de aplicación o con algún software de virtualización. Este tipo de virtualización es usada para permitir a las aplicaciones de características como portabilidad o

compatibilidad, por ejemplo para ser ejecutadas en sistemas operativos para los cuales no fueron implementadas.

Virtualización de escritorio: Consiste en la manipulación de forma remota del escritorio de usuario (aplicaciones, archivos, datos), que se encuentra separado de la máquina física, almacenado en un servidor central remoto en lugar de en el disco duro del computador local. El escritorio del usuario es encapsulado y entregado creando máquinas virtuales. De esta forma, es posible permitir al usuario el acceso de forma remota a su escritorio desde múltiples dispositivos, como pueden ser computadores, dispositivos móviles, etc.

2.4.2 Cloud Computing

A continuación abordaremos de manera breve los conceptos inherentes a la Cloud computing, dado que, parte del core de negocio y de los procesos a ser mejorados, están alineados a estos conceptos como servicios a ser entregados a los clientes, por ejemplo, el manejo de Nube pública o Nube privada, el tema Cloud IAAS (Infrastructure as a Service), entre otros. Para nuestro caso particular, la mejora a ser planteada está orientada originalmente al manejo de Nube privada.

El Cloud Computing, es un modelo de entrega de servicio que permite a cualquier persona o empresa acceder a una gran cantidad de recursos de cómputo con la ventaja de poder aumentar o disminuir la cantidad de servicios consumidos, es un término utilizado para describir

a los servicios de cómputo como software y hardware que son entregados como un servicio a través de internet en demanda, proporcionando recursos, software e información compartida a través de equipos de cómputo y otros dispositivos con la característica de ser un servicio medido y escalable. Mientras más recursos se utilicen más se paga por ello [5].

Una definición más ampliada está determinada por el National Institute of Standards and Technology (NIST), define al cómputo en nube como “un modelo que permite un acceso a la red muy cómodo y bajo demanda a un conjunto compartido de recursos de cómputo configurables, como redes, servidores, almacenamiento, aplicaciones y servicios, que pueden aprovisionarse y liberarse rápidamente con un mínimo esfuerzo de gestión o interacción con el proveedor de servicios” [21].

Existen varios modelos de servicios, los cuales son:

- **Software as a Service (SaaS):** en este modelo las aplicaciones son ofrecidas a través de internet, accesible desde un navegador o interfaz. Como estas aplicaciones son entregadas en demanda, pueden ser desplegadas muy rápidamente.

- **Platform as a Service (Paas):** este servicio ofrece una plataforma de desarrollo sin necesidad de adquirir tecnología como OS, bases de

datos, middleware o programas de desarrollo. Además, el hardware y software en este modelo es administrado por el proveedor.

- **Infrastructure as a Service (IaaS):** esta categoría ofrece todos los servicios de infraestructura de propósito general, incluyendo bases de datos, almacenamiento, red y recursos de cómputo. El usuario tiene control total sobre el sistema operativo, almacenamiento del servidor y aplicaciones instaladas.

Además, existen 3 modelos de despliegue de cloud computing:

- **Public Cloud:** la infraestructura es administrada por un proveedor que ofrece servicios como aplicaciones y almacenamiento, a disposición del público en general a través de Internet. Los servicios públicos en la nube pueden ser gratuitos u ofrecidos en un modelo de pago por uso.

- **Private Cloud:** es un tipo de cloud computing que ofrece ventajas similares al public cloud, incluyendo escalabilidad y autoservicio, pero a través de una arquitectura propietaria. A diferencia de la anterior, que ofrecen servicios a múltiples empresas o usuarios, el private cloud está dedicado a una sola organización. Además puede ser construido y administrado por el área de TI de la propia empresa o por un proveedor de cloud.

- **Hybrid Cloud:** es la combinación de dos o más clouds que poder ser privadas, comunitarias o públicas. Al permitir que las cargas de trabajo se desplacen entre nubes privadas y públicas a medida que cambian

las necesidades informáticas y los costos, la nube híbrida ofrece a las empresas una mayor flexibilidad y más opciones de despliegue de datos.

2.5 VMware Cloud Directory

En este último sub capítulo, mencionaremos la tecnología sobre la cual gira la propuesta, dado que, es parte de las herramientas utilizadas en la organización para este tipo de procesos, si bien es cierto, se mencionaron varias tecnologías como referencia en la sección de Virtualización, en este, nos enfocaremos en lo que VMWare nos ofrece como herramienta para virtualizar.

2.5.1 Overview

VMWare

VMWare lanzó su primer hipervisor de servidor en 2001, VMWare ESXi es la capa de virtualización, o hipervisor, que se ejecuta en servidores físicos para abstraer el procesador, la memoria, almacenamiento y recursos en múltiples máquinas virtuales. A medida que la tecnología de virtualización maduraba, se hizo posible virtualizar granjas de servidores físicos [22].

VMWare vSphere

Es un conjunto de tecnologías con muchas funciones que incluye el hipervisor VMware ESXi en su núcleo. VMware vSphere virtualiza y

agrega los recursos físicos subyacentes en varios sistemas y proporciona grupos de recursos virtuales al centro de datos. Esto permite a las organizaciones construir una infraestructura virtual completa con VMware vSphere, escalando a través de un gran número de servidores físicos interconectados y dispositivos de almacenamiento.

Una característica de la suite vSphere llamada Distributed Resource Scheduler (DRS) asigna y equilibra dinámicamente capacidad de computación entre colecciones de recursos de hardware para las máquinas virtuales. Al igual que DRS, vSphere Storage DRS asigna y equilibra dinámicamente la capacidad de almacenamiento y E/S entre colecciones de almacenes de datos. Las características de DRS representan una nueva forma de gestionar los recursos de hardware en el centro de datos. Las organizaciones ya no tienen que asignar estáticamente servidores almacenamiento o recursos de ancho de banda de red a cada aplicación en aplicación en la empresa.

Otra característica importante es *vSphere High Availability (HA)*, que proporciona alta disponibilidad para las máquinas virtuales. Si un servidor falla, las máquinas virtuales afectadas se reinician en otros servidores disponibles que tienen capacidad de sobra.

Una de las características más visibles de la suite vSphere es *vMotion*, en donde, las máquinas virtuales encendidas se migran de un de un

servidor físico a otro sin tiempo de inactividad preservando la integración de la máquina virtual y la continuidad del negocio. Cuando una máquina virtual se migra utilizando vMotion, sólo los archivos de configuración de la CPU y de memoria se mueven y los archivos de disco permanecen en el almacén de datos original.

Por otro lado, la gestión de uno o varios servidores de hipervisor no es necesariamente una tarea de enormes proporciones, sin embargo, cuando el entorno virtualizado se amplía e incorpora cientos de hipervisores, se necesita una gestión centralizada, para esto VMWare utiliza el *vCenter Server* [22].

VMware vCenter Server, es el punto central para la configuración, aprovisionamiento y gestión de entornos virtualizados. El aprovisionamiento de una máquina virtual consiste en asignar recursos de CPU, memoria, almacenamiento y red de un conjunto de recursos virtualizados a través de varios hipervisores. Otro servicio básico relacionado con el aprovisionamiento es el concepto de aplicación virtual o vApp, el cual es un contenedor lógico utilizado para agrupar múltiples máquinas virtuales. Dado que una vApp es un límite lógico para las máquinas virtuales, este puede gestionarse como una entidad única y separada en un entorno virtualizado. Por ejemplo, una vApp puede utilizarse para empaquetar y gestionar una aplicación de varios niveles compuesta por uno o más servidores web frontales, un servidor de aplicaciones y un servidor de base de datos back-end. Las tareas

de gestión, como el encendido y apagado, son aplicables al objeto vApp, además de tareas más avanzadas como clonar o copiar la vApp [22].

VMware vCloud Director

Anteriormente conocido como vCloud Director for Service Providers, VMWare Cloud Director es una plataforma cuya finalidad es operar y administrar la infraestructura y los recursos computacionales virtuales disponibles en la nube. Brinda a sus usuarios la sensación de acceder a centros de cómputos virtuales dedicados. Sus servicios están orientados a providers y a empresas como usuarios finales, llamados tenants.

Las siguientes entidades corresponden al entorno ofrecido a los tenants de VMware Cloud Director [23]:

- **Virtual Data Center:** Ambiente que permite alojar máquinas virtuales, vApps, VMs, templates, etc.
- **Virtual Machine:** Es la unidad básica del Virtual Data Center. Tiene asociadas una determinada capacidad de cómputo, memoria, almacenamiento y acceso a red. Se pueden crear desde templates o nuevas desde cero así como también instalar un sistema operativo desde una imagen ISO.

- **vApp:** Es un contenedor para almacenar VMs que operan juntas para correr una aplicación multicomponente. Los componentes de dicha aplicación se ejecutan en diferentes VMs. Las vApps de VMware suelen agrupar y administrar múltiples VMs que realizan tareas comunes. Se pueden crear templates de vApps para el rápido y sencillo despliegue de vApps y VMs. También es posible tener una vApp con una única VM.
- **Catálogos:** Son usados para almacenar VMs, imágenes ISO de instalación, etc. Los usuarios los pueden utilizar para crear de forma sencilla ambientes con especificaciones e imágenes predefinidas para las nuevas vApps y VMs .
- **Red de VDC - Org VDC network:** es la red de un Virtual Data Center de una organización (o tenant), que está disponible para todas las vApps y VMs. La red del Virtual Data Center de la organización puede estar aislada, sin acceso a internet, o enrutada con acceso a internet.
- **Red de vApp - vApp network:** Es una red que está disponible sólo dentro de una vApp para los componentes (VMs) de la misma. VMs de otras vApps no pueden acceder a la red de esta vApp. Este enfoque provee un nivel de separación adicional entre diferentes vApps. Cada red de vApp tiene su propia puerta de enlace para conectarse a la red del Centro de Procesamiento Virtual (VDC) de la organización.

Un grupo de servidores de VMware Cloud Director está conformado por uno o más servidores instalados en plataforma Linux. Cada servidor del grupo ejecuta un conjunto de servicios denominados la célula de VMware Cloud Director (“cell”, según la documentación del inglés original). Todas las células comparten una única base de datos y almacenamiento del servidor de transferencia (transfer server storage). Cada célula de VMware Cloud Director se conecta a la instancia de vSphere y los recursos de red de los que depende para la parte de infraestructura. Para mayor información sobre la arquitectura, consultar la documentación disponible en la web de referencia [23].

2.5.2 Arquitectura vCloud Director

En resumen, VMware vCloud Director ofrece a las organizaciones empresariales la posibilidad de crear nubes privadas seguras como base para una solución de infraestructura como servicio (IaaS). Junto con VMware vSphere, vCloud Director ofrece computación en la nube para los centros de datos existentes mediante la agrupación de recursos de infraestructura virtual y su entrega a los usuarios como servicios basados en catálogos [24].

Una arquitectura general con detalla la integración de los componentes se detallan a continuación:

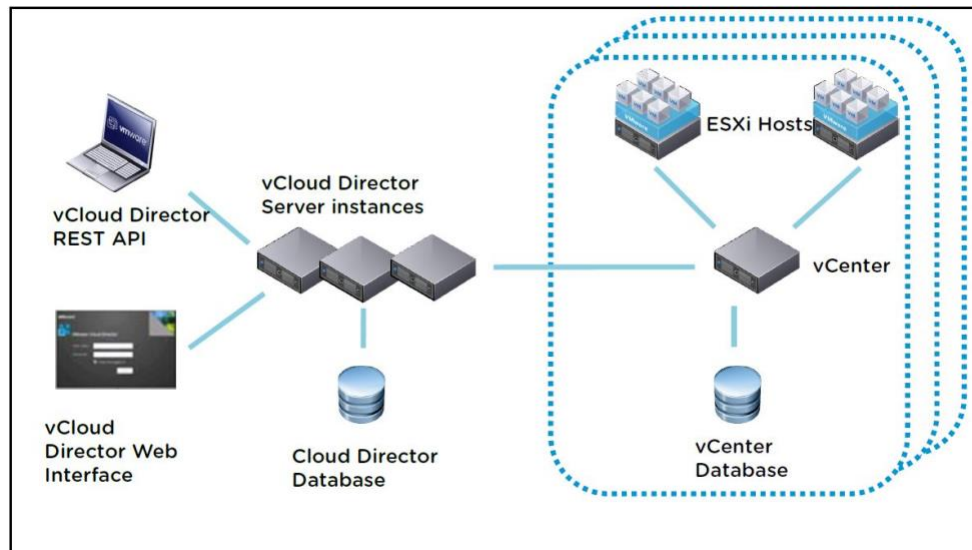


Imagen 2.16 : vCloud Director - Arquitectura general

Fuente : Diseño general vCloud Director [24]

La solución está compuesta por los siguientes componentes:

- Instancias o células del servidor de vCloud Director
- Base de datos de vCloud Director
- Uno o más entornos VMware vSphere compuestos por:
 - Uno o más hosts VMware ESXi
 - Uno o más servidores VMware vCenter
 - Base de datos del servidor vCenter
- Interfaces de clientes:
 - vCloud Director REST API
 - Interfaz web de vCloud Director

Esta configuración de vCloud Director permite la creación de vCloud Virtual Datacenters (vDC), que es simplemente una colección de recursos como redes, almacenamiento, CPU y memoria.

2.5.3 Integración vía vCloud Directory APIs

La API permite a los desarrolladores de software de terceros producir aplicaciones que funcionen con cualquier proveedor de nube que la soporte. Los usuarios finales pueden crear nuevas herramientas e interfaces para manipular los objetos de la nube disponibles, mientras que los administradores pueden programar la funcionalidad de despliegue y control para simplificar sus operaciones rutinarias. Las interacciones de la API también pueden integrarse en aplicaciones, lo que permite a los sistemas conscientes de la nube manipular la plataforma en la que se ejecutan. Por ejemplo, un servicio web podría detectar un pico de tráfico y desplegar automáticamente nuevos recursos para compensar. Así, la API es fundamental para el autoescalado PaaS o SaaS sobre la oferta IaaS.

La API de vCloud está estrechamente vinculada a las abstracciones de la nube permitiendo a los usuarios manipular programáticamente las entidades de la misma. Por lo tanto, hereda muchas de las propiedades deseables que existen en el modelo: es puramente virtual y sencilla de utilizar, a la vez que admite aplicaciones heredadas y usuarios de la

empresa. Su alcance se limita a la funcionalidad que necesitan los consumidores de recursos en la nube, en lugar de los productores.

Como se mencionó, la comunicación API REST termina siendo la indicada que permita trabajar con las APIs que se exponen dentro de la plataforma, dado por su flexibilidad y acople con infraestructuras web [10].

A continuación se muestra una lista de operaciones básicas que vCloud ofrece para consumo vía REST:

vApp operations
POST <vapp-uri>/action/{deploy, undeploy}
POST <vapp-uri>/power/action/{powerOn, powerOff}
POST <vapp-uri>/power/action/{reset, suspend}
POST <vapp-uri>/power/action/{shutdown, reboot}
GET <vapp-uri>/screen
POST <vapp-uri>/screen/action/acquireTicket
vApp Configuration Operations
POST <vapp-parent-element-uri>
DELETE <vapp-element-uri>
PUT <vapp-element-uri>
Inventory Listing
GET <vapp-uri>
GET <vdc-uri>
GET <vAppTemplate-uri>
GET <media-uri>
GET <network-uri>
Catalog Management
GET <catalog-uri>
POST <catalog-uri>/catalogItems
Upload/Download/Provisioning Operations
POST <vdc-uri>/action/composeVApp
POST <vdc-uri>/action/instantiateVAppTemplate
POST <vdc-uri>/action/instantiateOvf
POST <vdc-uri>/action/annotate
POST <vdc-uri>/action/uploadVAppTemplate
POST <vdc-uri>/media
PUT <upload-uri>
GET <download-uri>
DELETE <resourceEntity-uri>
Task Management
GET <tasks-list-uri>
GET <task-uri>
POST <task-uri>/action/cancel

Imagen 2.17 : Operaciones básicas vCloud API

Fuente : vCloud API , Operaciones básicas [10]

Adicional también se detalla las API básicas que el componente que vSphere nos entrega como una unidad más particular [25]:

Método	Api path	Descripción
GET	https://<host>/api/vcenter/folder	Api encargado de obtener la carpeta/ruta en donde se almacena la nueva máquina virtual
GET	https://<host>/api/vcenter/cluster/{cluster}	Api encargado de entregar el Cluster en donde se obtiene los recursos de aprovisionamiento
GET	https://<host>/api/vcenter/datastore	Api encargado de obtener el datastore de donde validarán las capacidades
GET	https://<host>/api/vcenter/datastore/{datastore}	Api encargado de obtener el datastore dado uno particular para obtener mayor detalle
GET	https://<host>/api/vcenter/resource-pool	Api encargado de obtener el pool de recursos disponibles para aprovisionamiento
GET	https://<host>/api/vcenter/resource-pool/{resource}	Api encargado de obtener el pool de recursos dado uno particular para obtener detalle
POST	https://<host>/api/vcenter/host	Api encargado de crear el host que se vinculará a una nueva máquina virtual
POST	https://<host>/api/vcenter/vm	Api encargado de crear la nueva máquina virtual, asignación de nombre y SO
POST	https://<host>/api/vcenter/vm/{vm}?action=relocate	Api encargado de asignar a la máquina virtual los detalles técnicos necesarios para el despliegue

Tabla 3 : Definición de API básica provista por vSphere

Fuente: Autor/VMware

La API permite a los usuarios instanciar, configurar y controlar vApps de forma programada ó de interactuar directamente sobre la implementación de las máquinas a nivel más técnico (vSphere), así como consultar el inventario de una organización. También permite la gestión de catálogos y la supervisión de tareas asíncronas de larga duración que se ejecutan en nombre de un cliente (como el encendido de una vApp). Y enfocando en lo que vSphere nos brinda como una unidad importante de la arquitectura, tomando como referencia la *Tabla 3*, también podemos complementar la implementación con estas

capacidades y es sobre esto, que para efecto concreto, enfocaremos de manera técnica el cómo trabajar la automatización planteada.

CAPÍTULO 3

LEVANTAMIENTO Y ANÁLISIS DEL PROCESO DE HOSTING ACTUAL

En el siguiente capítulo se abarcará todo lo concerniente al Servicio de HOSTING que la empresa provee, detallando desde las características del mismo y cómo está correlacionado con otros servicios, cuáles son las etapas de implementación y su análisis, los stakeholders, involucrados, áreas de gestión y el detalle del tiempo efectivo que este proceso requiere para su cumplimiento, para al final detallar el proceso AS-IS del mismo.

3.1 Características de requerimientos de Hosting

Actualmente, para poder brindar el servicios de HOSTING respectivamente relacionados con la configuración, despliegue y entrega máquinas virtuales, se debe tomar en cuenta ciertas reglas de negocio que la empresa tiene declaradas para que el servicio sea entregado de manera completa además de insumos necesarios para dar inicio al proceso.

Cada que un cliente levanta la necesidad, un primer insumo, se lo conoce como Anexo Técnico, este documento conglera toda la necesidad del cliente, en donde, se define qué conjunto de servicios se requieren para satisfacer su necesidad además del relacionado a las máquinas virtuales.

Cuando nos referimos a las reglas de negocio, esto hace referencia a que, por definición, cada servicio de HOSTING debe venir acompañado por un conjunto de servicios complementarios de diferentes verticales del negocio, por ejemplo, la entrega de una máquina virtual NO podría ser funcionar si un servicio de Internet o Datos que permita que lo que se instale dentro de ella tenga completa funcionalidad, también debe venir acompañado de un conjunto de licencias en función del Sistema Operativo que sea configurado en la misma y que tendrán sus reglas acorde a cada proveedor y otros servicios adicionales complementarios, esta conjunción de Servicios es llamado como **SOLUCIÓN**, la cual es gestionada de manera íntegra dentro del flujo de aprovisionamiento y es el término que iremos utilizando comúnmente.

A continuación se muestra un detalle del cómo una Solución es estructurada una vez que un Cliente, en conjunto con el asesor comercial, llegan a un acuerdo respecto a los servicios a ser contratados (los datos presentados, son referencia):

Solución	#123456
Cliente	Cliente ABC
Servicio	Línea de Negocio
Cloud IAAS	CLOUD
Internet	COMUNICACIONES
Firewall	SEGURIDAD LÓGICA
BaaS	CLOUD
Cloud Mail	CLOUD
Licenciamiento	CLOUD

Tabla 4 : Solución de servicios para Cliente ABC

Fuente : Autor

La Tabla 3 mostrada, nos muestra un ejemplo del cómo pueden estar construidas las soluciones, en donde, el Servicio Cloud IAAS es nuestro servicio de enfoque, el cual hace referencia a las máquinas virtuales, de ahí, acorde a la necesidad, pueden agregarse servicios complementarios como Firewall, Backup, mail, etc. Como se observa, los 3 servicios marcados con negrilla son parte de las reglas de negocio establecidas, en donde, nos indican que un servicio de máquinas virtuales debe venir acompañado con un servicio de Internet y con las respectivas licencias. En definitiva, siempre dependerá de cada cliente el cómo será armada su respectiva solución en función de las capacidades y recursos que se encuentren disponibles.

Una vez explicado a manera general el qué es y cómo se compone una Solución, vamos a enfocarnos en el servicio que hace referencia a las máquinas virtuales (Cloud IAAS). Básicamente, este servicio tiene algunas características técnicas necesarias para su implementación, lo que el cliente, en su etapa de requerimiento o necesidad, lo que solicita es una cantidad de recursos que en su todo, es llamado Pool de Recursos. Este Pool esta compuesto por una cantidad de DISCO, MEMORIA y PROCESADOR, capacidades necesarias que se podrán distribuir en una o varias máquinas virtuales, luego, de acuerdo a la cantidad requerida de estos recursos, se establece el precio acorde a una función para cálculo de precio determinado por los diferentes tipos de recursos disponibles para cada característica mencionada, por ejemplo:

- DISCO : Rápido/Ultra rápido/etc
- MEMORIA: Vmware/Oracle/etc
- PROCESADOR: Vmware/RHEV/etc

Entonces, de acuerdo a estos valores, el número de recursos necesitado y en la suma de los servicios complementarios, se establecen los valores a ser facturados al cliente, esta es la forma en la que se configura en base a sus características cada servicio.

Otro punto a resaltar y que es parte de las configuraciones, es que, una vez segmentados los recursos en una o varias máquinas, es la selección de las

licencias acorde al Sistema Operativo o aplicación a ser instalado, cumpliendo las reglas que los proveedores de estas licencias definen.

A continuación, una muestra del cómo se distribuyen los recursos y cómo estos son asignados a N máquinas virtuales y que al final serán nuestro insumo principal para la implementación de los procesos de automatización:

POOL DE RECURSOS (Cloud IAAS)	
Solución	#123456
Cliente	Cliente ABC
Recursos	Cantidad
DISCO	500 GB
MEMORIA	24 GB
PROCESADOR	24 GB

Tabla 5 : Asignación de Recursos al POOL

Fuente : Autor

Una vez asignados los recursos al Pool, mostraremos cómo estos pueden ser distribuidos a las máquinas virtuales que puedan requerirse:

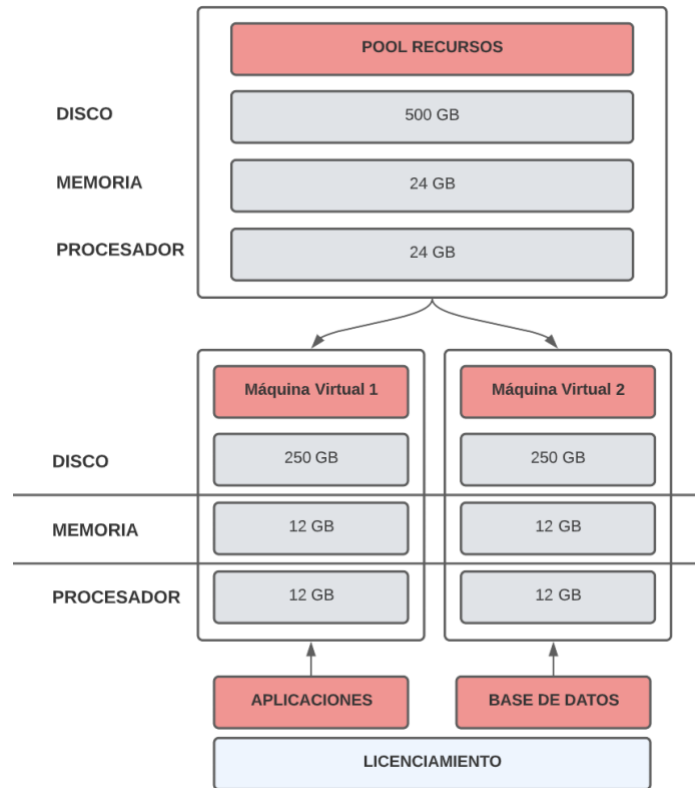


Imagen 3.1 : Segmentación de recursos del POOL

Fuente : Autor

Como se observa en la imagen, a partir del POOL asignado, el cliente puede tomar decisión de cuántas máquinas virtuales crear y qué tipo de tecnología configurar en cada una de ellas.

3.2 Etapas de implementación de servicios de Hosting

En este sub capítulo se detallan las diferentes etapas por las que la Solución va siendo gestionada por parte de las diferentes áreas o departamentos de manera muy general.

3.2.1 Matriz RACI

A continuación, detallaremos los diferentes actores y responsabilidades que permitan entender más adelante cómo intervienen en el flujo de servicio.

Primero, se establece la definición de la terminología que debe ser entendida respecto a la matriz de responsabilidades:

- **R** : Responsable
- **A** : Aprobador
- **C** : Consultado
- **I** : Informado

Luego, detallaremos los actores que actualmente interviene en el flujo de aprovisionamiento, estos son:

Roles	Área/Departamento
Asesores Comerciales	Comercial
Asistentes comerciales	Comercial
Equipo de Infraestructura	Técnico
Equipos de Operaciones/Soporte	Técnico

Tabla 6 : Roles o actores del flujo

Fuente : Autor

Como vemos en la tabla, estos son las áreas que generalmente interactúan dentro del flujo, en donde, los Asesores comerciales son quienes atienden de primera mano la necesidad del cliente, los

asistentes se encargan básicamente de continuar con el proceso dentro del sistema y darle inicio al flujo, el equipo de TI enfocado en la validación de recursos lógicos y asignación de los mismos en base al requerimiento y el equipo de Soporte u operaciones se encarga de, sobre los recursos asignados, generar el despliegue de las máquinas virtuales y completar lo que se define en inicio entre el cliente y el Asesor.

Una vez explicado los actores, mostraremos las actividades generales y sus responsabilidades con la ayuda de matriz RACI, a continuación:

Matriz RACI (Cloud IAAS)	Roles			
	Asesores Comerciales	Equipo de IT	Equipo de Operaciones	Asistentes Comerciales
Actividades				
Generación Anexo Técnico	R			I
Ingreso Orden Servicio	C			R
Asignación de Recursos Lógicos	I	R,A		C
Generación de Máquinas Virtuales			R,A	
Activación y entrega de Servicio	I		R,A	

Imagen 3.2 : Matriz RACI flujo de Cloud IAAS

Fuente : Autor

En la imagen anterior, se muestran los roles previamente explicados y sus responsabilidades para tener una visión de la interacción y soporte entre las áreas definidas.

3.2.2 Análisis de las etapas de implementación

Analizaremos una vez conocidos los actores del flujo, cuales son las diferentes etapas por la que se desenvuelven las actividades para cumplir con la implementación requerida:

Código	Actividades/Procesos
A1	Generación de Anexo Técnico
A2	Generación de Orden de Servicio
A3	Solicitar Factibilidad técnica
A4	Revisión de anexo técnico
A5	Verificación de disponibilidad de recursos
A6	Asignar recursos lógicos
A7	Generar tarea de implementación
A8	Implementación de máquinas virtuales
A9	Planificar entrega y activación de servicio

Tabla 7 : Actividades para implementación de Servicio de Hosting

Fuente : Autor

Estas actividades son básicamente las etapas para el Servicio de Cloud IAAS, en donde:

- **Generación de Anexo Técnico:** es donde se ingresan los requerimientos con los aspectos técnicos, esto es un proceso entre el Asesor Comercial y el cliente.
- **Generación de orden de servicio:** es el proceso sobre le cual la información es ingresada al sistema ERP de la empresa y

posteriormente se da inicio a los flujos, el objeto de negocio esperado en este punto es el Anexo Técnico.

- **Solicitud Factibilidad técnica:** es la validación de disponibilidad de recursos lógicos que soporten la configuración del servicio.
- **Revisión de anexo técnico:** es la verificación de que la información ingresada en el documento sea consistente antes de hacer la verificación de inventario lógico.
- **Verificación de disponibilidad de recursos:** es la validación de recursos disponibles acorde la necesidad del cliente ingresada en el anexo técnico, esta verificación es realizada contra la infraestructura cloud existente.
- **Asignar recursos lógicos:** es la etapa en donde el departamento de Infraestructura se encarga de validar disponibilidad y acorde a eso asignar los recursos lógicos sobre el cual se realizarán las implementaciones de las máquinas virtuales.
- **Generar tarea de implementación:** una vez asignado recursos lógicos en donde se realizará el aprovisionamiento, se crea una tarea de forma manual hacia el área operativa y de soporte para la realización de su trabajo.
- **Implementación de máquinas virtuales:** en esta etapa, una vez definido los recursos lógicos, se realizará el despliegue de cada una de las máquinas virtuales con su respectivo sistema operativo y demás características requeridas, como aplicaciones, licencias, etc.

- **Planificar entrega y activación de servicio:** está enfocado en la verificación de lo implementado en conjunto con los demás servicios complementarios, por ejemplo, Internet o Datos, para darle entrega al cliente posterior a la planificación de la misma para finalmente cambiar el estado del servicio en el sistema.

En el modelo AS-IS se detallará de mejor manera la línea de flujo e integración de las cada etapa.

Por último, recordar que, parte del problema a querer ser resuelto, es la existencia de muchos actores para un flujo que puede ser mejorado y mucho más ágil, en el siguiente sub capítulo se analizará en función de tiempos como esto puede llegar a impactar en la percepción de calidad de servicio de los clientes.

3.2.3 Análisis de Tiempos de gestión de las etapas

Conociendo entonces, actores y sus actividades, a continuación nos enfocaremos en el análisis cuantitativo de las falencias del flujo en función de los tiempos que toman las etapas, sobre todo las que tienen que ver con la implementación de las máquinas virtuales.

Para empezar este análisis nos remitiremos a la Tabla 1, en donde tenemos las variables que usaremos la medición requerida. Básicamente el enfoque es, validar los tiempos de aprovisionamiento que sería aquel entre que se genera la orden de servicio y se asigna recursos lógicos (Factibilidad técnica) y el tiempo de configuración y

entrega, que es el comprendido entre que se generó la factibilidad y la activación del servicio.

Para poder obtener la información, nos basaremos en la información recopilada desde que en el Sistema empezó a registrar la trazabilidad del proceso en una primera etapa de automatización. Por qué se menciona como primera etapa? Porque antes de ese punto, la información no se registraba en el sistema de forma adecuada y toda gestión se realizaba por fuera, en archivos de Excel o simplemente en correos de autorización, sin embargo, a pesar de empezar a automatizar estos procesos, aun tiene falencias en la gestión como ya se especificó en el capítulo 1.

Las estadísticas a ser mostradas son referente a datos capturados desde el 2018 hasta el 2022, cabe recalcar que este producto no tiene una alta demanda, sin embargo es parte del Core principal del negocio del Data Center de la empresa, pero a su vez, tiene alta proyección respecto a la venta.

Primero se mostrará todos los segmentos de tiempo involucrados en el flujo para que a partir de ahí, determinemos cuáles son aquellos tiempos que nos ayudarán con nuestras variables de análisis. A continuación la descripción de estos tiempos:

Tipo	Descripción del intervalo de tiempo
T1	Tiempo en que se ingresa la orden y se solicita Factibilidad Técnica
T2	Tiempo entre que se solicita Factibilidad Técnica y se asigna recursos lógicos
T3	Tiempo entre que se asignó recursos lógicos y se planifica asignación de tarea de implementación
T4	Tiempo entre que se Planifica el trabajo implementación y se asigna el trabajo
T5	Tiempo entre que se implementa la Máquina virtual y se entrega y activa el servicio al cliente

Tabla 8 : Definición de la descripción de intervalos de tiempos de análisis

Fuente : Autor

A continuación, se muestra la tabulación con todos los tiempos descritos en el intervalo de tiempo mencionado, los tiempos están dados en minutos:

AÑO	Tiempo en Minutos				
	T1	T2	T3	T4	T5
2018	907,62	998,84	0,00	0,00	189235,28
2019	243,28	3138,33	3783,78	31722,97	121796,25
2020	587,41	8162,89	6742,06	2617,22	64855,16
2021	1272,30	4660,31	881,48	1460,92	21220,89
2022	492,25	1783,87	1086,13	2021,36	16582,03

Tabla 9 : Tabulación del tiempo promedio en minutos de cada etapa (2018-2022)

Fuente : Autor

Se muestra una gráfica respecto a los datos tabulados:



Imagen 3.3 : Gráfico de barras de tiempos de flujo por años

Fuente : Autor

En la siguiente tabla se muestra un consolidado promediado de los 5 años de referencia, con los tiempos en minutos y en días para poder realizar el análisis concreto, en donde como detalle no menor, desde que se empezó a usar el sistema, se ha ido mejorando la etapa de implementación y posterior entrega del servicio a los clientes aunque aun así, los tiempos no resultan óptimos en el consolidado.

UNIDAD DE TIEMPO	T1	T2	T3	T4	T5
Minutos	700,57	3748,85	2498,69	7564,49	82737,92
Días	0,49	2,60	1,74	5,25	57,46

Tabla 10 : Consolidado (5 años) de tiempo promedio en minutos y días

Fuente : Autor

Como observamos, el mayor tiempo se invierte en la etapa de implementación de las máquinas virtuales y en la posterior entrega y activación del servicio a los clientes, tal cual se especifica en la descripción del problema. Ahora, hagamos la relación con las variables de medición especificadas en la Tabla 1, en donde el *Tiempo de aprovisionamiento* corresponde a el **T2** y el *Tiempo de implementación y entrega* corresponde al **T5**.

Por tanto, podemos concluir que, la etapa de aprovisionamiento de recursos lógicos demora alrededor de 3 días lo cual se vuelve un tiempo crítico dado que sólo se trata de asignar aspectos de virtualización a utilizar y validar disponibilidades sobre la plataforma y la etapa más crítica, la de implementación de la máquina virtual, configuración, instalación y posterior entrega al cliente puede tomar hasta 2 meses (en el escenario más crítico), lo cual, evidentemente se traduce en un malestar para el cliente.

En conclusión, este será el punto de partida y de comparación que permitirá tener como referencia para evaluar la eficiencia que la mejora del proceso tendría.

El objetivo en este punto es que la asignación de Factibilidad técnica y el aprovisionamiento sea un proceso automatizado que me reduzca considerablemente el tiempo de gestión, de un promedio máximo de 2 meses en el caso más extremo a un proceso que demore minutos, dejando el resto de la gestión a cargo de los departamentos mencionados sólo para la parte de gestión y comunicación frente a cada cliente.

3.3 Modelo AS-IS del proceso actual

En esta sección ya conociendo la generalidad del proceso, vamos a mostrar el diagrama que representa el flujo actual, este proceso está basado en un consolidado de información provista por el departamento de Procesos para este tipo de servicio, agregando, la parte de inicio a todo el flujo para que se entienda a detalle el ciclo de vida del proceso.

A continuación, se muestra el diagrama de procesos mencionado:

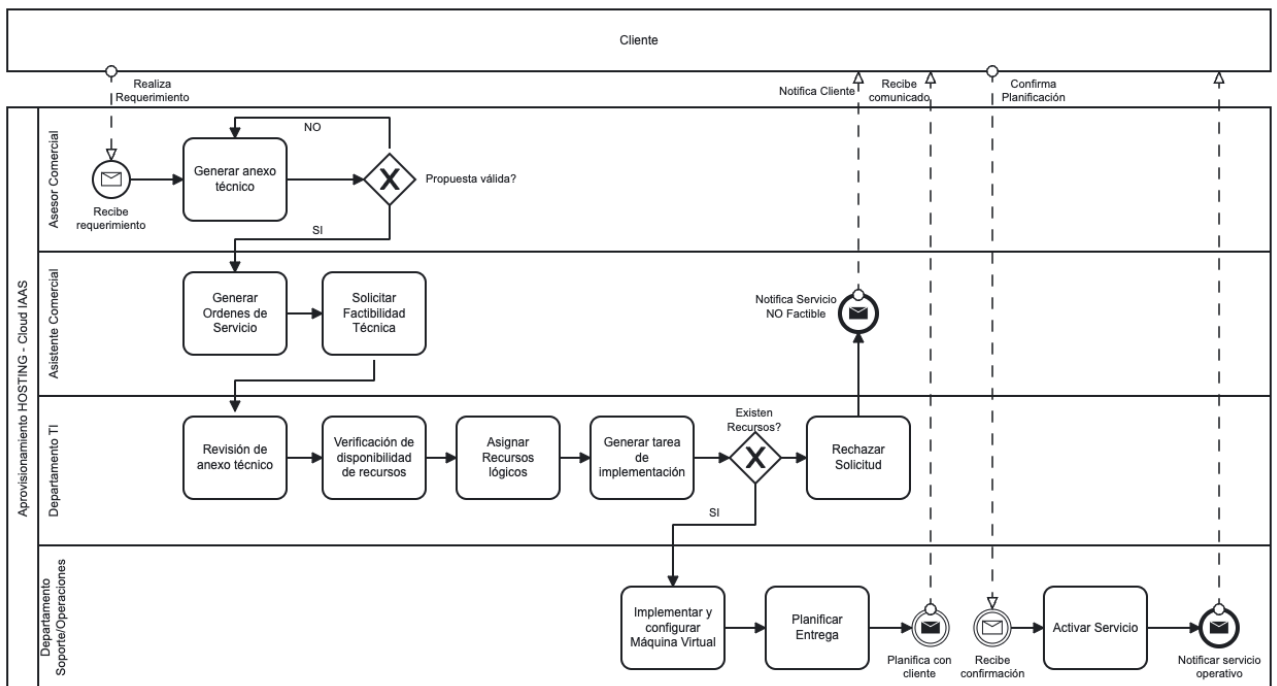


Imagen 3.4 : Modelo AS-IS Flujo Aprovisionamiento

Fuente : Autor

Como se ve en la imagen Imagen 3.4, se detallan las cuatro áreas contempladas como actores dentro del flujo de aprovisionamiento, observamos al cliente como un actor transversal a todo el flujo dado que este, es quien es informado por cada etapa, desde la solicitud del requerimiento, la planificación hasta la confirmación y entrega del Servicio.

Así terminamos de definir cómo el proceso funciona a hoy, se detalló y explicó las etapas, actividades y el problema resultante de estas múltiples interacciones en función del tiempo de entrega.

Cabe recalcar que para poder validar y proponer la oportunidad de mejora para este proyecto, el cual, se verá reflejado en el siguiente capítulo,

manejaremos un alcance que estará basado en la posibilidad de disponer de los recursos técnicos y lógicos que permitan poder validar y representar funcionalmente la solución. Para esto, no usaremos directamente el API ni la plataforma de vCloud Director, dado que, por temas de licencia y disponibilidad de recursos, actualmente, esta sólo está hábil para la operación de la empresa, por tanto, se plantea poder utilizar parte del Stack tecnológico que la solución de VMWare nos provee y la cual se mencionó dentro del marco teórico, esto es, usar la definiciones que VMWare vSphere dispone, para poder simular la como el proceso de aprovisionamiento se comportaría, dado que este es parte de la arquitectura a un nivel más interno y que al final, nos provee también una capacidad de configuración con igual resultados que vCloud Director, solo, que este último se encuentra más arriba de todo el ecosistema y que permite abstraer todos los recursos de infraestructura. Para el efecto práctico de la prueba de concepto a ser realizada más adelante, se realizará un Mock (Simulación de consumo) con las definiciones del API de vSphere, dado que, al igual que vCloud Director, no se tiene disponibilizado un ambiente real, pero las APIs que posee nos ayudarán a entender mejor el comportamiento esperado.

CAPÍTULO 4

DISEÑO DE LA SOLUCIÓN

El siguiente capítulo, detallaremos la solución que permitirá poder resolver el problema planteado en esta tesis, empazando por el modelamiento del proceso TO-BE mencionando la abstracción que deberíamos tener para poder agilizar el proceso en base a la integración de componentes de Software de manera automática. Luego, ya se detalla los diferentes diagramas de Arquitectura de Software, en donde, se planteará la propuesta de implementación de toda la solución a nivel de componentes generales, el cual nos dará la guía del cómo debe realizarse esta transformación digital y finalmente un diagrama de secuencia que permitirá ver las interacciones directas entre los componentes a ser creados o consumidos según los casos de uso más comunes.

4.1 Modelo TO-BE para el flujo de aprovisionamiento y configuración

Una vez establecido el punto de partida con el modelo AS-IS referenciado en la Imagen 3.4, a continuación se detalla el proceso de mejora que involucra la integración del Sistema ERP con la plataforma que VMWare nos provee, tomar en cuenta que, el flujo será muy similar en alguna de sus etapas actividades iniciales, sobre todo en la planificación y entrega y verificación con el cliente, dado que al final se requiere validar y comprobar aspectos del SLA (Service Level Agreement) del servicio.

Primero referenciaremos las actividades que se agregarán en esta mejora del proceso, para lo cual nos basaremos en el detalle descrito en el capítulo 3, en la Tabla 7:

Código	Actividades/Procesos	Tipo
A1	Generación de Anexo Técnico	M
A2	Generación de Orden de Servicio	M
A3	Solicitar Factibilidad técnica	M
A4	Validar disponibilidad y capacidades	A
A5	Crear máquinas virtuales	A
A6	Asignar recursos lógicos	M
A7	Implementación de máquinas virtuales	M
A8	Planificar entrega y activación de servicio	M

Tabla 11 : Detalle de Actividades para el proceso de mejora

Fuente : Autor

A diferencia de las actividades detalladas en el status inicial, en esta instancia, si observamos, se agregó un campo llamado "Tipo", este básicamente nos

ayuda a diferenciar si las actividades son manuales o son automatizaciones y además de eso se agregaron 2 actividades adicionales, que son las A4 y A5, que pueden resultar redundantes con las existentes, sin embargo, nos indica que estas serán realizadas por un proceso automático y reemplazan aquellas tareas manuales especificadas en la **Tabla 7**, específicamente sustituyen las tareas A4, A5, A6 y A7 y la razón por la cual parecen repetidas, es por las características y definiciones a nivel de operación, en donde, si la integración automática no es capaz de dar un resultado positivo, el usuario debe realizar la validación particular en donde pueda tomar decisión en base a un análisis de la infraestructura y si al final, no se logra asignar recursos, se procede a indicar al cliente la factibilidad rechazada de su requerimiento. A continuación, el detalle de lo mencionado:

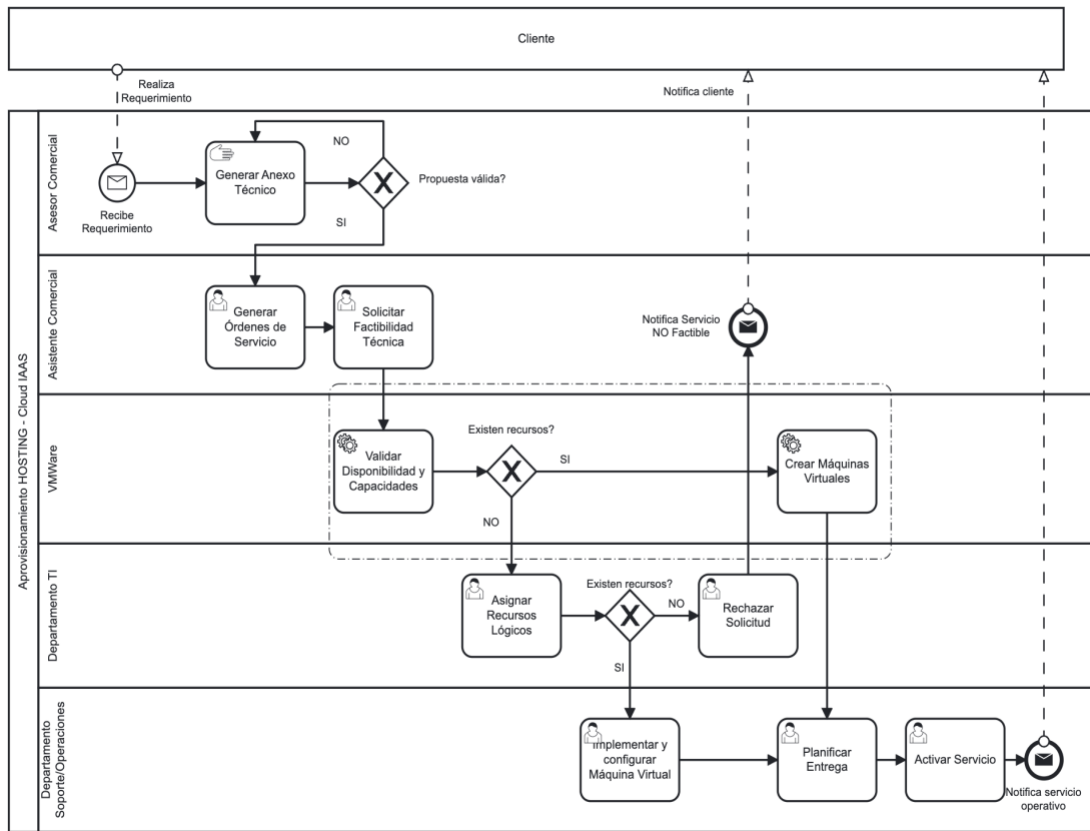


Imagen 4.1 : Modelo TO-BE de la Solución

Fuente : Autor

Como se observa en la imagen y a diferencia del diagrama de status inicial, se agregó un segmento como actor vinculado a *VMWare* a muy alto nivel, en donde, se agregan dos tareas automáticas, la una para *Validar la disponibilidad de recursos lógicos* dentro de la infraestructura y la segunda actividad, en caso de confirmarse que es factible continuar, que se encargaría de *Implementar la máquina virtual* de forma automática, este proceso es completamente autónomo y luego de que se realiza su configuración notificará a la siguiente tarea de planificación y entrega final del servicio al cliente. Todo

esta interacción si nos damos cuenta, termina siendo un proceso híbrido entre la gestión humana y la automatización.

Este proceso deberá integrarse con nuevos Microservicios cuyos tasks estarán enfocados en la comunicación vía API REST con la plataforma de aprovisionamiento dentro del vCenter que es al final, donde las máquinas virtuales son creadas, alojadas y administradas.

Cabe recordar que, nos ayudaremos del API que despliegue y validación que vSphere nos provee y que son útiles para el aprovisionamiento requerido, de acuerdo al escenario o necesidad como pasos básicos para las pruebas e integraciones.

Adicional, cabe mencionar que el flujo tiene limitantes establecidas, esto es, si definitivamente se realiza la validación automática con el inventario lógico o se la realiza de forma manual y al final, no se encuentran capacidades, se deberá notificar al cliente que su requerimiento eventualmente no podrá ser atendido hasta que el inventario haya sufrido cambios y sea complementado por el área encargada. Esto se traduce en un proceso de compra de recursos para el Data Center.

4.2 Diseño arquitectónico de componentes de las solución

En este sub capítulo hablaremos y detallaremos los diagramas UML (Unified Modeling Language) necesarios para poder establecer la Arquitectura de Software requerida para planificar la implementación de la solución planteada en el proceso TO BE.

Para el diseño de arquitectura nos ayudaremos del diagrama de componentes y el diagrama de secuencia, este último, a ser revisado en el siguiente sub capítulo.

Como información previa, el diagrama de componentes de Software es aquel diagrama en donde se detallan todos los artefactos involucrados dentro de una solución, estos pueden ser: microservicios, aplicaciones, canales de comunicación, integraciones con vendors o proveedores, bases de datos, etc. Estos componentes integrados y referenciados entre sí, entrega un overview general del cómo debe funcionar una determinada solución.

Primero enlistaremos los artefactos intervinientes en la solución de este problema:

Componente	Descripción
Sistema ERP	Sistema de gestión que mantiene los flujos de productos en la pre-venta y post-venta
Camunda	Orquestador de actividades basada en microservicios
ms-factibilidad-lógica	Microservicio encargado de validar dentro de la infraestructura, la disponibilidad de recursos lógicos.
ms-aprovisionamiento	Microservicios que se encargará de la integración contra la plataforma de provisioning
VMWare	Plataforma de aprovisionamiento, cuyo producto es el vCloud Director
Base de Datos	Almacenamiento de la información de los clientes y productos
Canales	Medio por el cual el usuario interactúa con el flujo

Tabla 12 : Componentes de Software de la solución

Fuente : Autor

Los componentes detallados en la

Tabla 12, son los que mostrarán a nivel general, cómo sería el ecosistema de la solución, a medida que esta vaya creciendo se irán agregando y optimizando artefactos en el camino acorde la demanda de la operación. A continuación el diagrama de la solución:

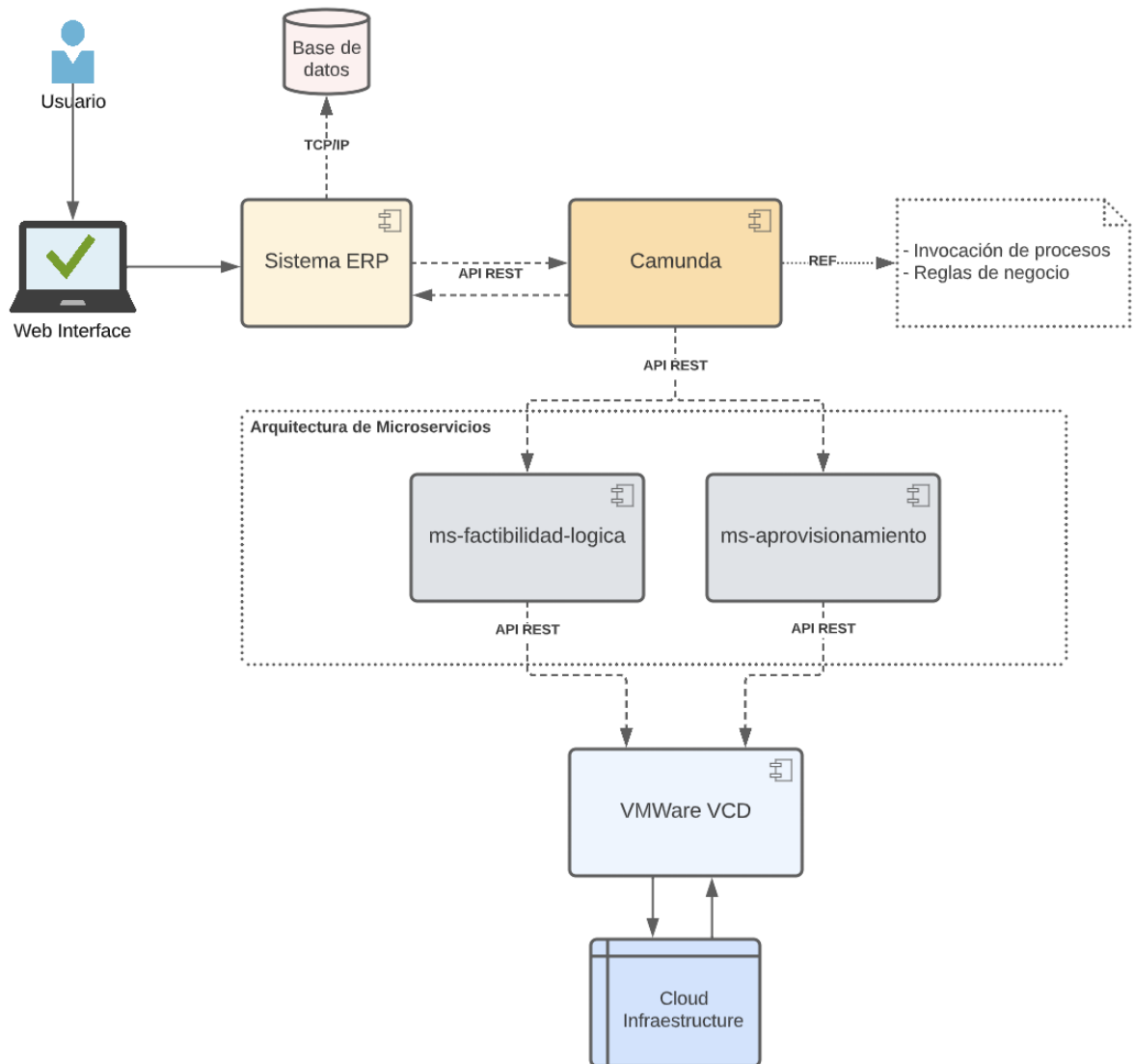


Imagen 4.2 : Diagrama de componentes de Software

Fuente : Autor

Como se observa en la imagen, se describe claramente los componentes mencionados, en donde, el Sistema ERP es el punto de entrada del flujo por parte del usuario, en este caso las Asistentes Comerciales, luego tenemos a Camunda que se encargará de orquestar aquellos microservicios encargados de los aspectos de automatización e integración con la plataforma de VMWare, básicamente, al final de la interacción de todos estos componentes, siempre el ERP será quien reciba las respuestas necesarias en los diferentes escenarios para que el usuario pueda seguir dando gestión al flujo.

En el siguiente sub capítulo, se detallará el orden de los procesos por cada etapa en el diagrama de secuencias correspondiente.

4.3 Diagramas de secuencias de la solución

En esta última sección detallaremos los casos de uso más comunes que se pudiesen presentar en el flujo representados.

Para poder detallar los casos de uso, nos ayudaremos de las actividades principales que se dan para el proceso, estas se encuentran referenciadas en la Tabla 11.

Primero mostraremos un gráfico general de estos casos y luego los iremos detallando.

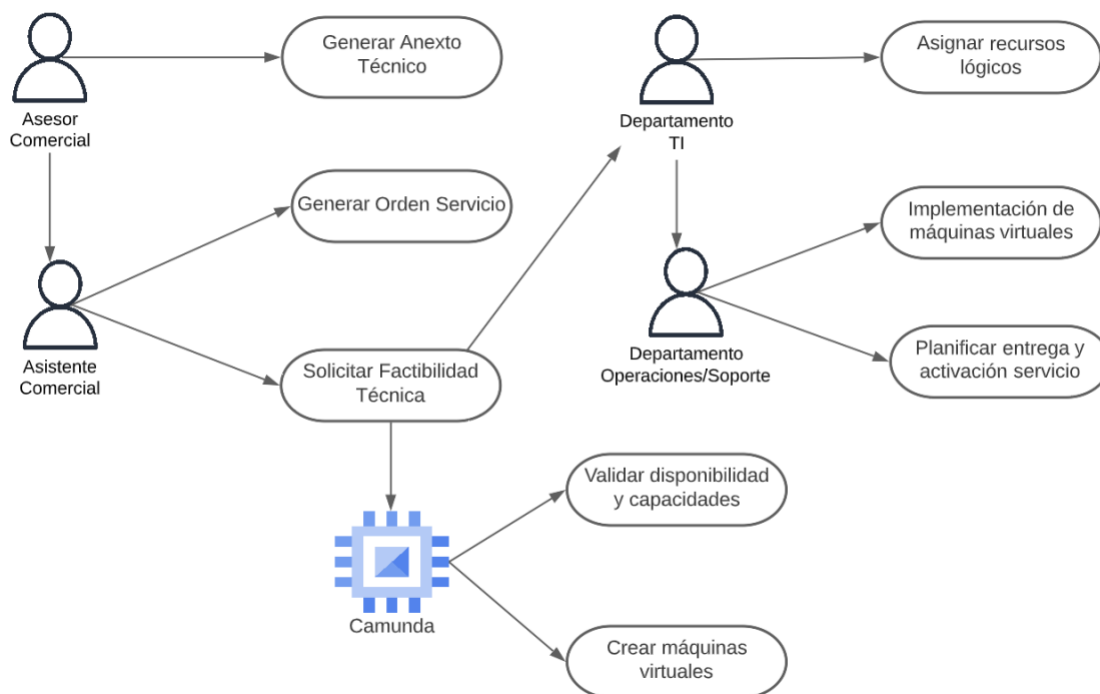


Imagen 4.3 : Vista general de casos de usos y actores

Fuente : Autor

Como nos damos cuenta, Camunda está especificado como un engine encargado de las dos actividades automáticas y el resto de actores con sus actividades específicas.

Ahora, se presentará el diagrama de secuencia general de toda la solución, en donde se podrán observar las interacciones de los diferentes procesos, pudiendo validar posibles escenarios, controles y demás situaciones que permita al equipo encargado de la implementación entender el cómo debe validarse el flujo. De ser posible más adelante, cada caso de uso tendrá también su diagrama de secuencia particular en función de las reglas de negocio dictadas por la operación.

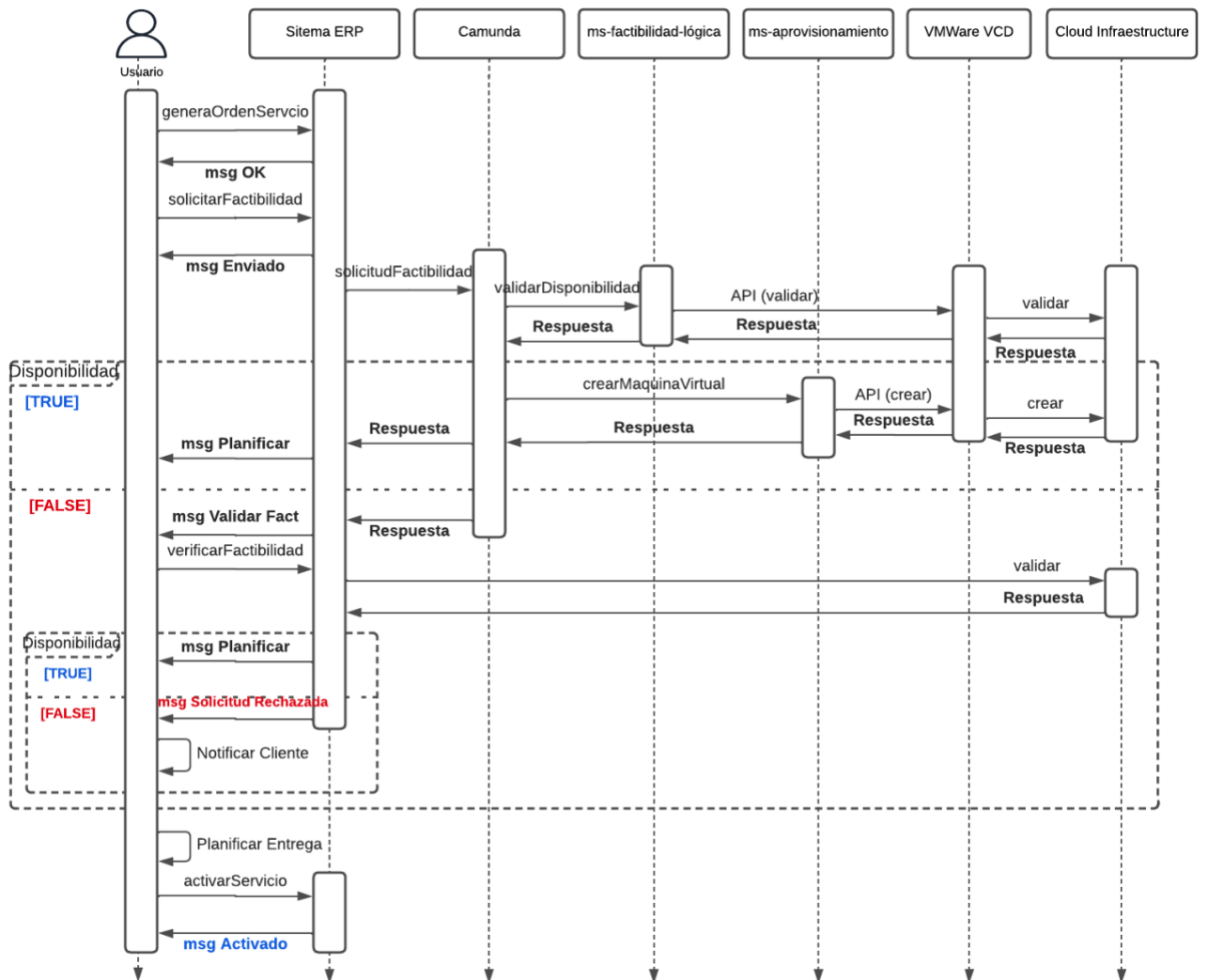


Imagen 4.4 : Diagrama de secuencia general

Fuente : Autor

La imagen nos muestra una vista general de las interacciones de los diferentes componentes y actores de la solución en donde se detalla o estima las posibles solicitudes/comunicaciones que van de proceso en proceso hasta

terminar con la entrega del servicio en caso exitoso o entrega de la información de rechazo del mismo.

A continuación empezamos a detallar todos los casos de uso de la solución, correlacionados con las actividades mencionadas en la Tabla 11:

CU001	Generar Anexo Técnico	
Actividad	A1	
Dependencias	Comunicación con el cliente	
	Entender necesidad del cliente y levantar la información	
Pre condición	El Asesor comercial levanta la información con el cliente con toda su necesidad	
Descripción	Actividad para llenar el anexo técnico con el requerimiento del cliente	
Secuencia	Paso	Acción
	1	Se solicita información
	2	Se llena documento técnico
Post condición	Anexo técnico es entregado al asistente comercia para el posterior ingreso en el sistema	

Tabla 13 : CU001 - Generación de Anexo Técnico

Fuente : Autor

CU002	Generación de Orden de Servicio	
Actividad	A2	
Dependencias	Creación del anexo técnico	
	Entrega del anexo por parte del Asesor Comercial	
Pre condición	Entrega del anexo por parte del Asesor Comercial	
Descripción	Actividad que permite generar la orden en el sistema de información	
Secuencia	Paso	Acción
	1	Se ingresa los registros del anexo técnico al sistema
	2	Se guarda y crea la Solución
Post condición	Se solicita factibilidad técnica sobre el servicio de Cloud IAAS una vez creada la solución	

Tabla 14 : CU002 - Generación de Orden de Servicio

Fuente : Autor

CU003	Solicitar Factibilidad Técnica	
Actividad	A3	
Dependencias	Orden de Servicio creada (Solución)	
Pre condición	Solución creada	
Descripción	Actividad que permite generar la solicitud de Factibilidad y disparar el flujo de CAMUNDA	
Secuencia	Paso	Acción
	1	Se selecciona generar Factibilidad técnica
	2	Se dispara flujo automático
Post condición	Se validará si el flujo automático es satisfactorio, caso contrario se realiza flujo manual	

Tabla 15 : CU003 - Solicitar Factibilidad Técnica

Fuente : Autor

CU004	Validar Disponibilidades y capacidades	
Actividad	A4	
Dependencias	Solicitud generada para iniciar el flujo	
Pre condición	Solicitud creada	
Descripción	Actividad encargada de validar disponibilidad técnica contra el API de VMWare	
Secuencia	Pas o	Acción
	1	Se envia petición hacia la plataforma VMWare (API)
	2	Se obtiene respuesta
Post condición	Acorde a la respuesta del API, el servicio seguirá el flujo o terminará el mismo para iniciar flujo manual	

Tabla 16 : CU004 - Validar disponibilidad y capacidades

Fuente : Autor

CU005	Crear Máquinas Virtuales	
Actividad	A5	
Dependencias	Factibilidad técnica generada	
Pre condición	Factibilidad técnica generada de forma exitosa	
Descripción	Actividad encargada de crear las máquinas virtuales contra el API de VMWare	
Secuencia	Pas o	Acción
	1	Se envia petición hacia la plataforma VMWare (API)
	2	Se obtiene respuesta
Post condición	Si la máquina virtual es creada, se informa para que se realice planificación con el cliente	

Tabla 17 : CU005 - Crear máquinas virtuales

Fuente : Autor

CU006	Planificar entrega y activación de servicio	
Actividad	A8	
Dependencias	Máquina virtua creada correctamente	
Pre condición	Máquina creada automáticamente como manualmente	
Descripción	Actividad encargada de planificar la entrega del servicio al cliente y posterior activación del servicio	
Secuencia	Paso	Acción
	1	Planificación con el ciente
	2	Activación del Servicio
Post condición	Monitoreo post activación	

Tabla 18 : CU006 - Planificar entrega y activación de servicio

Fuente : Autor

Los casos de uso mostrados, servirán como guía para la implementación del flujo, se toma como referencia, aquellos que cumplen una condición de NO Error en el flujo que interviene la automatización, para poder esquematizar de mejor manera el comportamiento del proceso.

Finalmente, en el siguiente y último capítulo, validaremos el proceso y su posible comportamiento que nos permita con todo lo mencionado y analizado, mediante una prueba de concepto que pueda demostrar los objetivos planteados en este documento.

CAPÍTULO 5

ANÁLISIS DE RESULTADOS

En este último capítulo, nos enfocaremos precisamente en poder demostrar cómo la propuesta de mejora aporta a la optimización del proceso actual, realizaremos una proyección de los tiempos posibles de respuesta que resulten de la automatización vs los obtenidos en el transcurso de 5 años como se menciona en el capítulo anterior.

5.1 Prueba de concepto

Para esta prueba de concepto nos remitiremos al alcance detallado en el capítulo 3 en donde, en primera instancia el enfoque se basa en la simulación de los procesos de mejora planteadas y su posible comportamiento operativo, esto es, microservicios de acción en donde tendremos dos enfoques: validación de recursos y el aprovisionamiento de las máquinas virtuales como tal, estos luego tendrán que ser ejecutados por medio del flujo BPMN (modelo

TO-BE) para poder ir simulando esta oportunidad de mejora, el microservicio será el encargado de interactuar contra el API del Virtualizador para realizar el trabajo de automatización requerido.

Sin embargo, como detalle no menor y como se mencionó anteriormente, al no tener un ambiente controlado disponibilizado que nos permita comprobar de forma real las respuestas de ejecución del API, usaremos una aplicación que permita realizar un Mock con las definiciones de la misma, con aquellas funcionalidades que permitan demostrar nuestro objetivo.

Dado lo mencionado, tomaremos como referencia lo establecido en la Tabla 3, y realizaremos el modelamiento de las mismas con la herramienta de Mock (Wiremock – <https://app.wiremock.cloud>)

Mock APIs



Name	Base URL
Vcloud Simulator	https://vcloud.wiremockapi.cloud/  

Imagen 5.1 : Definición del Mock con HOST virtual

Fuente: Autor

En donde simularemos el host que se establecería en producción pero estableciendo las peticiones acorde como se establece en la documentación y que serán útiles para la prueba.

Definición del API

A continuación el detalle del cómo se establece cada definición del API con la simulación de request y response para poder ser llamado por un RestClient tipo Postman o específicamente desde la ejecución del flujo BPMN automatizado usando el microservicio:

Folder: /api/vcenter/folder

Request

Name

Vcenter Folder List

+ Add description

GET



/api/vcenter/folder

Imagen 5.2 : Folder Request

Fuente: Autor

Request duration: 7ms

```
HTTP/1.1 200 OK
Matched-Stub-Id: 2b901108-e6c2-4bd4-a0f9-d82de170dd0e
Matched-Stub-Name: Vcenter Folder List
Vary: Accept-Encoding, User-Agent

[
  {
    "folder": "cliente-prueba",
    "name": "cliente-prueba",
    "type": "DATACENTER"
  }
]
```

Imagen 5.3 : Definición de folder response en base a documentación

Fuente: Autor

Cluster: /api/vcenter/cluster/{cluster}

Request

Name

Vcenter Get Cluster

+ Add description

GET



/api/vcenter/cluster/cluster-clientes

Imagen 5.4 : Cluster API Request

Fuente: Autor

Request duration: 6ms

```
HTTP/1.1 200 OK
Matched-Stub-Id: 11386876-1d5d-4df7-b022-ac64254c78dc
Matched-Stub-Name: Vcenter Get Cluster
Vary: Accept-Encoding, User-Agent

{
  "name": "cluster-clientes",
  "resource_pool": "pool-clientes"
}
```

Imagen 5.5 : Definición de cluster response en base a documentación

Fuente: Autor

Datastore: /api/vcenter/datastore

Request

Name

Vcenter datastore All

+ Add description

GET

/api/vcenter/datastore

Imagen 5.6 : Datastore API Request

Fuente: Autor

Request duration: 6ms

```

HTTP/1.1 200 OK
Matched-Stub-Id: 1472e192-218a-490f-b0bc-dbb9fab960be
Matched-Stub-Name: Vcenter datastore All
Vary: Accept-Encoding, User-Agent

[
  {
    "capacity": 1000,
    "datastore": "clientes-datastore",
    "free_space": 500,
    "name": "clientes-datastore",
    "type": "VMFS"
  },
  {
    "capacity": 1000,
    "datastore": "clientes-datastore",
    "free_space": 200,
    "name": "clientes-datastore-alt",
    "type": "VMFS"
  }
]

```

Imagen 5.7 : Definición de datastore response en base a documentación

Fuente: Autor

Resource Pool: /api/vcenter/resource-pool/{resource}

Request

Name

[+ Add description](#)

GET



Imagen 5.8 : Resource Pool API Request

Fuente: Autor

Request duration: 7ms

```
HTTP/1.1 200 OK
Matched-Stub-Id: af4db591-b3b3-4ac8-a6ff-b531cad2f33f
Matched-Stub-Name: Vcenter Resource Pool
Vary: Accept-Encoding, User-Agent

{
  "cpu_allocation": {
    "expandable_reservation": false,
    "limit": 48,
    "reservation": 0
  },
  "memory_allocation": {
    "expandable_reservation": false,
    "limit": 48,
    "reservation": 0
  },
  "name": "pool-clientes"
}
```

Imagen 5.9 : Definición de Resource-pool response en base a documentación

Fuente: Autor

Host: /api/vcenter/host

Request

Name

Vcenter create Host

⊕ Add description

POST

/api/vcenter/host

Imagen 5.10 : Host API Request

Fuente: Autor

Request duration: 12ms

```
HTTP/1.1 200 OK
Matched-Stub-Id: f87efe3d-2380-4c97-8abc-ebd59bcd990b
Matched-Stub-Name: Vcenter create Host
Vary: Accept-Encoding, User-Agent

{
  "hostname": "host-clientes"
}
```

Imagen 5.11 : Definición de Host response en base a documentación

Fuente: Autor

Máquina Virtual: /api/vcenter/vm

Request

Name

Vcenter create VM

⊕ Add description

POST

/api/vcenter/vm

```
Request body
{
  "guest_OS": "DOS",
  "placement": {
    "datastore": "string",
    "folder": "string"
  }
}
```

Imagen 5.12 : Máquina Virtual API Request

Fuente: Autor

Request duration: 8ms

```
HTTP/1.1 200 OK
Matched-Stub-Id: 6c86e032-c923-4d7c-854a-3acbcblbd96e
Matched-Stub-Name: Vcenter create VM
Vary: Accept-Encoding, User-Agent

{
  "name": "maquina-virtual-local"
}
```

Imagen 5.13 : Definición de MV response en base a documentación

Fuente: Autor

Máquina Virtual Relocate: /api/vcenter/vm/{mv}?action=relocate

Request

Name

[+ Add description](#)

GET

```

Request body
{
  "disks": {
    "key": {
      "datastore": "string"
    }
  },
  "placement": {
    "cluster": "string",
    "datastore": "string",
    "folder": "string",
    "host": "string",
    "resource_pool": "string"
  }
}

```

Imagen 5.14 : Máquina Virtual Relocate API Request

Fuente: Autor

Una vez detallado las definiciones útiles para el efecto práctico de la demostración, a continuación procedemos a explicar y detallar el componente (microservicio) encargado de la comunicación con la plataforma Cloud (en la práctica), pero que por ahora haremos que se comunica contra el Simulador y validar el comportamiento posible del flujo.

Microservicio

En este bloque se detalla modularmente cómo se encuentra creado el microservicio de comunicación, en donde, podremos en su ejecución, ir viendo los logs de consumo de las diferentes APIs para las dos etapas de automatización establecidas en el modelo TO-BE de la *Imagen 4.1*.

El microservicio creado tiene como nombre ms-aprovisionamiento, este se encargará de ejecutar tanto el proceso de validación de capacidades como de

creación de máquinas virtuales de forma automática, a continuación, se muestra cómo se encuentra armado modularmente:

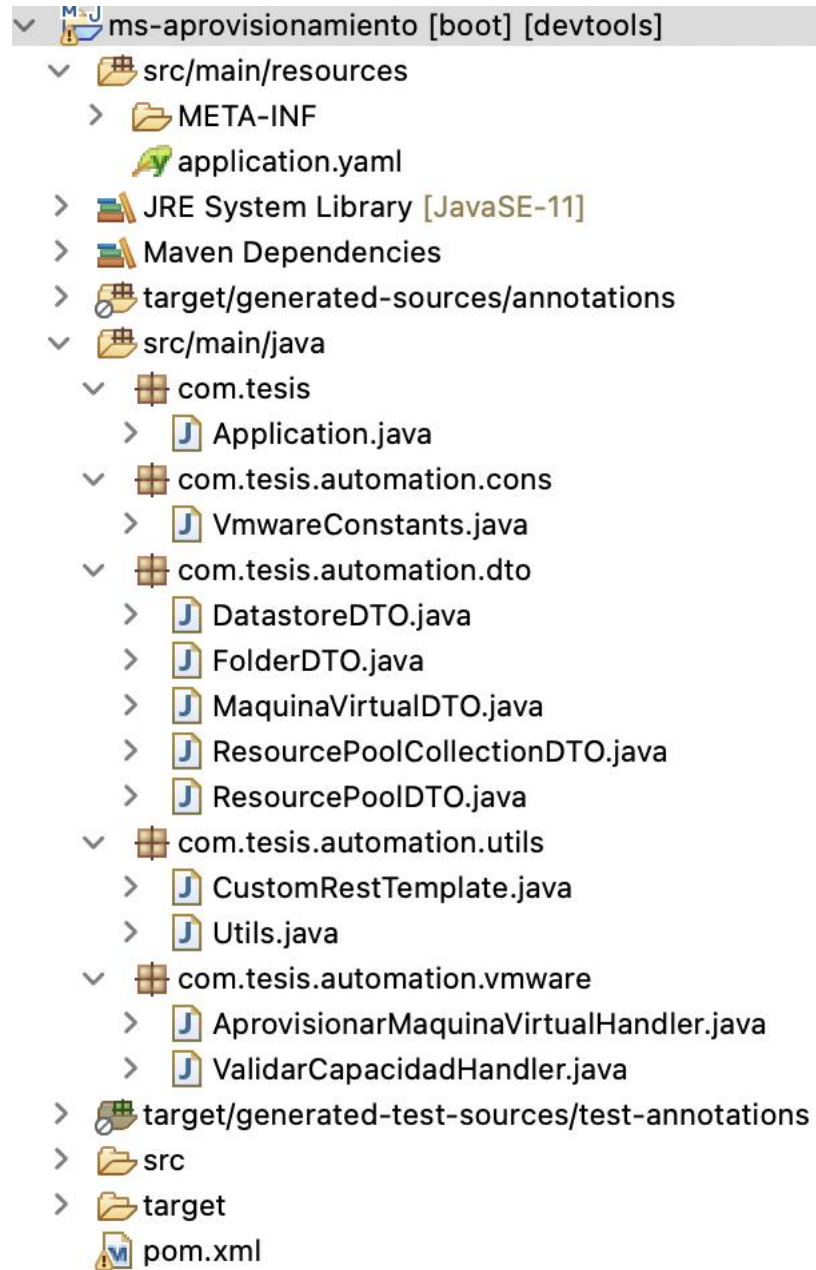


Imagen 5.15 : Estructura del microservicio de aprovisionamiento

Fuente: Autor

La implementación de los microservicios en general, son hechas utilizando SpringBoot como framework, y en donde, la capa com.tesis.automation.vmware contiene los servicios que se encargarán de la integración mencionada:

- ValidarCapacidadHanlder.java
- AprovisionarMaquinaVirtualHandler.java

Un punto importante de entender, es que, el microservicio además de la lógica de negocio, tiene el motor de CAMUNDA embebido dentro de su implementación, esto es, cuando se despliega es capaz de levantar una instancia del Cockpit en el servidor donde nos permitirá de forma rápida deployar el modelo planteado para darle gestión y seguimiento al flujo.

Dentro de la estructura tenemos el archivo de configuración application.yaml que es en donde registramos las APIs a ser ejecutadas en la implementación, el path del simulador de las definiciones, el puerto sobre el cual se levantará el servicio, las credenciales para acceder al Cockpit, el path y puerto del servicio de CAMUNDA embebido:

```

spring.datasource.url: jdbc:h2:file:./camunda-h2-database
server.port: 8001

} camunda.bpm:
} admin-user:
  id: demo
  password: demo
  firstName: Demo
} filter:
  create: All tasks

} camunda.bpm.client:
  base-url: http://localhost:8001/engine-rest # the URL pointing to the Camunda Platform Runtime REST API
  lock-duration: 10000 # defines how many milliseconds the External Tasks are locked until they can be fetched again

logging.level.org.camunda.bpm.client: INFO # increase the log level of the application

#vCloud vCenter simulator
vmware.vcenter.host: http://vcloud.wiremockapi.cloud
#GET
vmware.vcenter.folder.api: /api/vcenter/folder
vmware.vcenter.cluster.api: /api/vcenter/cluster/cluster-clientes
vmware.vcenter.datastore.api: /api/vcenter/datastore/{variable}
vmware.vcenter.datastore.all.api: /api/vcenter/datastore
vmware.vcenter.resource-pool.all.api: /api/vcenter/resource-pool
vmware.vcenter.resource-pool.api: /api/vcenter/resource-pool/{variable}
#POST
vmware.vcenter.host.api: /api/vcenter/host
vmware.vcenter.vm.api: /api/vcenter/vm
vmware.vcenter.vm.relocate.api: /api/vcenter/vm/{variable}?action=relocate

#HOST
vmware.host.hostname: domain.cluster-clientes
vmware.host.username: username
vmware.host.password: password

```

Imagen 5.16 : Configuraciones del servicio

Fuente: Autor

Una vez creada la definición del API y el servicio encargado de realizar el proceso de automatización, procederemos a levantar el mismo. Una vez arriba y para efecto de la simulación, crearemos un formulario con ayuda del **Camunda Modeler**, que me permita crear y ejecutar las tareas previas y posteriores al flujo de automatización, en donde, agregaremos la información técnica del requerimiento de un cliente para que el proceso pueda realizar las validaciones y creación de nuevas máquinas virtuales.

A continuación, se muestra a detalle un overview del segmento del modelo que está enfocado a la automatización:

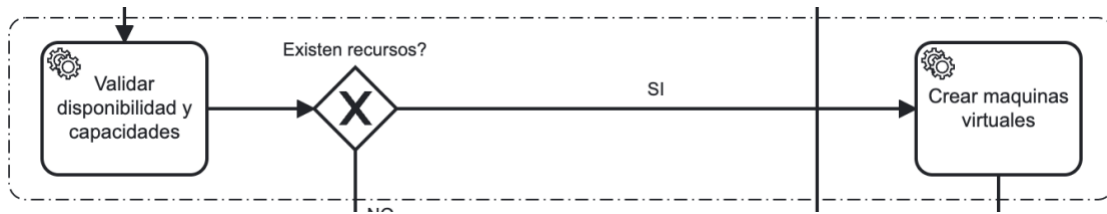


Imagen 5.17 : Segmento del Modelo TO-BE automatizado

Fuente: Autor

Procederemo a levantar el microservicios con el CAMUNDA embebido:

```
21:47:57.916 [Thread-0] DEBUG org.springframework.boot.devtools.restart.classloader.RestartClassLoader - Created RestartClassLoader org.springfram
```

```

Spring-Boot: (v2.7.3)
Camunda Platform: (v7.18.0)
Camunda Platform Spring Boot Starter: (v7.18.0)

2022-12-13 21:47:58.161 INFO 72291 --- [ restartedMain] com.tesis.Application : Starting Application using Java 17.0.3 on Mac
2022-12-13 21:47:58.162 INFO 72291 --- [ restartedMain] com.tesis.Application : No active profile set, falling back to 1 defai
2022-12-13 21:47:58.195 INFO 72291 --- [ restartedMain] o.s.b.devtools.restart.ChangeableUrls : The Class-Path manifest attribute in /Users/a
2022-12-13 21:47:58.196 INFO 72291 --- [ restartedMain] .e.DevToolsPropertyDefaultsPostProcessor : Devtools property defaults active! Set 'spring
2022-12-13 21:47:58.196 INFO 72291 --- [ restartedMain] .e.DevToolsPropertyDefaultsPostProcessor : For additional web related logging consider si
2022-12-13 21:47:58.932 INFO 72291 --- [ restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8001 (http)
2022-12-13 21:47:58.939 INFO 72291 --- [ restartedMain] org.apache.catalina.core.StandardService : Starting service [Tomcat]
2022-12-13 21:47:58.940 INFO 72291 --- [ restartedMain] org.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/9.0.6]
2022-12-13 21:47:58.993 INFO 72291 --- [ restartedMain] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationCon
2022-12-13 21:47:58.993 INFO 72291 --- [ restartedMain] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization coi
2022-12-13 21:47:59.008 INFO 72291 --- [ restartedMain] .c.b.s.b.s.r.CamundaJerseyResourceConfig : Configuring camunda rest api.
2022-12-13 21:47:59.020 INFO 72291 --- [ restartedMain] .c.b.s.b.s.r.CamundaJerseyResourceConfig : Finished configuring camunda rest api.
2022-12-13 21:47:59.123 INFO 72291 --- [ restartedMain] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Starting...
2022-12-13 21:47:59.295 INFO 72291 --- [ restartedMain] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Start completed.
  
```

Imagen 5.18 : Inicialización del Microservicio de aprovisionamiento

Fuente: Autor

Como la aplicación se levanta en localhost en el puerto 8001, nos dirigimos al navegador para poder acceder a la administración de Camunda, por medio del cockpit levantado. En este punto, ya podemos deployar tanto el modelo TO-BE como el formulario que se integrará a uno de los Task creados en el mismo, la URL a usar es la siguiente: <http://localhost:8001/camunda/app/welcome/default/#!/login> con las credenciales demo/demo según como se estableció en el archivo de configuración de la aplicación:

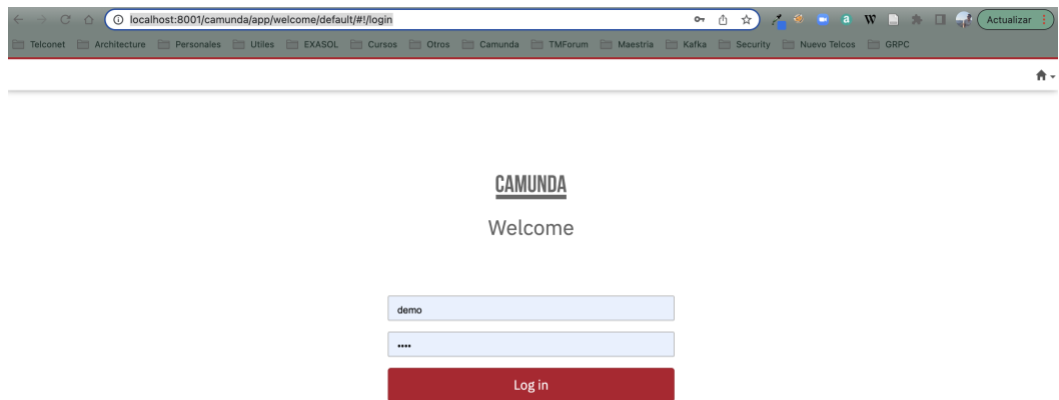
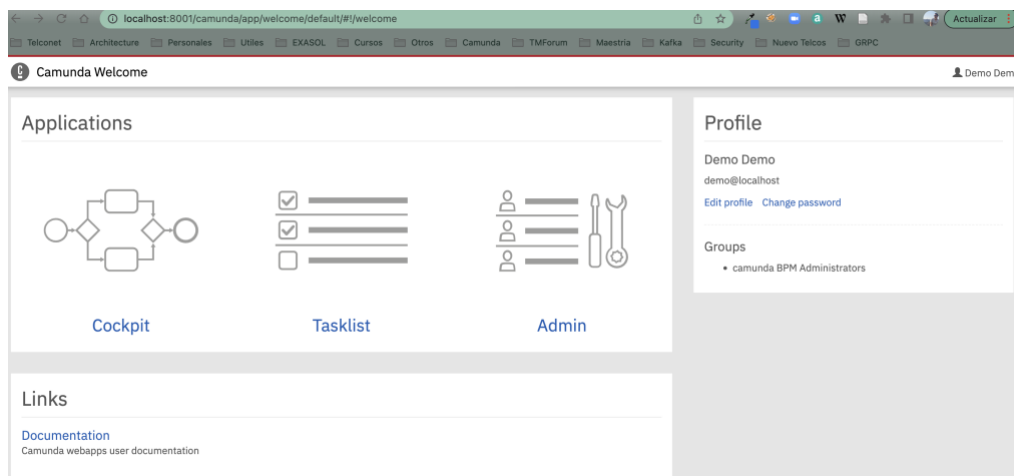


Imagen 5.19 : Login plataforma de Camunda

Fuente: Autor

En esta plataforma tendremos acceso a los 3 tipos de gestión que se puede tener sobre los flujos BPMN: Cockpit, Tasklist y Admin, el primero me permite tener visibilidad sobre el modelo deployado, el tasklist ejecutar las tareas manuales y el admin, en donde se crean los diferentes usuarios y grupos de acción sobre el flujo:



Imaga 5.20: Home page de Camunda

Fuente: Autor

Modelos BPMN

Luego de tener levantado el ambiente, procedemos con el despliegue del modelo TO-BE y del formulario de simulación de ingreso del requerimiento desde el Modeler de Camunda:

Anexo Tecnico

Ingresar Datos de Nueva Maquina Virtual:

Nombre maquina virtual*

Disco*

Cantidad Disco*

Memoria*

Cantidad Memoria

Procesador*

Cantidad Procesador

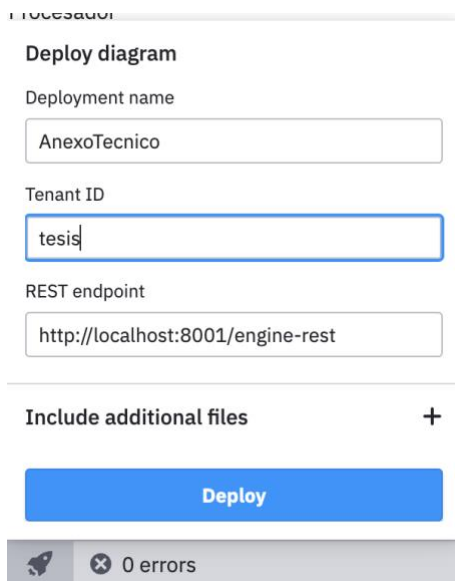
Sistema Operativo:*

Generar Orden

Imagen 5.21: Formulario de requerimientos técnicos

Fuente: Autor

Este formulario, básicamente me permite ingresar las capacidades requeridas e información básica que será validada a posterior por el proceso contra la plataforma virtual y los recursos alojados en vmware vCloud utilizando el API.



Procesador

Deploy diagram

Deployment name
AnexoTecnico

Tenant ID
tesis

REST endpoint
http://localhost:8001/engine-rest

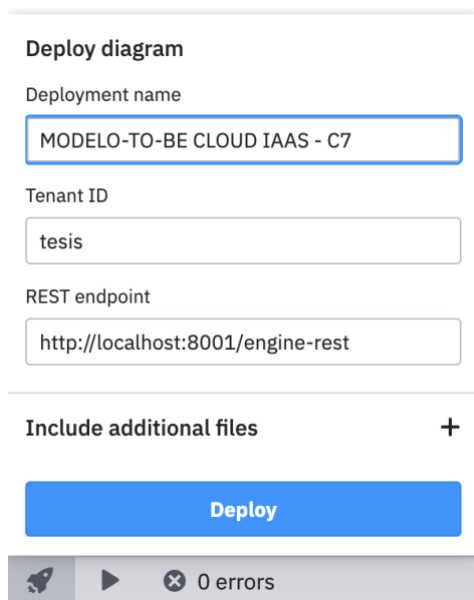
Include additional files +

Deploy

0 errors

Imagen 5.22: Despliegue del formulario sobre Camunda

Fuente: Autor



Deploy diagram

Deployment name
MODELO-TO-BE CLOUD IAAS - C7

Tenant ID
tesis

REST endpoint
http://localhost:8001/engine-rest

Include additional files +

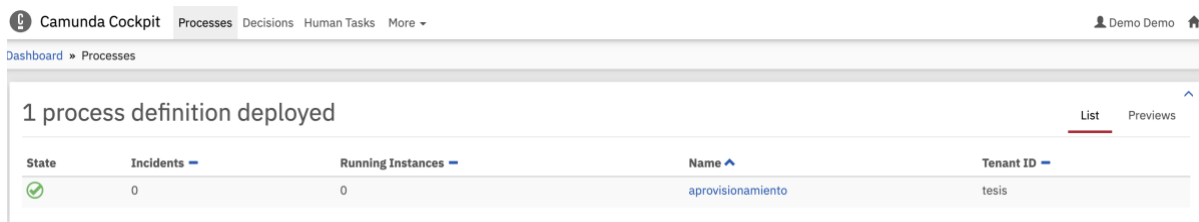
Deploy

0 errors

Imagen 5.23: Despliegue del modelo TO-BE sobre Camunda

Fuente: Autor

Una vez desplegado cada uno de los modelos creados, verificamos el formulario creado como proceso dentro del Cockpit de camunda:



The screenshot shows the Camunda Cockpit interface with the 'Processes' tab selected. It displays a table with one row representing a deployed process definition. The table has columns for State, Incidents, Running Instances, Name, and Tenant ID. The process name is 'aprovisionamiento' and the tenant ID is 'tesis'.

State	Incidents	Running Instances	Name	Tenant ID
✓	0	0	aprovisionamiento	tesis

Imagen 5.24 : Proceso cargado a la plataforma

Fuente: Autor

A continuación, se verifica justamente el detalle del modelo cargado y en donde nos enfocaremos en una variable de definición, que es la que nos ayudará a iniciar el flujo mediante el API que Camunda nos brinda para ese efecto:

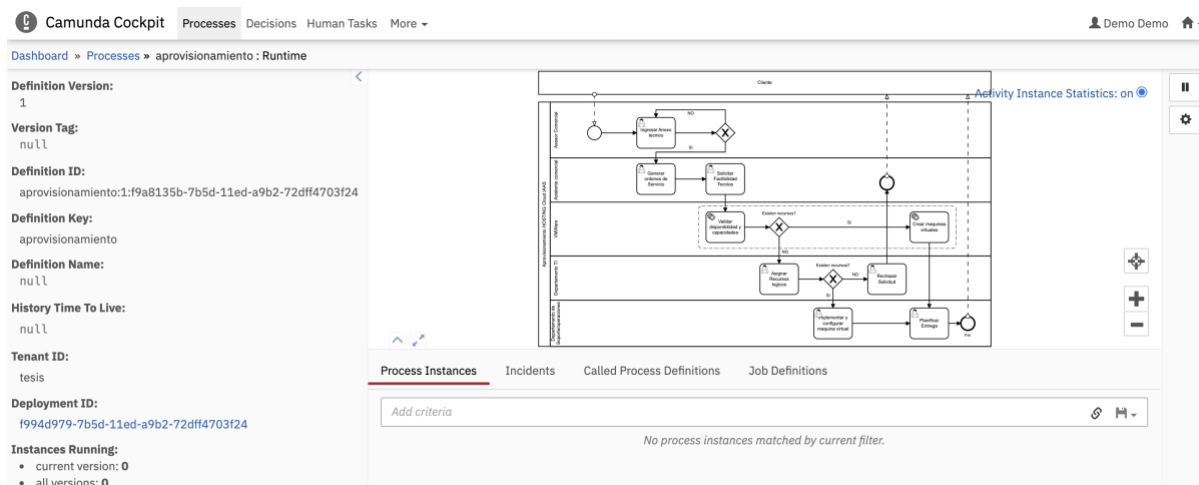


Imagen 5.25: Proceso con sus valores de definición

Fuente : Autor

La variable en mención es la *Definition ID* :

aprovisionamiento:1:f9a8135b-7b5d-11ed-a9b2-72dff4703f24

Esta variable es la que nos ayudará a iniciar el proceso sobre la plataforma con ayuda de la siguiente petición REST <http://localhost:8001/engine-rest/process-definition/aprovisionamiento:1:f9a8135b-7b5d-11ed-a9b2-72dff4703f24/start>:

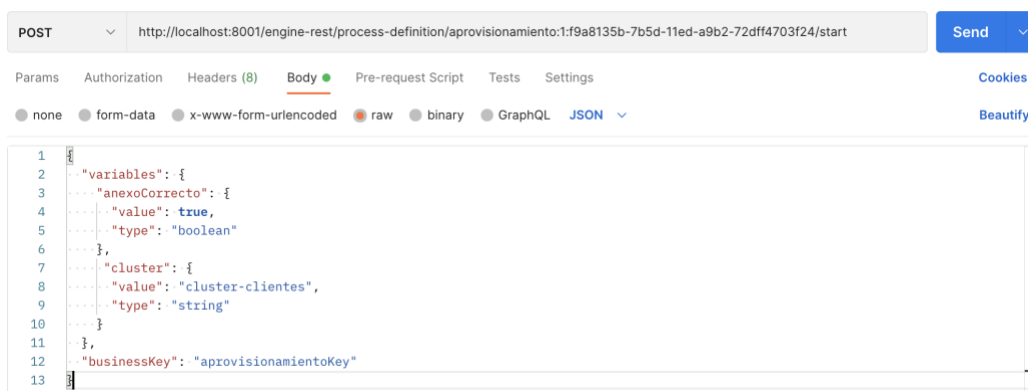


Imagen 5.26: Rest API para inicializar flujo el flujo del modelo

Fuente: Autor

En esta API, podemos agregar las variables necesarias adicionales que servirán como referencia en el flujo, por default y efecto práctico enviaremos un Cluster ficticio que es donde la máquina virtual se debería desplegar.

Previo a ejecutar el flujo, validaremos si el API de definición de vSphere simulado está operativo para la prueba, para esto utilizando de la misma manera el Postman, básicamente debería responder acorde a lo que se especificó previamente, para ejemplo, se ejecutará el API para obtener los

datastore

disponibilizados

(<http://vcloud.wiremockapi.cloud/api/vcenter/datastore>):

GET <http://vcloud.wiremockapi.cloud/api/vcenter/datastore> Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

KEY	VALUE	DESCRIPTION	Bulk Edit	Presets
Key	Value	Description		

Body Cookies Headers (5) Test Results Status: 200 OK Time: 195 ms Size: 313 B Save Response

Pretty Raw Preview Visualize Text

```

1  {
2  }
3  "capacity": 1000,
4  "datastore": "clientes-datastore",
5  "free_space": 500,
6  "name": "clientes-datastore",
7  "type": "VMFS"
8  },
9  {
10 "capacity": 1000,
11 "datastore": "clientes-datastore",
12 "free_space": 200,
13 "name": "clientes-datastore-alt",
14 "type": "VMFS"
15 }
16

```

Imagen 5.27 : Ejecución del API Simulador de vmware (datastore)

Fuente: Autor

Como se observa, la respuesta hace referencia a la definición creada en la Imagen 5.6 y la Imagen 5.7. Una vez comprobado que el API se encuentra funcional, nos dirigimos a dar inicio al flujo BPMN ejecutando el API REST de la Imagen 5.26:

The screenshot shows a REST client interface with a POST request to `http://localhost:8001/engine-rest/process-definition/aprovisionamiento:1:f9a8135b-7b5d-11ed-a9b2-72dff4703f24/start`. The request body is a JSON object:

```

1  {
2    "variables": {
3      "anexoCorrecto": {
4        "value": true,
5        "type": "boolean"
6      },
7      "cluster": {
8        "value": "cluster-clientes",
9        "type": "string"
10     }
11   },
12   "businessKey": "aprovisionamientoKey"
13 }

```

The response status is 200 OK, with a time of 118 ms and a size of 523 B. The response body is a JSON object:

```

1  {
2    "links": [
3      {
4        "method": "GET",
5        "href": "http://localhost:8001/engine-rest/process-instance/37fd6228-7b61-11ed-a9b2-72dff4703f24",
6        "rel": "self"
7      }
8    ],
9    "id": "37fd6228-7b61-11ed-a9b2-72dff4703f24",
10   "definitionId": "aprovisionamiento:1:f9a8135b-7b5d-11ed-a9b2-72dff4703f24",
11   "businessKey": "aprovisionamientoKey",
12   "caseInstanceId": null,
13   "ended": false,
14   "suspended": false,
15   "tenantId": "tesis"

```

Imagen 5.28: Inicio del flujo BPMN via Camunda API REST

Fuente: Autor

Acorde al orde establecido en el flujo recordar que tenemos el siguiente orden de tareas/procesos que se irán ejecutando:

- Ingresar Anexo técnico – manual
- Generar orden de servicio – manual
- Solicitar solicitud de factibilidad – manual
- *Validar disponibilidad y capacidades – automático*
- *Crear máquina virtual – automático*

- Planificación de entrega – manual

Vemos en la siguiente imagen como se da inicio al flujo, con el task de ingresar anexo técnico:

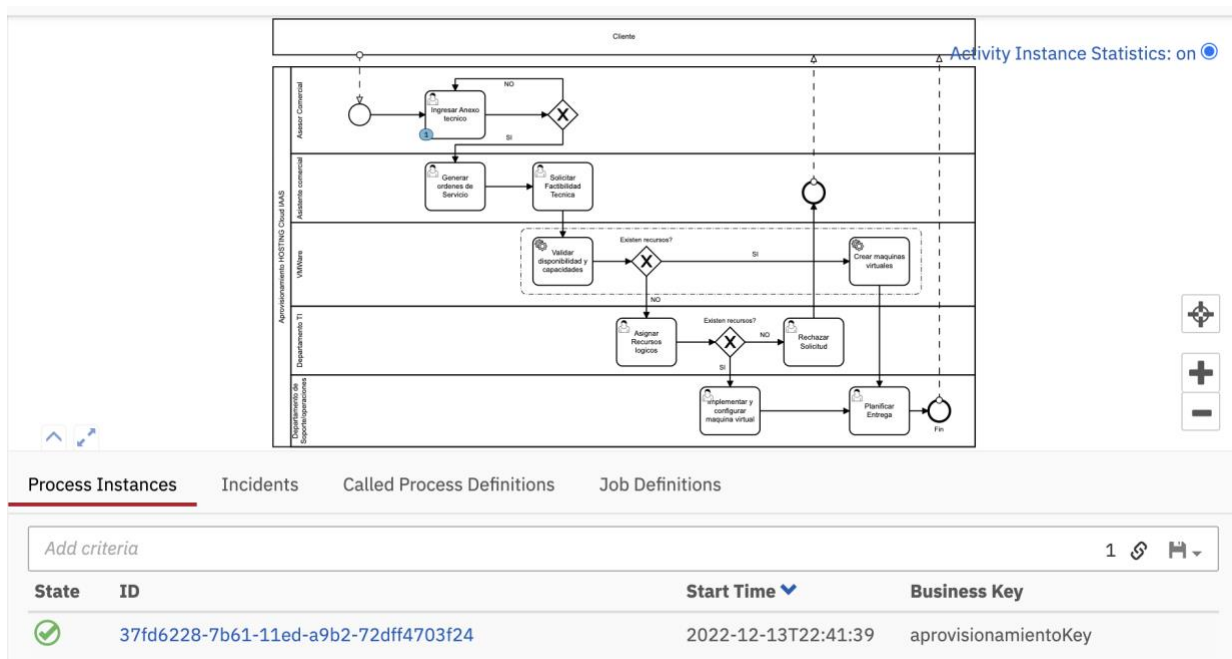


Imagen 5.29 : Inicio de proceso BPMN de aprovisionamiento

Fuente: Autor

Dado que el BPMN me indica que es un User Task, nos dirigimos a la sección de Task List para empezar las ejecuciones de este tipo:

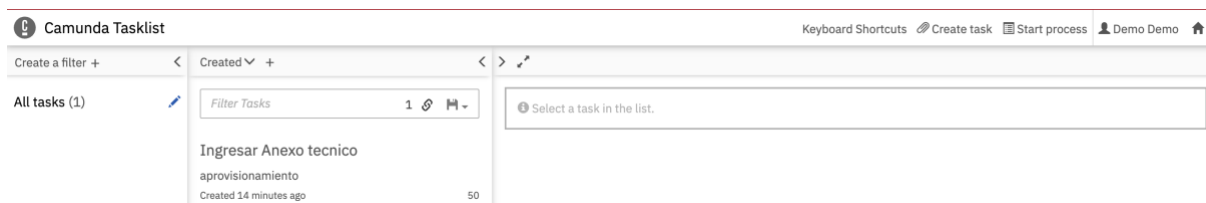


Imagen 5.30 : Ejecución del Task - Ingresar anexo técnico

Fuente: Autor

The screenshot displays the Camunda Tasklist interface. On the left, a task list shows a task titled 'Ingresar Anexo tecnico' with a status of 'aprovisionamiento' and a duration of '50' minutes. The main area shows the details of this task, titled 'Anexo Tecnico'. The task description is 'Ingresar Datos de Nueva Maquina Virtual:'. The form contains the following fields:

- Nombre maquina virtual***: maquina-virtual-local
- Disco***: SSD Rapido
- Cantidad Disco***: 250
- Memoria***: Memoria VMWare
- Cantidad Memoria**: 24
- Procesador***: Procesador VMWare
- Cantidad Procesador**: 24
- Sistema Operativo:***: Red Hat Linux 2.1

Imagen 5.31 : Ingreso de valores en formulario para el Task de anexo técnico

Fuente: Autor

Se realiza la ejecución del task manual hasta llegar al Task previo de la ejecución automática:

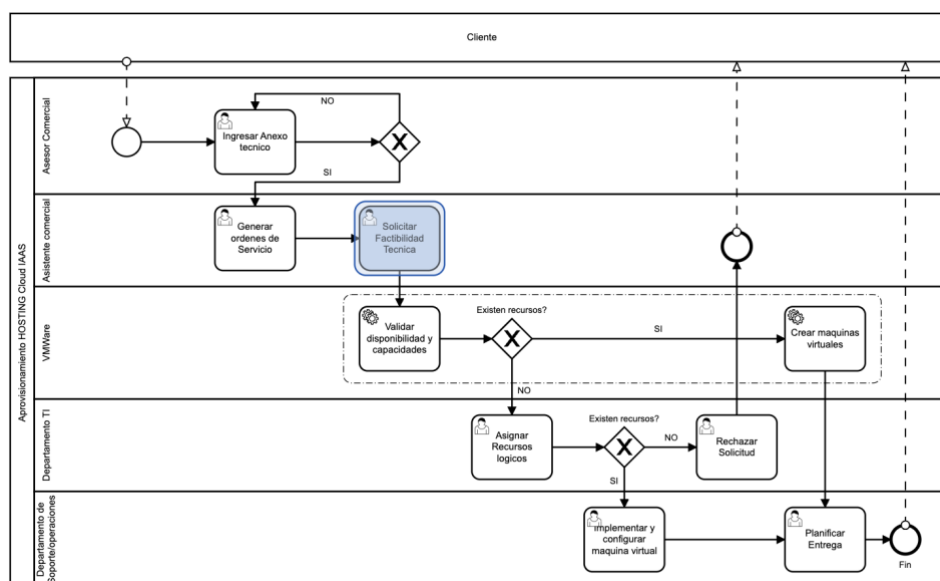


Imagen 5.32: Ejecución de flujo manual de Solicitar Factibilidad técnica

Fuente: Autor

Una vez ejecutado el último proceso manual, verificaremos dentro del log del servicio como se va reflejando los diferentes consumos del API de virtualización, primero verificamos que sucede en la validación de capacidades, como se ve en la siguiente imagen, es en el punto donde se encuentra el flujo:

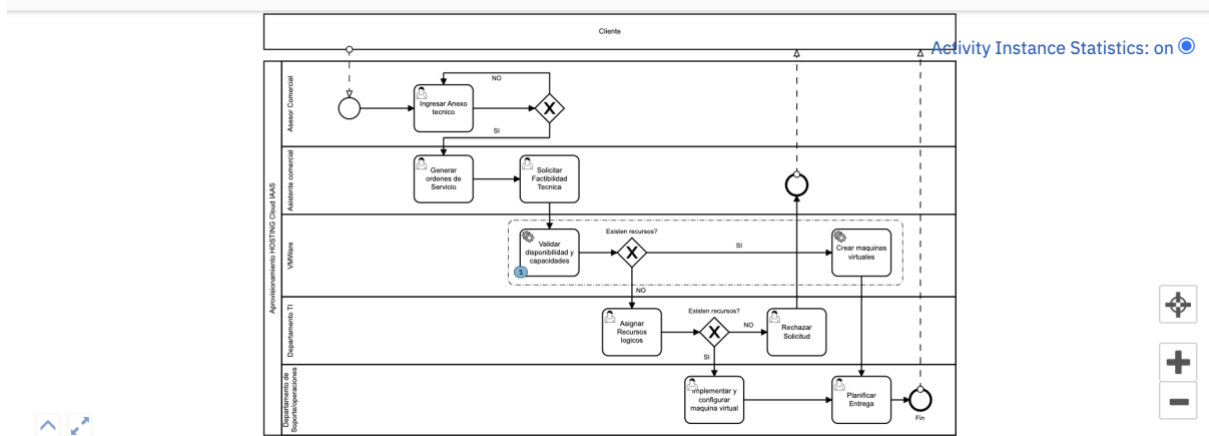


Imagen 5.33: Proceso de validación de capacidades dentro del flujo

Fuente: Autor

Verificamos el log de ejecución simulada:

```

INFO 72291 --- [criptionManager] Proceso Validar capacidades : -----Validaciones de capacidades-----
INFO 72291 --- [criptionManager] Proceso Validar capacidades : -----Validaciones de capacidades-----
INFO 72291 --- [criptionManager] Proceso Validar capacidades : Informacion de Maquina Virtual a validar capacidades :
INFO 72291 --- [criptionManager] Proceso Validar capacidades : Nombre Maquina : maquina-virtual-local
INFO 72291 --- [criptionManager] Proceso Validar capacidades : Sistema Operativo : REDHAT
INFO 72291 --- [criptionManager] Proceso Validar capacidades : Disco
INFO 72291 --- [criptionManager] Proceso Validar capacidades : --- Tipo: SSD Rapido
INFO 72291 --- [criptionManager] Proceso Validar capacidades : --- Cantidad: 250
INFO 72291 --- [criptionManager] Proceso Validar capacidades : Memoria
INFO 72291 --- [criptionManager] Proceso Validar capacidades : --- Tipo: Memoria VMWare
INFO 72291 --- [criptionManager] Proceso Validar capacidades : --- Cantidad: 24
INFO 72291 --- [criptionManager] Proceso Validar capacidades : Procesador
INFO 72291 --- [criptionManager] Proceso Validar capacidades : --- Tipo: Procesador VMWare
INFO 72291 --- [criptionManager] Proceso Validar capacidades : --- Cantidad: 24
INFO 72291 --- [criptionManager] Proceso Validar capacidades : -----Validando capacidad de disco-----
INFO 72291 --- [criptionManager] Proceso Validar capacidades : -----Validando capacidad de disco-----
INFO 72291 --- [criptionManager] Proceso Validar capacidades : Obteniendo los datastore disponibles...
INFO 72291 --- [criptionManager] Proceso Validar capacidades :
INFO 72291 --- [criptionManager] Proceso Validar capacidades : METHOD GET http://vcloud.wiremockapi.cloud/api/vcenter/datastore
INFO 72291 --- [criptionManager] Proceso Validar capacidades :
INFO 72291 --- [criptionManager] Proceso Validar capacidades : Datastores registrados = [{capacity=1000, datastore=clientes-datastore, free_space=500, name=clientes-datastore, typ
INFO 72291 --- [criptionManager] Proceso Validar capacidades : Validando capacidades de disco...
INFO 72291 --- [criptionManager] Proceso Validar capacidades :
INFO 72291 --- [criptionManager] Proceso Validar capacidades : Datastore disponible = { name : clientes-datastore , freeSpace : 500 GB}

```

Imagen 5.34 : Validación de capacidad de disco

Fuente: Autor


```

INFO 72291 ---- [criptionManager] Proceso creacion maquina virtual : -----
INFO 72291 ---- [criptionManager] Proceso creacion maquina virtual : -----Aprovisionamiento de MVs-----
INFO 72291 ---- [criptionManager] Proceso creacion maquina virtual : -----
INFO 72291 ---- [criptionManager] Proceso creacion maquina virtual : -----
INFO 72291 ---- [criptionManager] Proceso creacion maquina virtual : ----- Asignacion de Directorio de MV-----
INFO 72291 ---- [criptionManager] Proceso creacion maquina virtual : -----
INFO 72291 ---- [criptionManager] Proceso creacion maquina virtual : Obteniendo el Folder a asignar a la nueva Maquina Virtual
INFO 72291 ---- [criptionManager] Proceso creacion maquina virtual :
INFO 72291 ---- [criptionManager] Proceso creacion maquina virtual : METHOD GET http://vcloud.wiremockapi.cloud/api/vcenter/folder
INFO 72291 ---- [criptionManager] Proceso creacion maquina virtual :
INFO 72291 ---- [criptionManager] Proceso creacion maquina virtual : Folders registrados = [{folder=cliente-prueba, name=cliente-prueba, type=DATACENTER}]
INFO 72291 ---- [criptionManager] Proceso creacion maquina virtual : Folder disponible = [ name : cliente-prueba , type : DATACENTER]
INFO 72291 ---- [criptionManager] Proceso creacion maquina virtual :
INFO 72291 ---- [criptionManager] Proceso creacion maquina virtual : ----- Creando nuevo HOST para MV-----
INFO 72291 ---- [criptionManager] Proceso creacion maquina virtual : -----
INFO 72291 ---- [criptionManager] Proceso creacion maquina virtual : Creando un nuevo HOST
INFO 72291 ---- [criptionManager] Proceso creacion maquina virtual : Hostname : domain.cluster-clientes
INFO 72291 ---- [criptionManager] Proceso creacion maquina virtual : Username : username
INFO 72291 ---- [criptionManager] Proceso creacion maquina virtual : Password : password
INFO 72291 ---- [criptionManager] Proceso creacion maquina virtual :
INFO 72291 ---- [criptionManager] Proceso creacion maquina virtual : METHOD POST http://vcloud.wiremockapi.cloud/api/vcenter/host
INFO 72291 ---- [criptionManager] Proceso creacion maquina virtual :
INFO 72291 ---- [criptionManager] Proceso creacion maquina virtual : Host creado = [{folder=cliente-prueba, name=cliente-prueba, type=DATACENTER}]
INFO 72291 ---- [criptionManager] Proceso creacion maquina virtual : Host disponible = [ hostname : domain.cluster-clientes ]
INFO 72291 ---- [criptionManager] Proceso creacion maquina virtual :
INFO 72291 ---- [criptionManager] Proceso creacion maquina virtual : ----- Creando nueva Maquina Virtual-----
INFO 72291 ---- [criptionManager] Proceso creacion maquina virtual : -----
INFO 72291 ---- [criptionManager] Proceso creacion maquina virtual : Datos de Maquina Virtual a ser creada:
INFO 72291 ---- [criptionManager] Proceso creacion maquina virtual : Name : maquina-virtual-local
INFO 72291 ---- [criptionManager] Proceso creacion maquina virtual : Guest_OS : REDHAT
INFO 72291 ---- [criptionManager] Proceso creacion maquina virtual : Cluster : cluster-clientes
INFO 72291 ---- [criptionManager] Proceso creacion maquina virtual : Datastore : null
INFO 72291 ---- [criptionManager] Proceso creacion maquina virtual : Folder : cliente-prueba
INFO 72291 ---- [criptionManager] Proceso creacion maquina virtual : Host : domain.cluster-clientes
INFO 72291 ---- [criptionManager] Proceso creacion maquina virtual : Resource : pool-clientes
INFO 72291 ---- [criptionManager] Proceso creacion maquina virtual :
INFO 72291 ---- [criptionManager] Proceso creacion maquina virtual : METHOD POST http://vcloud.wiremockapi.cloud/api/vcenter/vm/maquina-virtual-local?action=relocate
INFO 72291 ---- [criptionManager] Proceso creacion maquina virtual :
INFO 72291 ---- [criptionManager] Proceso creacion maquina virtual : Maquina Virtual creada = {name=maquina-virtual-local}

```

Imagen 5.37: Creación de máquina virtual

Fuente: Autor

Como se observa, los pasos para esta parte del proceso consiste en la creación de nuevo host donde se relaciona a la nueva máquina virtual, asignación del directorio en donde se va a alojar y por último la creación de la máquina virtual como tal, en conjunción de todos los datos recopilados en todo el flujo:

```

: Datos de Maquina Virtual a ser creada:
: Name : maquina-virtual-local
: Guest_OS : REDHAT
: Cluster : cluster-clientes
: Datastore : null
: Folder : cliente-prueba
: Host : domain.cluster-clientes
: Resource : pool-clientes
:
: METHOD POST http://vcloud.wiremockapi.cloud/api/vcenter/vm/maquina-virtual-local?action=relocate
:
: Maquina Virtual creada = {name=maquina-virtual-local}

```

Imagen 5.38: Máquina Virtual con información para creación

Fuente: Autor

Por último, ejecutamos el último proceso manual para darle final a la ejecución del proceso:

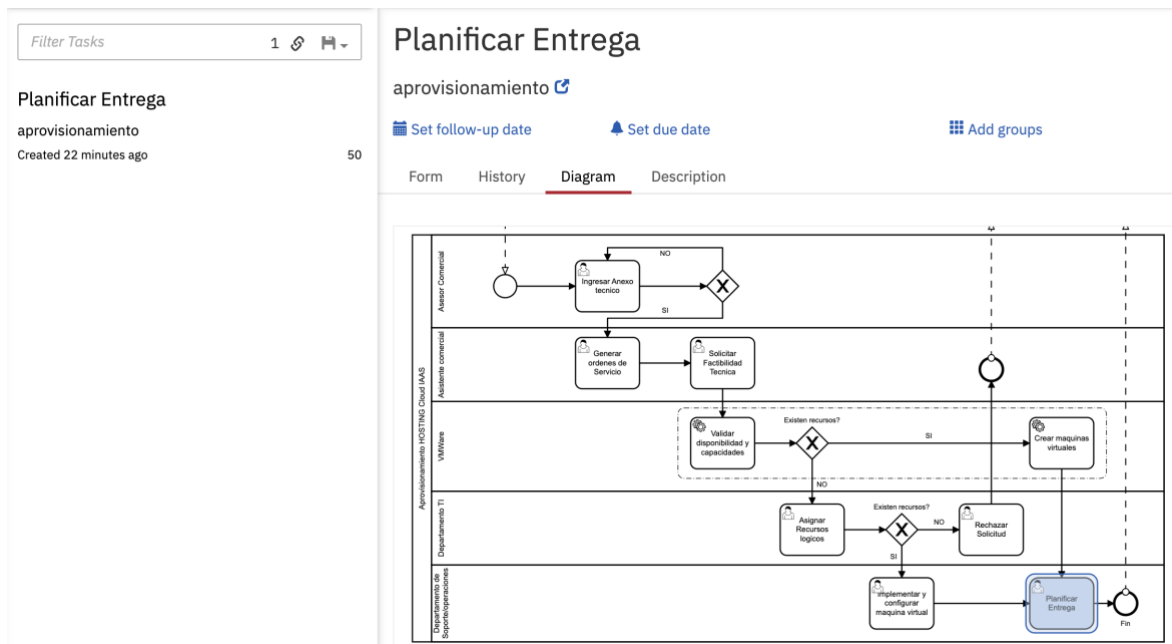


Imagen 5.39: User Task para planificación de entrega

Fuente: Autor

Una vez completado el último task (dentro de un caso de uso ideal), se da por completado la ejecución de la instancia del proceso.

Con esto, se presentó una prueba de concepto funcional en donde se pudo mostrar el modelo TO-BE planteado en ejecución en conjunto con el microservicio encargado del procesamiento y gestionado dentro del Cockpit y Tasklist que Camunda provee.

5.2 Análisis de resultados

En este último sub capítulo haremos énfasis en la prueba de concepto realizada, la cual nos brinda una visión del cómo esta implementación, una vez realizada, puede ayudar a mejorar considerablemente el problema actual.

Como se explicó en el capítulo 3 y según se detalla en la *Tabla 8*, los tiempos respecto a verificación de capacidades técnicas y los tiempos de aprovisionamiento de máquinas virtuales, fueron el enfoque de la simulación dentro de la prueba de concepto (servicios de integración dentro del microservicio), en donde, podríamos ser capaces de determinar que el tiempo total de este segmento del flujo estaría ligado 100% a la disponibilidad del servicio expuesto por la plataforma, es decir, si la misma está levantada en una infraestructura de alta disponibilidad, podríamos asegurar que el tiempo de respuesta en el peor escenario podría ser tan lento como las políticas de Timeout que se configuren en el servicio que consume el API, es decir, aplicando un patrón de Circuit Breaker para evitar intentos de consumo sin respuesta válidas o conectividad, o tan rápido como el tiempo de respuesta promedio del mismo, que en términos de eficiencia, no debería pasar de milisegundos entre cada una de las llamadas según el listado mostrado en la *Tabla 3*, sumado a que, podríamos tener también interacción con una base de datos para validar o transaccionar elementos que la operación necesite.

En términos generales entonces podríamos concluir que, el proceso que actualmente existe, el cual llega a demorar en el mejor escenario, un promedio

2 días acorde al resumen de la *Tabla 10*, con el trabajo de gestión manual, en su mejora, podría llegar a demorar lo que el API en el peor escenario pueda responder (máximo minutos de procesamiento) según los factores mencionados, por tanto, la solución terminará siendo bastante eficiente.

Claro está, como detalle no menor, que si el API llegase a fallar, queda la contingencia del proceso manual que permite darle continuidad al flujo, pero, tomando en cuenta que se debe siempre asegurar la alta disponibilidad de los sistemas.

CONCLUSIONES Y RECOMENDACIONES

Conclusiones

1. Tener claro los procesos y sobre todo, saber elegir cual de estos podría ser parte de una transformación digital, es vital para la mejora del servicio que la compañía.
2. El buen uso y entendimiento del modelado BPMN, lo convierte en una herramienta potente en el momento de poder establecer oportunidades de mejora de los procesos internos.
3. El trabajo manual no siempre termina siendo eficiente, sobre todo, cuando aun existen Silos dentro de una compañía tanto a nivel humano como a nivel de sistemas de información.
4. La tecnología siempre es un aliado clave en el momento de pensar en la transformación digital de procesos críticos en la compañía y no sólo pensar que el trabajo manual es siempre seguro o ágil.

Recomendaciones

1. Entender las necesidades del negocio, se convierte en un factor clave dentro de la compañía, de cara a la calidad de servicio.
2. Adquisición de conocimiento técnico que permita a las áreas poder tener apertura en confiar en la tecnología que permita resolver problemas de forma más eficiente.
3. Saber identificar aquellos indicadores que puedan causar puntos de dolor, entrega una visión general de dónde se debe enfocar una propuesta de mejora.
4. Tener alineada cada una de las áreas para entender las necesidades de estas dentro de un mismo flujo o proceso de trabajo ayudaría a agilizar estos, permitiendo así, ofrecer servicios de calidad y mejor aprovechamiento de recursos.

BIBLIOGRAFÍA

- [1] W. G. S. Guerrero, W. M. C. Rojas, M. del P. D. Sánchez, y A. E. Villamizar, «Arquitectura empresarial–dominios y beneficios», *FACE Rev. Fac. Cienc. Económicas Empres.*, vol. 16, n.º 1, pp. 87-92, 2016.
- [2] Jéssica Angeli, «¿Qué es el mapeo de procesos AS IS/TO BE?», 29 de junio de 2018. <https://www.neomind.com.br/es/blog/que-es-el-mapeo-de-procesos-as-is-to-be/>
- [3] M. B. A. María Pérez Savelli y V. Quiñones, «El diagrama o blueprint del servicio: Herramienta de diseño y control en la prestación de los intangibles», *Horiz. Empres.*, vol. 8, n.º 1, pp. 63-72, 2009.
- [4] B. Hitpass, *BPM: Business Process Management: Fundamentos y Conceptos de Implementación 4a Edición actualizada y ampliada*. Dr. Bernhard Hitpass, 2017.
- [5] C. I. Vázquez-Ramírez, «Migración de sistemas de información a IAAS, PAAS y SAAS en una Pyme», 2021.
- [6] CAMUNDA, *Camunda Platform User Guide*. [En línea]. Disponible en: https://docs.camunda.org/manual/7.16/?__hstc=252030934.ed79f99bf12068997ab385d4322c448b.1649007391907.1649007391907.1649007391907.1&__hssc=252030934.5.1649007391908&__hsfp=1160391030
- [7] I. Aguilar Cuba, «Implementación de procesos de negocio en motores de código abierto: un estudio de caso», Universitat Politècnica de València, 2019. [En línea]. Disponible en:

<https://riunet.upv.es/bitstream/handle/10251/127337/Aguilar%20-%20Implementaci%C3%B3n%20de%20procesos%20de%20negocio%20en%20motores%20de%20c%C3%B3digo%20abierto%3A%20un%20estudio%20de%20caso.pdf?sequence=1>

[8] D. López y E. Maya, «Arquitectura de Software basada en Microservicios para Desarrollo de Aplicaciones Web», 2017.

[9] vmware, *VMWARE VCloud API*. [En línea]. Disponible en: <https://developer.vmware.com/apis/72/vmware-cloud-director>

[10] O. Krieger, P. McGachey, y A. Kanevsky, «Enabling a marketplace of clouds: VMware's vCloud director», *ACM SIGOPS Oper. Syst. Rev.*, vol. 44, n.º 4, pp. 103-114, 2010.

[11] M. Brunnello, M. Rocha, y C. M. R. Vargas, «Modelado de procesos», *Igarss 2014*, vol. 1, pp. 1-5, 2011.

[12] J. F. G. Estupiñán, «Análisis de BPMN como herramienta integral para el Modelado de Procesos de Negocio [Analysis of BPMN as Integral Tool for Business Process Modeling]», *Ventana Informática*, n.º 30, 2014.

[13] SYDLE, «Referencia de tipos de mapa de procesos». [En línea]. Disponible en: <https://www.sydle.com/es/blog/mapear-procesos-as-is-to-be-to-do-60a81ebd22559e108ed7f51e/>

[14] A. Roman, «Adaptable Processes: Concepts, Design and Implementation in Camunda», 2021.

- [15] J. Bonhomme y E. Camejo, «Plataforma de Integración basada en Microservicios», 2019.
- [16] M. A. Murazzo, M. J. Guevara, M. Tello, N. R. Rodríguez, F. Piccoli, y M. Giménez, «Orquestación de servicios para el desarrollo de aplicaciones para big data», presentado en VI Jornadas de Cloud Computing & Big Data (JCC&BD)(La Plata, 2018), 2018.
- [17] DECIDE, «DECIDE - Arquitectura de microservicios». [En línea]. Disponible en: <https://decidesoluciones.es/arquitectura-de-microservicios/>
- [18] V. Surwase, «REST API modeling languages-a developer's perspective», *Int J Sci Technol Eng*, vol. 2, n.º 10, pp. 634-637, 2016.
- [19] Y. F. Romero y K. G. Pombo, «Virtualización», *Telemática*, vol. 10, n.º 3, pp. 61-73, 2011.
- [20] J. Gómez y E. Villar, «Introducción a la virtualización», 2018.
- [21] P. Mell y T. Grance, «The NIST definition of cloud computing», 2011.
- [22] D. Mirzoev y R. Alvarez, «Leveraging VMware vCloud director virtual applications (vApps) for operational expense (OpEx) efficiency», *ArXiv Prepr. ArXiv14042157*, 2014.
- [23] A. Díaz Betizagasti y M. Sureda, «Extensibilidad de herramienta de gestión para VMware Cloud Director», 2022.
- [24] VMGuru, «VMware vCloud Director design guidelines». [En línea]. Disponible

en: <https://vmguru.com/2012/01/vmware-vcloud-director-design-guidelines/>

[25] VMWARE, «VMWARE vCenter API REST». [En línea]. Disponible en: <https://developer.vmware.com/apis/vsphere-automation/latest/vcenter/>