

GA7-220501096-AA1-EV01 INFORME TÉCNICO DE PLAN DE TRABAJO PARA
CONSTRUCCIÓN DE SOFTWARE

PRESENTADO POR
UBEIMAR BEDOYA ARANGO

PROFESOR
CRISTIAN ARIAS

SENA
REGIONAL CALDAS
ANÁLISIS Y DESARROLLO DE SOFTWARE 2721447
MEDELLÍN
2024

INTRODUCCIÓN SOFTWARE A DESARROLLAR

En la práctica actual de las ciencias veterinarias existen grandes vacíos en cuanto a la cobertura abarcada por los centros, consultorios y clínicas veterinarias que en su mayoría ejercen de manera privada, representando mas del 90% de la atención Médico veterinaria en el país. Sin embargo los servicios ofrecidos por estas empresas son de manera presencial, es decir que debemos asistir al centro veterinario para una consulta, servicio de estética, exámenes diagnósticos, entre otros. Todavía existe muy poca cobertura de servicio a domicilio de esta índole, teniendo en cuenta los cambios globales que se han venido presentando post-pandemia como es el caso de los trabajos remotos "Home office", lo que representa un reto para la mayoría de empresas del sector industria, comercio, producción y servicios que los ha obligado a migrar y adaptarse a esta nueva modalidad en los hogares. Es así como la implementación de servicios veterinarios a domicilio impartidos por profesionales calificados, representa una gran oportunidad tanto para Médicos como para técnicos en estética de mascotas, auxiliares veterinarios, paseadores caninos, centros de guardería y demás segmentos de la industria. Al no contar con un servicio a domicilio, los dueños de mascotas pueden enfrentar barreras significativas para acceder a la atención veterinaria de manera oportuna y cómoda. La implementación de un servicio a domicilio puede abordar estos problemas y mejorar la experiencia tanto para los dueños de mascotas como para sus animales.

Objetivo general

Evaluar la viabilidad y el impacto potencial de la implementación de una plataforma de servicios médicos veterinarios a domicilio en Colombia, considerando las necesidades y preferencias de los dueños de mascotas, la accesibilidad a la atención veterinaria y los beneficios para la salud y bienestar de los animales, con el fin de proporcionar una solución conveniente y personalizada para el cuidado de mascotas en el país.

Objetivos específicos

- Realizar encuestas y entrevistas para comprender las preferencias y desafíos que enfrentan los dueños de mascotas al acceder a servicios veterinarios, y determinar qué servicios específicos serían más valorados a través de una plataforma de atención a domicilio
- Investigar la infraestructura tecnológica requerida, la logística de visitas a domicilio y la disponibilidad de profesionales veterinarios dispuestos a brindar servicios a través de la plataforma.

- Realizar un seguimiento de casos piloto para evaluar cómo la atención a domicilio afecta el nivel de estrés de las mascotas, su comodidad y la calidad de la atención médica recibida en comparación con visitas a una clínica veterinaria tradicional.

HERRAMIENTA PARA UTILIZAR DE GIT EN MI PROYECTO

GitHub

Características:

1. **Colaboración Fácil:** GitHub facilita la colaboración en equipo, permitiendo a los desarrolladores trabajar en ramas separadas y fusionar cambios de manera eficiente.
2. **Seguimiento de Problemas:** Ofrece un sistema robusto para el seguimiento de problemas y la gestión de proyectos, lo que facilita la comunicación y el trabajo conjunto.
3. **Integración Continua:** Puede integrar GitHub con herramientas de integración continua para automatizar pruebas y despliegues.
4. **Comunidades y Redes Sociales:** GitHub es una plataforma social, lo que facilita la participación en comunidades, contribuir a proyectos de código abierto y compartir el trabajo.
5. **Herramientas de Revisión de Código:** Dispone de herramientas integradas para la revisión de código, lo que mejora la calidad del código y facilita la colaboración.

Razones para Utilizar GitHub:

- **Amplia Adopción:** GitHub es ampliamente utilizado en la comunidad de desarrollo, lo que facilita la colaboración con otros equipos y desarrolladores.
- **Variedad de Funcionalidades:** Ofrece una gama completa de herramientas para el desarrollo de software, desde control de versiones hasta gestión de proyectos y colaboración.
- **Integración con Servicios:** Se integra fácilmente con servicios de CI/CD, herramientas de desarrollo y servicios de alojamiento web.

GitHub es una elección sólida para proyectos de desarrollo de software debido a su amplia adopción, sus características robustas y su capacidad para facilitar la colaboración y la gestión del proyecto.

OTRAS HERRAMIENTAS POPULARES SON:

Gitlab: es una plataforma integral de gestión de repositorios de código basada en Git. A continuación presento algunas de sus características clave:

- a. Repositorios de Código:
Permite la creación y gestión de repositorios Git para tu código fuente.
- b. Control de Versiones:
Ofrece un sistema de control de versiones distribuido que facilita el eguimiento de cambios en el código.
- c. Gestión de Ramas:
Permite la creación y fusión de ramas, facilitando el desarrollo paralelo y la implementación de nuevas funciones.
- d. Colaboración en Proyectos:
Proporciona herramientas para gestionar proyectos de desarrollo de software de manera colaborativa, incluyendo la asignación de tareas y el seguimiento del progreso.
- e. Integración Continua (CI) y Despliegue Continuo (CD):
Ofrece una potente funcionalidad de CI/CD que automatiza las pruebas y despliegues, mejorando la calidad del código y la eficiencia del desarrollo.
- f. Seguimiento de Problemas y Gestión de Proyectos:
Incluye un sistema de seguimiento de problemas y funciones avanzadas de gestión de proyectos para facilitar la planificación y el monitoreo del progreso.
- g. Wiki y Documentación:
Permite crear wikis y documentación para proyectos directamente dentro de la plataforma.
- h. Registros y Análisis:
Proporciona registros detallados y herramientas de análisis para evaluar el rendimiento y la eficacia del desarrollo.
- i. Gestión de Configuración:
Facilita la gestión de la configuración del proyecto y la definición de reglas de fusión para mantener la integridad del código.
- j. Seguridad:
Incluye características de seguridad avanzadas, como escaneo de vulnerabilidades en el código y la capacidad de establecer políticas de seguridad.

- k. Escalabilidad y Flexibilidad:
Es escalable y se adapta bien a diferentes tamaños de equipos y proyectos.
- l. Comunidad y Soporte Activo:
Cuenta con una comunidad activa y un sólido soporte técnico.

GitLab es una plataforma completa que abarca desde la gestión de código fuente hasta la entrega continua, proporcionando a los equipos todas las herramientas necesarias para el desarrollo colaborativo de software. Puede ser utilizado como servicio en la nube o instalado en un servidor propio, ofreciendo así flexibilidad en su implementación.

Bitbucket: es una plataforma de alojamiento de repositorios basada en Git que proporciona herramientas para el desarrollo de software colaborativo. Algunas de las características clave son:

- ✓ Alojamiento de Repositorios Git y Mercurial:
Bitbucket admite tanto repositorios Git como Mercurial, brindando flexibilidad a los desarrolladores según sus preferencias.
- ✓ Control de Acceso y Privacidad:
Ofrece opciones robustas de control de acceso y privacidad, permitiendo a los equipos gestionar quién puede ver o contribuir a sus proyectos.
- ✓ Herramientas de Colaboración:
Facilita la colaboración en equipo con herramientas como solicitudes de extracción (pull requests), que permiten revisar y fusionar cambios de manera eficiente.
- ✓ Integración con Jira:
Bitbucket se integra estrechamente con Jira, otra herramienta popular de Atlassian, permitiendo una gestión de proyectos más completa, desde el seguimiento de problemas hasta el código fuente.
- ✓ Pipelines para Integración Continua:
Ofrece Bitbucket Pipelines, una característica de integración continua (CI) que permite automatizar la construcción, las pruebas y los despliegues directamente desde el repositorio.
- ✓ Herramientas de Revisión de Código:
Proporciona herramientas para la revisión de código, lo que facilita que los equipos mantengan altos estándares de calidad y colaboración.

- ✓ Wikis y Páginas de Proyectos:
Incluye funciones de wiki para documentación interna y la capacidad de crear páginas de proyectos que pueden contener información relevante para el equipo.
- ✓ Soporte de Integración con Herramientas de Desarrollo:
Bitbucket es compatible con diversas herramientas de desarrollo, como integración con IDE, servicios de CI/CD, y más.
- ✓ Bitbucket Data Center:
Para empresas que necesitan escalabilidad y alta disponibilidad, Bitbucket ofrece una versión empresarial llamada Bitbucket Data Center.
- ✓ Bitbucket Server:
Si prefieres alojar Bitbucket en tu propio servidor, Atlassian también ofrece Bitbucket Server.

Bitbucket es especialmente popular entre los equipos que ya utilizan otras herramientas de Atlassian, como Jira y Confluence, ya que se integra perfectamente con estas para proporcionar una solución integral de desarrollo de software. La elección entre GitHub y Bitbucket a menudo depende de las preferencias del equipo y del ecosistema de herramientas que mejor se ajuste a sus necesidades.

OTRAS HERRAMIENTAS DE VERSIONAMIENTO EXISTENTES

1. Mercurial: es otro sistema de control de versiones distribuido similar a Git. Aunque ha perdido popularidad frente a Git, sigue siendo utilizado en algunos proyectos.
2. Subversion (SVN): es un sistema de control de versiones centralizado que sigue siendo utilizado en algunos entornos. A diferencia de Git y Mercurial, SVN no sigue el modelo distribuido.
3. Perforce (Helix Core): también conocido como Helix Core, es un sistema de control de versiones centralizado que se utiliza comúnmente en empresas grandes y en proyectos que manejan grandes cantidades de archivos binarios.
4. AWS CodeCommit: es un servicio de control de versiones alojado en la nube proporcionado por Amazon Web Services (AWS). Está completamente integrado con otros servicios de AWS.
5. Microsoft Azure Repos: es parte de Azure DevOps y ofrece servicios de control de versiones Git y Team Foundation Version Control (TFVC).
6. Bazaar: es un sistema de control de versiones distribuido que se destaca por su enfoque amigable y fácil aprendizaje.

7. Fossil: es un sistema de control de versiones distribuido y un sistema de seguimiento de problemas en uno solo. Es simple de configurar y usar.
8. Darcs: es otro sistema de control de versiones distribuido que se centra en la facilidad de uso y la teoría de parches.

La elección de la herramienta de control de versiones depende de varios factores, como las necesidades específicas del proyecto, la preferencia del equipo, y la integración con otras herramientas y servicios utilizados en el desarrollo. Cada una de estas herramientas tiene sus propias fortalezas y debilidades, así que es importante evaluarlas en función de los requisitos del proyecto