Indice

Historial de revisiones

Versión	Autor(es)	Descripción	Fecha
1.0	Roberto Zuñiga Roman, Diana Iratze Solano Uscanga	Creación del documento	04/11/2025
1.1	Roberto Zuñiga Roman, Diana Iratze Solano Uscanga	Correcciones al documento	11/03/2025

Alcance

El objetivo de este documento es funcionar como una guía detallada para la planificación y ejecución de las actividades básicas de un proceso de pruebas, en este caso para el sistema VirtX. El plan debe busca cubrir la validación de las funciones principales o básicas, por ejemplo, la gestión de citas, rutinas y ejercicios.

Para esto, se planea realizar pruebas dinámicas que nos aseguren la precisión y consistencia de datos ingresados por los usuarios en tiempo real.

Las pruebas considerarán diversos escenarios en los que interactúan diferentes roles de usuario, asegurando que las restricciones y permisos se apliquen correctamente. Para lograrlo:

- **Pruebas funcionales:** Validar que las funcionalidades estipuladas, en el levantamiento de requisitos que estas sean correctas y que cumplan con el comportamiento esperado. Además, se evaluarán todos los nodos pertenecientes al proyecto virtX.
- **Pruebas de caja negra:** Validar las funciones mediante el llenado de formularios.
- **Pruebas de caja blanca:** Validar que la solución a cada requisito funcional sea la correcta, evaluando su rendimiento y asegurando que cumpla con los estándares esperados.

Las especificaciones de las pruebas se encuentran en el Anexo de Elementos de prueba, página 6.

Referencias

- Norma ISO 29119
- Arquitectura Cliente-Servidor. (s. f.). https://reactiveprogramming.io/blog/es/estilos-arquitectonicos/cliente-servidor
- Descripción del sistema. Anexo B, pagina 17.
- Listado de historias de usuario. Anexo B, página 18.
- Recursos para el desarrollo. Anexo B, página 29.
- Arquitectura del sistema. Anexo B, página 30.
- Diagrama de componentes. Anexo B, página 31.
- Descripción de los componentes. Anexo B, página 32.
- Diagrama de base de datos. Anexo B, página 33.

Glosario

- CRUD: Acrónimo de Create (Crear), Read (Leer), Update (Actualizar) y Delete (Borrar).
- ISTQB: The International Software Testing Qualifications Board (Junta Internacional de Calificaciones de Pruebas de Software)

A. Contexto de las Pruebas

I. Proyecto / Subprocesos de Prueba

Hasta el momento ya se conocen los requisitos funcionales necesarios para el sistema VirtX, a continuación se describirán los módulos que contendrá la aplicación y cómo es que los requisitos se implementan en ellos.

Módulo de Autenticación

1. Registro
2. Iniciar sesión

Módulo de Recorridos

1. Elegir categorias
2. Elegir recorridos
3. Elegir un recorrido
4. Visualizar información de un recorrido
5. Iniciar un recorrido
6. Pausar y reanudar video

Módulo de Navegación

1. Desplazarse entre menús
2. Seleccionar categorias
3. Seleccionar recorridos
4. Salir al menú principal

II. Elementos de Prueba

A continuación se muestran las pruebas a realizar en los distintos módulos. En la sección de anexos A se encuentran ejemplos de las tablas de casos de prueba.

Módulo de Autenticación
Consta de dos vistas, una para el inicio de sesión y otra para el registro, con el fin de validar las credenciales de los usuarios y que puedan acceder al sistema.

Pruebas a realizar:

- Validar que solo los usuarios con credenciales correctas puedan iniciar sesión.
- Confirmar que las credenciales incorrectas o no registradas generen un mensaje de error adecuado.
- Verificar que los datos ingresados en el registro se guarden correctamente en la base de datos.
- Probar el flujo completo de inicio de sesión y registro, asegurando un proceso fluido y sin interrupciones.
- Validar el manejo de sesiones, incluyendo el cierre de sesión seguro.

Módulo de Recorridos
Módulo encargado de llevar a cabo los recorridos, principalmente la elección y la información de la cual el sistema hará uso.

Pruebas a realizar:

- Comprobar que el sistema permite cargar todos los recursos de un recorrido en específico.
- Validar que los recorridos se reproduzcan en la fase demostrativa.
- Validar que el usuario pueda pausar y reanudar el recorrido.
- Validar que el recorrido entre al modo interactivo al final de la fase demostrativa.

Módulo de Navegación

Módulo encargado de realizar el desplazamiento entre las vastas pantallas de la aplicación, encargada de cargar, renderizar y poner a disposición.

Pruebas a realizar:

1. Confirmar que las pantallas esperadas son renderizadas.
2. Confirmar que los elementos estén conectados de manera efectiva a sus funciones correspondientes.
3. Verificar que el motor de Unity funciona de manera apropiada cargando las pantallas esperadas

III. Alcance de la Prueba

Lo que se busca probar esencialmente son los factores funcionales del sistema, es decir, el comportamiento de manera que se describe en los artefactos en la etapa del diseño.
Los factores de calidad (no funcionales), como el rendimiento, la seguridad informática y la usabilidad no serán tomadas en cuenta en el documento presente, debido a que se profundizará más en ellos en otro plan de pruebas exclusivo para estos requisitos.

B. Comunicación de las Pruebas

La comunicación de las pruebas se realizará a través de **Jira** y **TestLink**, aprovechando las capacidades de cada herramienta. **Jira** nos permitirá tener un seguimiento detallado de los fallos identificados, gestionar tareas relacionadas y la generación de reportes diarios sobre el avance de las pruebas, mientras que **TestLink** será utilizado para documentar y estructurar los casos de prueba, así como para registrar los resultados obtenidos en cada ejecución de manera organizada.

Si se encuentran defectos críticos, se notificará de inmediato al líder de pruebas para tener dicho incidente como prioridad. Para casos de alta urgencia, se programarán reuniones virtuales extraordinarias para corregir defectos rápidamente. Al final de cada ciclo de pruebas, se harán reuniones de revisión para analizar y discutir los resultados, usando la información de TestLink y Jira para planificar los próximos pasos.

C. Registro de Riesgos

El proceso de pruebas lleva riesgos que podrían afectar la ejecución, calidad y resultados esperados de las pruebas. El impacto y la probabilidad se determinan con la escala de 1 al 5, donde 5 es el más alto. A continuación se mostrarán los posibles riesgos con sus aspectos importantes y su respectivo plan de mitigación.

No.	Riesgos	Probabilidad	Impacto	Severidad (Probabilidad* Impacto)	Plan de mitigación
1	Problemas al integrar los módulos (Retrasos, inconsistencias de datos o posibles errores en funcionalidades con dependencia en otras)	3	5	15	Realizar pruebas de integración de forma temprana durante el desarrollo.
2	Falta de experiencia con las librerías o frameworks a utilizar	2	3	6	Realizar cursos para la comprensión y resolución de problemas
3	Fallo de herramientas de hardware para el desarrollo	1	5	5	Realizar el mantenimiento adecuado o en el peor de los casos adquirir un nuevo equipo
4	Fallos en la base de datos	2	3	6	Ejecutar pruebas de estrés y realizar validación de datos cada que se interactúe con la base de datos
5	Cobertura incompleta de casos de prueba	1	4	4	Revisar minuciosamente los casos de prueba y priorizar zonas críticas como el módulo de autenticación y la gestión de las citas

D. Estrategia de Prueba

I. Subprocesos de prueba

Los tipos de pruebas que se realizarán son las siguientes:

- Pruebas funcionales
- Pruebas de caja negra
- Pruebas de caja blanca

Estas pruebas se han elegido con el fin de cubrir la mayoría del software, sometiendolo a una análisis riguroso de prueba y error, evaluando tanto las funcionalidades individuales para validar su correcto funcionamiento, tanto como si el sistema cumple con las expectativas del cliente. La realización de las pruebas permite llevar un análisis profundo del sistema lo cual contribuye a un sistema de calidad.

II. Entregables de Prueba

Para cada subproceso de pruebas se deben generar los siguientes artefacto y documentación:

- Casos de prueba
- Reporte de Estado de las pruebas
- Informe de Finalización del subproceso de prueba.

III. Técnicas de diseño de Prueba

En este punto se identifican las técnicas que se utilizara para el diseño de las pruebas:

- Particionamiento equivalente
- Análisis de Valores Límite
- Tabla de decision

IV. Criterio de Finalización y Prueba

Las pruebas deben ejecutarse de manera rigurosa, garantizando un cumplimiento mínimo del 70% de todos los requisitos del sistema. Esto implica una cobertura exhaustiva de las funcionalidades y escenarios de uso, así como una verificación detallada de cada resultado. Además, los procedimientos de prueba deben estar bien definidos y ejecutados sin incidencias críticas que comprometan la calidad del software.

V. Métricas

Las métricas que nos permitirán realizar las mediciones de manera efectiva durante la ejecución de las pruebas.

- Número de incidentes por categoría: Para llevar un control de los errores ocurridos durante la realización de las pruebas, esto siendo de cada módulo del sistema, lo que nos permitirá identificar las áreas críticas en el sistema.
- Número de incidentes resueltos por categoría: Estos son la cantidad de fallos solucionados por categoría, con el fin de compararlos con los números de incidentes.

VI. Requisitos del entorno de Pruebas

A. Ambiente de pruebas

Sistemas Operativos	Windows 11, Android, iOS
Ambiente de Desarrollo	Visual Studio Code, Unity HUB

B. Herramientas de Pruebas

Herramienta	Función
Test Link	Herramienta de gestión de pruebas de código abierto que sincroniza la especificación de requisitos y la especificación de prueba.
PostMan	Programa que permite enviar solicitudes personalizadas a una API y recibe las respuesta de esta misma a gran detalle.
Nunit	NUnit es un framework open source de Pruebas de unidad para Microsoft .NET y Mono.

VII. Re-testing y regresión de las Pruebas

Se realizan las pruebas de confirmación o mejor conocidas como re-testing para poder satisfacer los criterios de finalización plasmados anteriormente -En específico el cumplimiento de la mayoría de requisitos, con un mínimo aceptable del 70% de los requisitos totales .En el caso de las pruebas de regresión se realizarán una vez se hayan realizado 2 ciclos de prueba.

VIII. Criterios de Suspensión y Reanudación

A. Criterios de suspensión

- El sistema no cumple con las funcionalidades especificadas en el plan de pruebas y en la documentación
- El sistema colapse en medio de un análisis, por la realización de una acción en específico
- La funcionalidad del sistema no de los resultados esperados

B. Criterio de reanudación

- Se reanudan las pruebas cuando los defectos que ocasionaron los errores hayan sido solucionados.
- Se reanudan las pruebas al haber llegado a algún acuerdo con el cliente.

E. Actividades y Estimados de Prueba

Las pruebas se dividirán en las siguientes actividades, según lo que se creyó más conveniente para este proyecto y basándonos en el proceso de pruebas básico de la ISTQB:

1. Determinar el alcance y objetivos de las pruebas.
2. Medir y analizar los resultados.
3. Documentar los criterios de salida.
4. Establecer prioridad de los módulos a probar basados en los casos de uso.
5. Diseñar las pruebas de cada módulo utilizando combinaciones específicas de datos de prueba, acciones y resultados.
6. Establecer el entorno de pruebas.
7. Primer ciclo de ejecución de las pruebas.
8. Revisión y corrección de resultados.
9. Segundo ciclo de ejecución de las pruebas.
10. Segunda revisión y corrección de resultados.
11. Informe de reporte de estado de las pruebas en comparación a los criterios de salida documentados previamente.

F. Personal

I. Roles, Actividades y Responsabilidades

La matriz RACI a continuación muestra qué rol está implicado en qué actividades (de las mencionadas en el punto anterior) y cuál es su relación.

Rol	1	2	3	4	5	6	7	8	9	10	11
Test Lead	R	I	I	R	C	I	A	R	A	R	R
Test Designer	C	R	R	I	R	R	C	I	C	I	I
Test Executor	I	I	I	I	I	I	R	I	R	I	I

R: Responsable
A: Autoridad
C: Consultor
I: Informado

Recurso	Nombre	Descripción
RZ	Roberto Zuñiga Roman	Test Executor, Test Designer
DS	Diana I. Solano Uscanga	Test Lead

II. Necesidades de Contratación

No hay necesidad de contratación por el tipo de proyecto que se está realizando.

III. Necesidades de Entrenamiento

Capacitación básica para la familiarización con los conceptos básicos de Unity, y con la librería de Meta XR All in one SDK, estimando en total 20 horas.

G. Cronograma

Etapa/Semana	1	2	3	4	5	6	7	8	9	10
Planificación										
Control										
Análisis										
Diseño										
Implementación										
Ejecución										
Evaluación										

Anexo A. Casos de prueba

Autenticación

Secuencia	Caso de prueba	Entrada	Resultado esperado	Estado (Ok, Fail)
C1	Logueo del sistema	{ “email” : “zhrine1103@gmail.com”, “password”: “admin321” }	Muestra la pantalla de bienvenida del sistema con la opción para elegir las categorías.	-
C2	Registrar una cuenta sin especificar el correo electrónico	{ “user_name: “Roberto Z. Roman”, “email” : , “password” : “Zhr1n3%!” }	Muestra un modal con el mensaje “Por favor, rellene los campos faltantes”	-
C3	Registrar una cuenta con un correo electrónico ya usado	{ “user: “Diana Iratze”, “email” :”zhrine1103@gmail.com” , “password” : “ma11911” }	Muestra un modal con el mensaje “Correo electronico ya en uso”	-
C4	Registrar con una cuenta con un correo electrónico válido	{ “user: “Jade ”, “email” : “312j4d3@gmail.com”, “password” : “12345678” }	Muestra la pantalla de bienvenida del sistema con la opción para elegir las categorías.	-
C5	Registrar una cuenta con un correo no válido	{ “user: “Julian Ventura”, “email” : “toorbigmailcom” , “password” : “Zhr1n3%!” }	Muestra un modal con el mensaje “Correo Electrónico no válido”	-

Recorrido

Secuencia	Caso de prueba	Entrada	Resultado esperado	Estado (Ok, Fail)
C1	Empezar video	true	Se pausa el video de prueba en ese instante	-
C2	Reanudar video	false	Se reanuda el video de prueba en ese instante	-
C3	Terminar video	true	Termina el video y carga una escena nueva	-

Navegación

Secuencia	Caso de prueba	Entrada	Resultado esperado	Estado (Ok, Fail)
C1	Cargar menú principal	0	Muestra un mensaje en el log de unity que indica que las escenas coinciden	-
C2	Cargar categorías disponibles	2	Muestra un mensaje en el log de unity que indica que las escenas coinciden	
C3	Elegir la categoria esperada	3, 3	Muestra un mensaje en el log de unity que indica que la pantalla de la categoría indicada se compiló de manera exitosa	-
C4	Elegir el recorrido esperado	5, 5	Muestra un mensaje con el texto “Rutina creada exitosamente”.	-
C5	Iniciar recorrido organizacional	1	Muestra la primera etapa del recorrido organizacional.	-

Anexo B. Sistema

Descripción del problema

VirtX es una aplicación la cual prioriza la experiencia de usuario, esta misma se enfoca en plasmar con ayuda de la realidad virtual una experiencia lo más cercana posible a un recorrido el cual puede ser en zonas culturales, enfocadas al apartado educativo y organizacional, esto para que las personas en una situación de vulnerabilidad económica, personas que posean el síndrome de asperger las cuales padezcan ansiedad social en espacios con una gran cantidad de personas, y personas sedentarias las cuales por motivos personales, culturales, educacionales o médicos no puedan salir de sus viviendas o trasladarse a otros sitios.
Esto mediante la ayuda de lentes de realidad virtual en este caso con el modelo específico “Meta Quest 3”, utilizando entornos interactivos para contribuir al recorrido aportando, datos relevantes a este y la capacidad de elegir entre varias opciones de camino permitiendo el completo control del usuario en la experiencia.
VirtX de la misma manera busca a la vez brindar una experiencia inclusiva a gente que sufra de foto sensibilidad, empleando prácticas al momento de realizar el diseño de la interfaz de usuario, para brindar una mejor experiencia. Por otro lado, el apartado de la funcionalidad busca ser inclusivo para que las personas con alguna discapacidad auditiva o de lenguaje puedan usar el sistema como cualquier otro usuario sin crear distinciones.

Funcionalidades:

- Abarcar los apartados educacionales, culturales y organizacionales
- Inicio de Sesion
- Registro de cuenta
- Elección de categoría
- Elección de recorridos
- Iniciar recorridos
- Pausar y reanudar recorridos
- Elegir etapas dentro del recorrido
- Interactuar en las secciones interactivas
- Terminar recorridos

Recursos

Para el desarrollo de *Virtx*, se planea utilizar dos computadoras y unos lentes MetaQuest3 como recursos principales. Inicialmente, se había considerado implementar el proyecto en Unreal Engine apoyándonos con el lenguaje de programación de C++; Sin embargo, debido a la necesidad de que la solución sea una aplicación disponible para los meta Quest 3 y por la experiencia del equipo anterior, se decidió optar por **Unity** cómo programa principal para el desarrollo en el entorno para VR y **Actix-Web** como framework backend de Rust, el cual es el encargado de manejar funciones como la autenticación y los cambios a la información de los usuarios, permitiendo tener un desarrollo flexible.

En cuanto al tiempo estimado para el desarrollo, se espera un total de **105 horas**. A esto se agregan las horas necesarias para el aprendizaje de Unity y Rust, la configuración del entorno de desarrollo, y las etapas de análisis y diseño, lo que lleva a un estimado total aproximado de **250 horas**.

Arquitectura

El modelo cliente-servidor, también conocido como “principio cliente-servidor”, es un modelo de comunicación que permite la distribución de tareas dentro de una red de ordenadores, se basa en el intercambio de información entre dos entidades principales: el cliente y el servidor.

El modelo cliente-servidor tiene algunos rasgos característicos. Hay una clara distribución de tareas entre los clientes y los servidores. El servidor es el responsable de proporcionar los servicios. Se encarga de ejecutar los servicios solicitados y entrega la respuesta esperada. El cliente, en cambio, utiliza y solicita los servicios proporcionados. Finalmente, recibe la respuesta del servidor.

En el modelo cliente-servidor, un servidor sirve a varios clientes y, por ende, procesa múltiples peticiones de diferentes clientes. Para ello, presta su servicio de forma permanente y pasiva. Por su parte, el cliente solicita activamente los servicios del servidor e inicia las tareas del servidor.

Un servidor es un hardware que proporciona los recursos necesarios para otros ordenadores o programas, pero un servidor también puede ser un programa informático que se comunica con los clientes. Un servidor acepta las peticiones del cliente, las procesa y proporciona la respuesta solicitada. También existen diferentes tipos de clientes. Un ordenador o un programa informático se comunica con el servidor, envía solicitudes y recibe respuestas del servidor.

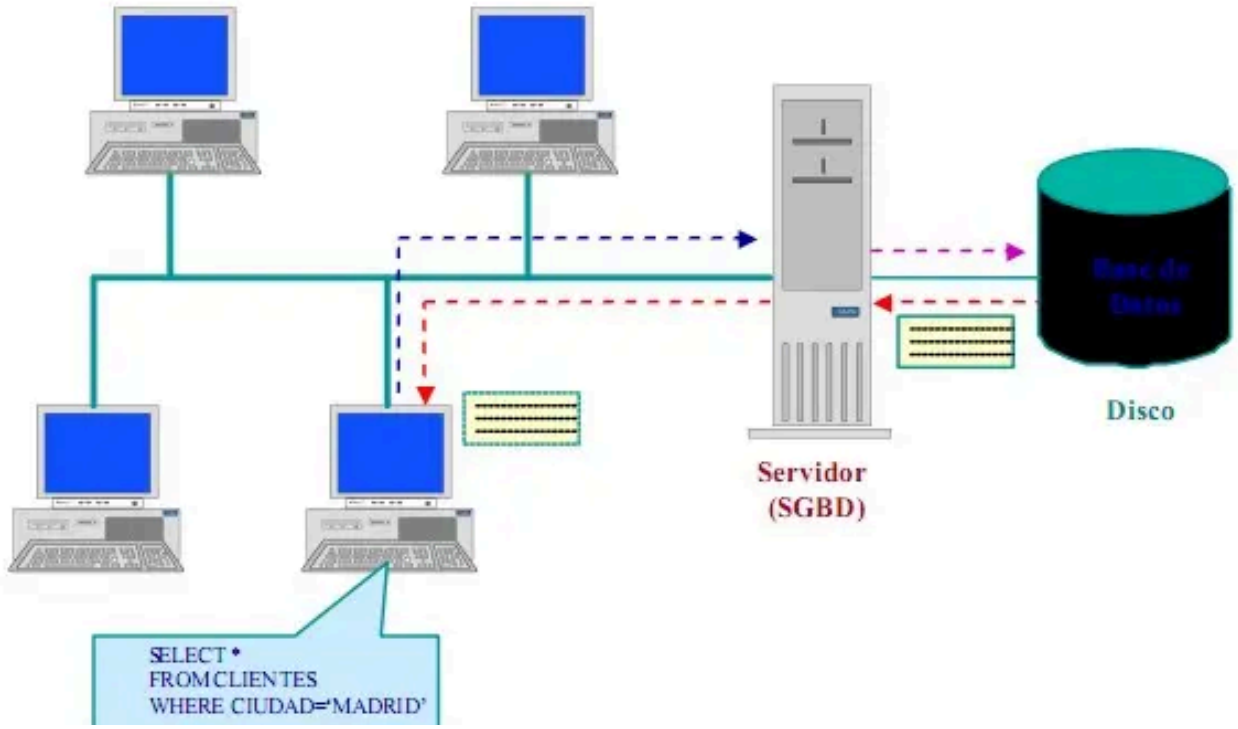
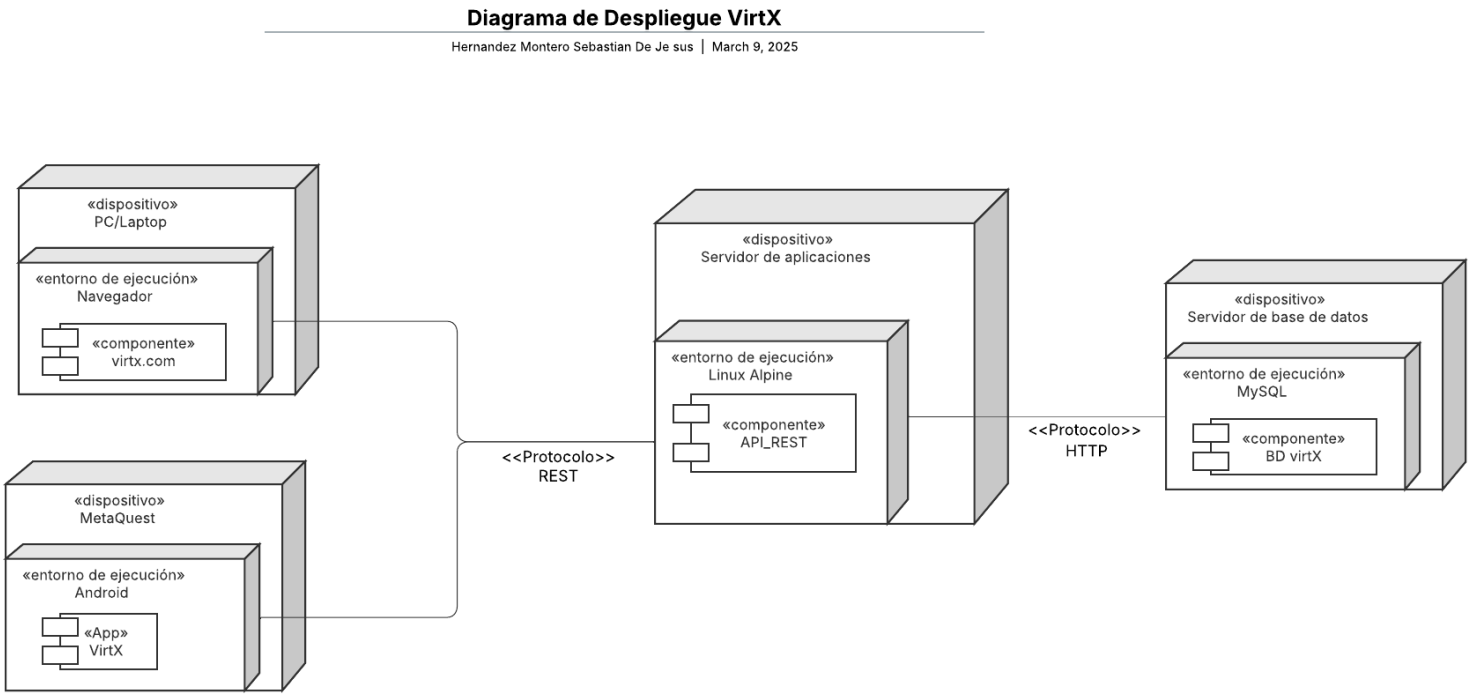


Diagrama de despliegue



Descripción de los nodos y artefactos

La aplicación principal de virtX se encuentra en el nodo de dispositivo Meta Quest 3, el cual contiene dentro un entorno de ejecución en android pues este es el principal sistema operativo de los meta quest 3, esta aplicación se comunica con un protocolo API REST en HTTP con el API Rest el cual es un componente que está localizado en un Servidor de aplicaciones, en un entorno de ejecución de linux Alpine, este se encarga de recibir las peticiones y realizar operaciones, principalmente funciones que tengan que ver con el usuario (Inicio de sesión, Registro de usuarios, Modificación de datos de usuario principalmente) y esta se comunica con el protocolo HTTP a la base de datos VirtX la cual se encarga del almacenamiento de la información del usuario.

La página web se encarga de proporcionar información a los usuarios, acerca del desarrollo de la aplicación con la finalidad de dar un enfoque transparente a nuestros usuarios la cual próximamente se tiene estipulado que sirva para consultar los datos de la cuenta esperada y pueda modificarlos de ser necesario esto mediante un protocolo HTTP con ayuda de la API REST.

Bibliografia

¿Cómo funciona el modelo cliente-servidor? (s/f). IONOS Digital Guide. Recuperado el 10 de marzo de 2025, de https://www.ionos.mx/digitalguide/servidores/know-how/modelo-cliente-servidor/

Gomez, C. (2024, julio 23). *Qué es la arquitectura cliente servidor y cómo funciona*. Daemon4. https://www.daemon4.com/empresa/noticias/arquitectura-cliente-servidor/