

Proyecto: TVet Marca: TPet

El proyecto va dirigido a todos aquellos dueños de perros que no tienen el tiempo o la capacidad de trasladar a su mascota a una veterinaria en caso de enfermedad o preocupación. Nuestro proyecto busca facilitar la asistencia médica y cuidado de las mascotas.

Lista de requerimientos

Prioridad 1

- Planes de subscripción
- Perfiles
- Historial médico
- Recordatorios de citas
- Consultas virtuales con cita previa
- Servicio express de farmacia
- Transcripción inteligente

Prioridad 2

- Reseñas de veterinarias
- Descuentos de las veterinarias

Prioridad 3

- Reseñas de usuarios (clientes)
- Teleseguimiento y tratamientos
- Orientación y nutrición<

Diagrama de arquitectura

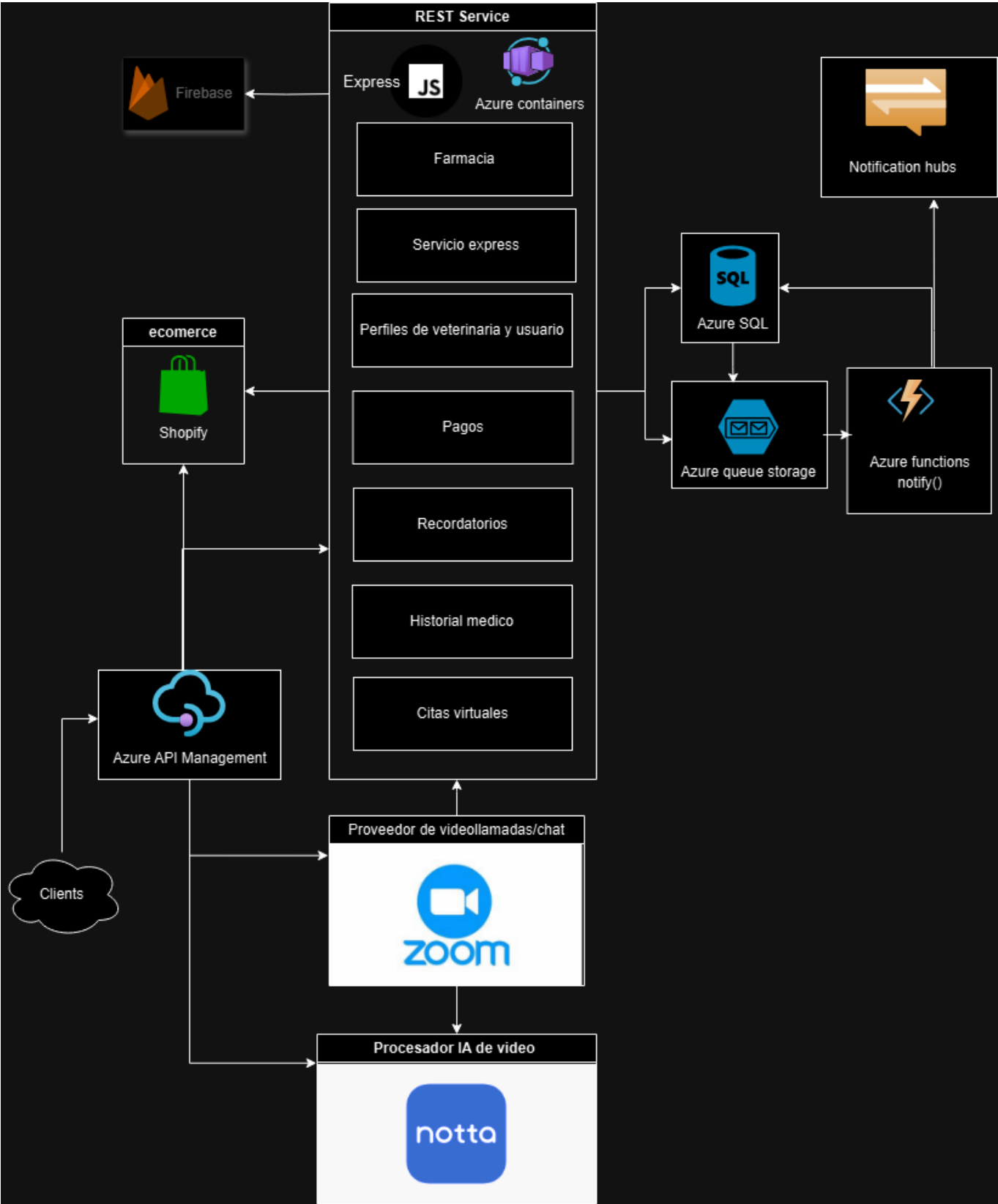


Diagrama de bases de datos datos



Scripts para crear base de datos

Problem Statement y Storyboard

Convencer al dueño de la mascota de agendar una cita veterinaria virtual.

Buscar Veterinaria





Cita Virtual



Historial de mascota



Asistencia con IA



Paquetes de suscripción

Buscar Veterinaria



Veterinaria Pet Shop



VETERINARY



Nuestra veterinaria cuenta con 20 años de experiencia en la atención de distintas mascotas

Precio consulta: 20,000

Reserva



PetAI Care



CAREVET

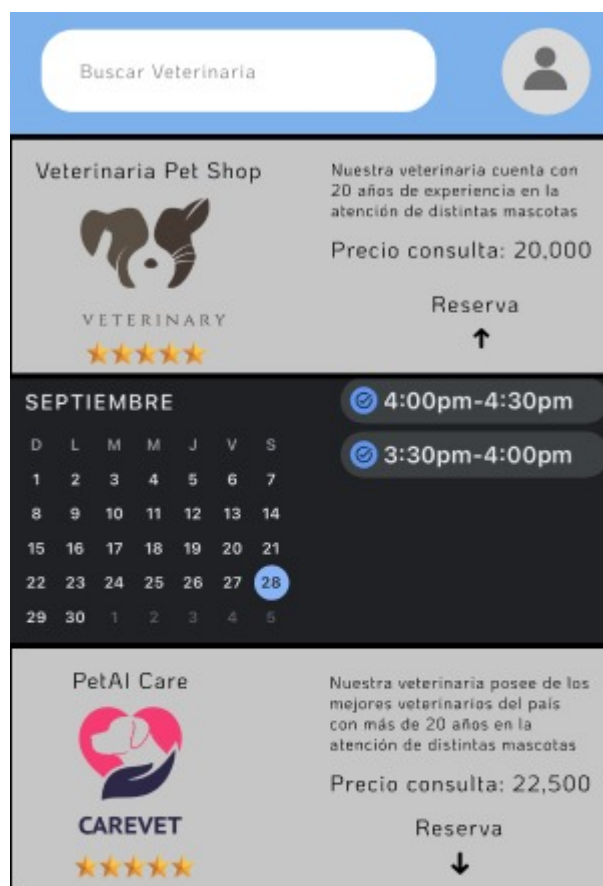


Nuestra veterinaria posee de los mejores veterinarios del país con más de 20 años en la atención de distintas mascotas

Precio consulta: 22,500

Reserva





Requerimientos no funcionales

Performance

1. ¿Cuáles son los tiempos de respuesta esperados para diferentes funciones en condiciones de carga normal y máxima?

Teniendo en cuenta la estimación de 1.900 clínicas y 76.000 citas realizadas mensualmente a través de la aplicación: Cálculo de usuarios pico y normales: Citas totales por mes: 76.000 citas. Citas promedio por día: $76.000 / 30 = 2533$ citas por día. Usuarios pico: Suponiendo un periodo pico de aproximadamente 2 horas por día donde se reservan o realizan el 25 % de las citas diarias: 25 % de las citas diarias = $2533 \times 0,25 = 633$ citas. Los usuarios pico (suponiendo una consulta por video + uso administrativo) podrían duplicar esta cifra (para usuarios que reservan y veterinarios que realizan): Usuarios pico simultáneos = $633 \times 2 = 1266$ Usuarios normales: suponiendo una distribución constante a lo largo de una jornada laboral de 8 horas: $(2533 - 633) / 6 = 317$ usuarios simultáneos en horas normales.

Bajo cargas normales (317 usuarios): El login y la autenticación su respuesta sería entre 1 y 2 segundos. El inicio de una cita virtual en menos de 5 segundos. Bookear la cita entre 2 y 3 segundos. Acceder al historial de la mascota entre 1 y 2 segundos. Bajo pico de cargas (1266 usuarios): El inicio de una cita virtual podría llegar a los 7 segundos. Las otras operaciones podrían aumentar su tiempo en un 50%.

2. ¿Cuántos usuarios simultáneos debe admitir el sistema?

Carga normal entre 300 a 500 usuarios simultáneos. Carga máxima: Escalable a 1500 usuarios simultáneos, especialmente durante las horas pico de la clínica.

3. ¿Cuáles son los puntos de referencia de rendimiento para operaciones críticas?

Consultas por video: Mantener la latencia por debajo de los 150 ms, con un tiempo de actividad del 99,9%. Operaciones de base de datos: Consultas de registros médicos y programación de citas completadas en 500 ms con carga normal y en menos de 1 segundo durante las horas pico.

Scalability

4. ¿Como se manejara la escalabilidad del sistema en caso de un aumento en la carga o volumen de datos?

El sistema debe de ser capaz de escalar con facilidad para mantener la posibilidad de expandirse a más países en un futuro. Actualmente, estimamos que aproximadamente al mes se procesarán 76 000 citas y por lo tanto, 2533 citas diarias. Por lo que se debe hacer uso del servicio de Azure Containers (2 vCPU, 2 GB de memoria) (\$39 por mes), Azure API management (\$187 por mes) y Azure SQL Database (4 vCores, 75 GB de almacenamiento) (\$148 por mes). Teniendo en cuenta lo anterior, para escalar el sistema se deben incrementar el número de contenedores del app o considerar una arquitectura de microservicios y para la base de datos se debe de aumentar los vCores de la base de datos y almacenamiento o considerar un motor como PostGres para soportar replicación e implementar una arquitectura master-slave.

5. ¿Hay algun requerimiento especifico para diferentes componentes?

Para la transcripción de llamadas se está utilizando el sistema Notta, en el cual se debe de comprar una suscripción con un plan Business (\$28 al mes por suscripción individual) el cual provee minutos ilimitados por mes, procesamiento de llamadas de hasta 5 horas y generación de notas. A pesar de que el servicio presenta minutos ilimitados de transcripción, este tiene un tope de videos por mes el cual es 200 y nosotros debemos procesar aproximadamente 76 000, por lo que estimamos que según nuestro caso, el costo mensual llega a los \$10,640. Considerando lo anterior, cada 200 videos extra que se deban de procesar, equivalen a un aumento de \$28 de la suscripción de Notta.

Reliability

6. ¿Cuál es el tiempo de inactividad aceptable para el sistema?

El sistema debe de estar disponible el 99.9% del tiempo. Es decir, la aplicación solo puede tener un tiempo de inactividad de 43 minutos por mes como máximo. Esto se puede garantizar por medio de 'Azure containers', 'Azure SQL' y 'Azure Queue Storage' ya que proporcionan alta disponibilidad.

7. ¿Cómo debe manejar el sistema los fallos y asegurar la integridad de los datos?

Para que el sistema maneje los fallos y asegure la integridad de los datos, se utilizará 'Azure SQL' que garantiza las transacciones ACID. Así mismo, para asegurar que no se pierda la cola de los mensajes y los mensajes durante los fallos, se utilizará 'Azure Queue Storage'. Finalmente, en caso de fallo los datos críticos deben de estar asegurados, por lo que se utiliza 'Firebase' para garantizar la redundancia y el acceso a dichos datos.

8. ¿Cuáles son los procedimientos de respaldo y recuperación?

La base de datos en 'Azure SQL' será respaldada automáticamente de manera diaria. Los logs y archivos importantes serán almacenados en 'Azure Queue Storage' con una retención de 30 días para asegurar la recuperación en caso de fallos. Para la base de datos y las colas, el tiempo de recuperación debe de ser de 2 horas y el punto de recuperación de 15 minutos.

Availability

9. ¿Cuáles son los requisitos de tiempo de actividad del sistema?

El sistema requiere de un tiempo de actividad del 99,9 %, por lo que la aplicación puede estar aproximadamente 43 minutos inactiva por mes. Esto genera una alta confiabilidad tanto para los veterinarios como para los dueños de mascotas. Esto se puede garantizar por medio de 'Azure containers', 'Azure SQL' y 'Azure Queue Storage' ya que proporcionan alta disponibilidad. **10. ¿Existen horarios específicos en los que el sistema debe estar disponible sin fallas?**

Horario comercial para veterinarias de 7 a. m. a 7 p. m los 7 días de la semana, el sistema debe estar disponible durante el horario comercial normal de los veterinarios, ya que es cuando se producirán la mayoría de las consultas y reservas. Para asistencia de emergencia también se debe garantizar la disponibilidad fuera del horario comercial para consultas de emergencia para manejar tráfico ocasional fuera de horas pico fuera del horario comercial estándar. Campañas nacionales de salud de mascotas o situaciones de emergencia para mascotas ya que durante estos eventos, debe asegurarse de que el sistema pueda manejar posibles aumentos de usuarios, lo que hace que el tiempo de actividad sea crítico. Cuando se presenten promociones o descuentos por día, ya que muchos usuarios van a querer aprovecharlos y buscarían agendar cita ese día, aumentando así la concurrencia de usuarios.

Security

11. ¿Qué requerimientos de seguridad hay para el almacenamiento y transmisión de datos?

La información personal de todos los usuarios (dirección, correo electrónico, teléfono, método de pago) deben de estar encriptados en la base de datos. Al igual que la tabla pagos.

12. ¿Cómo se deben gestionar la autenticación y autorización de usuarios?

Se debe utilizar el servicio externo Firebase para la autenticación y autorización de cuentas de todos los usuarios.

13. ¿Existen requisitos de cumplimiento relacionados con la seguridad de los datos?

Se utilizará el estándar internacional ISO/IEC 27001 para la protección de información.

Usability

14. ¿Cuáles son los estándares de usabilidad para la interfaz de usuario?

La interfaz de usuario (que será hecha en 'React') debe ofrecer una experiencia intuitiva y responsive, al igual que responder a las interacciones del usuario en menos de 1 segundo. En este caso, se utilizará 'Express.js' en el backend para procesar solicitudes y 'Firebase' para almacenar perfiles y datos de usuario.

15. ¿Cómo debe el sistema acomodar a los usuarios con discapacidades?

El sistema debe de incluir soporte para lectores de pantalla, compatibilidad con la navegación por teclado y la interfaz debe de tener un contraste de diseño adecuado. Además, las videollamadas a través de Zoom deben ofrecer opciones de subtítulos automáticos en caso que el usuario tenga discapacidades auditivas. Estos serán generados por 'Notta'.

16. ¿Existen requisitos específicos para la capacitación de usuarios y documentación?

Como requerimientos para la documentación y la capacitación de usuarios, se tiene que, dentro de la aplicación se tendrá un video tutorial acompañado por guías escritas. Esto debido a que los videos se consideran mucho más amigables y fáciles de seguir que un documento escrito, pero en caso que el usuario lo prefiera también tendrá a disponibilidad las guías escritas. En el caso de los administradores, se dará una documentación técnica por medio de 'Azure API Management' ya que esto facilitará la supervisión y gestión de la plataforma.

Maintainability

17. ¿Qué tan fácil debería ser actualizar y modificar el sistema?

Contenedorización con Azure Containers: dado que la aplicación se implementa a través de Azure Containers, las actualizaciones se pueden realizar creando nuevas imágenes de contenedor. Esto proporciona flexibilidad en el control de versiones. Azure Functions para los componentes sin servidor, las actualizaciones son fáciles y se pueden realizar sin afectar otras partes del sistema, lo que garantiza una escalabilidad. Azure API Management permite administrar versiones de API, lo que simplifica la implementación de nuevas versiones y la discontinuación gradual de las antiguas sin interrumpir el servicio.

18. ¿Cuáles son los requisitos para el registro y monitoreo del rendimiento del sistema?

Firebase proporciona herramientas como Firebase Crashlytics para monitorear fallas y rendimiento del sistema, especialmente útiles para identificar y solucionar problemas en tiempo real. Además de Firebase Analytics para realizar un seguimiento de registros detallados del comportamiento del usuario, los patrones de uso de la aplicación y el estado del sistema. Azure Log Analytics para consultar, analizar y visualizar los datos, lo que garantiza la trazabilidad total de las operaciones del sistema.

19. ¿Cómo debe gestionar el sistema el control de versiones y la implementación?

Control de versiones con Git para el control de código fuente, y GitHub para el control de versiones. El deploy de la aplicación usando Azure Containers.

Interoperability

20. ¿Cómo debería el sistema integrarse con el software y hardware existentes?

- **Zoom:** El sistema se conectará con zoom mediante su API de videollamadas para las consultas virtuales.
- **Notta:** Para la transcripción de vídeos, el sistema deberá de utilizar la API de Notta.
- **Shopify:** El ecommerce será integrado mediante la implementación de shopify, en donde se comunicarán mediante un API rest
- **Servicios de azure (API management, Azure containers, Azure SQL Server, Azure queue storage, Azure functions, Azure notification hub):** El sistema se conecta a estos servicios mediante los endpoints y direcciones brindadas por Azure.

21. Are there any standards or protocols that the system must adhere to?

Rest API: Para que todas las integraciones del sistema sean compatibles y faciles de leer. ISO/IEC 27001: Para garantizar la seguridad de la información.

Compliance

22. ¿Con qué requisitos legales y regulatorios debe cumplir el sistema?

El sistema debe cumplir con la 'Ley de Protección de Datos' perteneciente a Costa Rica. Teniendo en cuenta que en un futuro se podrá expandir el dominio de la aplicación, el sistema deberá cumplir de igual manera con las leyes de protección de datos de los países de centroamérica y Estados Unidos. Las comunicaciones y datos médicos deben ser cifrados de extremo a extremo, especialmente en servicios como 'Zoom' y el historial médico almacenado en 'Azure SQL' y 'Firebase'.

23. ¿Existen estándares específicos de la industria que se deban seguir? El sistema debe seguir los estándares de protección de datos médicos y asegurar que las videollamadas y transacciones de información médica cumplan con las regulaciones para proteger la privacidad del paciente. 'Azure SQL' y 'Firebase' son plataformas que ya cumplen con estos estándares.

Extensibility

24. ¿Cómo debería diseñarse el sistema para dar cabida a futuras mejoras?

API REST para manejar diferentes funcionalidades dentro de un solo servicio. Esto permite una administración y un desarrollo más sencillos. El código modular dentro del servicio API REST, lo que le permitirá agregar nuevas características o funcionalidades sin afectar significativamente el código existente. Control de versiones de API (/api/v1/, /api/v2/) para permitir futuras mejoras o cambios importantes mientras mantiene la compatibilidad con versiones anteriores para los clientes. **25. ¿Existen áreas específicas donde la extensibilidad es fundamental?**

La API REST para que se integre fácilmente con servicios de terceros para así proporcionar un marco flexible para puntos finales o enlaces adicionales. Esto mediante el uso del API de Notta para la IA y Shopify para integración de ecommerce. Se considera agregar puntos finales para análisis e informes, lo que habilitará capacidades futuras como el seguimiento del comportamiento de los usuarios, las tendencias de citas o el análisis de ingresos.

Localization

26. ¿Cuáles son los requisitos para soportar múltiples idiomas y regiones?

Inicialmente el frontend debe de estar completamente en español latinoamerica, esto debido a que nuestra audiencia objetivo es la población de Costa Rica, aunque conservando la modificabilidad del codigo para cuando la aplicación se extienda a mas paises.

27. ¿Cómo debería el sistema manejar diferentes formatos de fecha, hora y moneda?

El sistema debe aceptar tanto colones como dólares, ya que a pesar de que el sistema vaya dirigido solo a costarricenses, es muy común que estos ganen salarios en dólares y por lo tanto algunos pueden preferir pagar en dólares. Considerando que inicialmente el sistema solo estará disponible en Costa Rica, tanto la hora como la fecha deben de estar con (UTC - 6:00) America Central.

Documentation

28. ¿Qué documentación se requiere para usuarios, administradores y desarrolladores?

El sistema requiere de la siguiente documentación:

- Usuarios: Guías de usuario accesibles desde la aplicación, explicando cómo agendar citas, consultar historial médico, consultar el perfil, acceder a las citas virtuales, utilizar la IA ('Notta') y realizar pagos.
- Administradores: Documentación técnica en 'Azure API Management' para manejar la plataforma, incluyendo el monitoreo de las APIs y el mantenimiento de la infraestructura en 'Azure Containers'.
- Desarrolladores: Manuales que detallen el uso de 'Express.js', 'Firebase', y las interacciones con 'Azure Functions' para las notificaciones.

29. ¿Cómo debe mantenerse y actualizarse la documentación?

La documentación debe estar almacenada en un repositorio de 'GitHub' y actualizada automáticamente cada vez que se realicen cambios importantes en la arquitectura del sistema. Además se debe realizar una revisión de la documentación cada 6 meses.