



PRD 5: Orquestación de contenedores (microservicios) con kubernetes.

Proyecto: Eco-Adventure (Aplicación para la planificación de viajes sostenibles).

Revisión

ÍNDICE

1. PRD 5.....	3
2. Información general.....	3
3. Descripción.....	3
4. Metas.....	4
5. Alcance.....	4
6. Riesgos de Negocio-Ejecución.....	4
7. Problemas a Resolver (PR).....	5
8. Requerimientos y Criterios de Aceptación.....	5
9. Marco Regulatorio.....	6
10. Frecuencia y Volumen de Uso Esperado.....	6
11. Interesados.....	7
12. Matriz de Interesados y Procesos Impactados.....	7
13. Flujo de actividades.....	9
13.1 Diagrama de Gantt.....	9
14. Diagrama de proceso de negocios.....	10
15. Flujos de tareas.....	12
16. Datos e información.....	13
17. Analítica.....	13
18. Comunicación.....	13
19. Glosario.....	14
20. Anexos.....	15
21. Aprobación.....	16

1. PRD 5

Orquestación de contenedores (microservicios) con kubernetes.

2. Información general

Este PRD consiste en la implementación de una infraestructura de microservicios utilizando Kubernetes para la gestión de EcoAdventure. El objetivo es utilizar contenedores para segmentar las funcionalidades de la aplicación, lo cual proporcionará mayor flexibilidad, escalabilidad y fiabilidad. Al descomponer la aplicación en diferentes servicios, podremos desarrollar, probar e implementar cada función de manera individual, lo que agiliza el proceso de desarrollo y permite una mayor rapidez en la entrega de nuevas funciones o correcciones.

- ❖ Frontend
 - Flutter
 - Dart
- ❖ Backend
 - Node.js
 - Express
 - JavaScript
 - Kubernetes
 - minikube



3. Descripción

3.1 Frontend

- Flutter
Versión: 3.10.2
- Dart
Versión: 3.0.2

3.2 Backend

- Node.js
Versión: 18.16.0
- Express/Node.js
Versión : 4.18.2/18.16.0
- Kubernetes
Versión : 1.27
- minikube
versión 1.31

3.3 Patrón de diseño: Patrón de N capas

Descripción:

4. Metas

- Implementar una arquitectura de microservicios para la aplicación móvil usando Kubernetes.
- Mejorar la escalabilidad y la resiliencia de la aplicación.
- Facilitar la entrega continua y la integración continua (CI/CD) mediante la segmentación de la aplicación en contenedores individuales.
- Mejorar el tiempo de entrega de nuevas funciones y correcciones.
- Proporcionar una mayor flexibilidad para probar nuevas funciones y cambios sin afectar a los demás servicios de la aplicación.

5. Alcance

El alcance del proyecto incluirá el diseño, la configuración y la implementación de la arquitectura de microservicios utilizando Kubernetes. También incluirá la migración de la aplicación EcoAdventure actual a esta nueva infraestructura, la formación del equipo de desarrollo sobre cómo trabajar con esta nueva arquitectura, y la implementación de protocolos de monitorización y mantenimiento para asegurar el buen funcionamiento de los servicios en la plataforma Kubernetes.

6. Riesgos de Negocio-Ejecución

Inadecuada formación del equipo: Si el equipo de desarrollo no está familiarizado con Kubernetes o los microservicios, podría haber un retraso en la implementación del proyecto.

Dificultades técnicas: El despliegue de microservicios en Kubernetes puede ser un proceso complejo que puede encontrarse con varios problemas técnicos inesperados.

Costos de infraestructura: La implementación de Kubernetes requiere un nivel de recursos de hardware que podría superar las expectativas iniciales.

Dificultades de migración: La migración de una aplicación monolítica a una basada en microservicios puede resultar compleja y puede haber dificultades en la transferencia de datos y funcionalidades a los nuevos servicios.

Tiempo de inactividad durante la migración: Aunque la implementación de microservicios tiene como objetivo mejorar la resiliencia y la disponibilidad, podría haber un tiempo de inactividad no planificado durante la fase de migración.

7. Problemas a Resolver (PR)

- **Monolito y Escalabilidad:** La aplicación actual de Eco Adventure es un monolito, lo que dificulta la escalabilidad independiente de cada funcionalidad. Se necesita una arquitectura más modular que permita escalar y gestionar cada servicio de manera independiente.
- **Entrega Continua y Rápida:** La implementación actual no permite una entrega rápida y continua de nuevas funciones o correcciones, lo que retrasa la mejora de la aplicación. Se busca una solución que facilite la implementación y prueba rápida de nuevos servicios o actualizaciones sin afectar al resto de la aplicación.
- **Resiliencia y Fiabilidad:** La actual infraestructura puede ser susceptible a fallos, lo que afecta la disponibilidad y la experiencia del usuario. Se necesita una solución que garantice la resiliencia y fiabilidad de la aplicación, asegurando una alta disponibilidad.
- **Complejidad Técnica:** La adopción de Kubernetes y microservicios puede presentar desafíos técnicos y complejidad en la configuración y gestión de la infraestructura. Se busca una implementación que reduzca la complejidad técnica y facilite el despliegue de servicios.
- **Formación del Equipo:** El equipo de desarrollo actual puede no estar familiarizado con Kubernetes y microservicios. Se requiere una formación adecuada para garantizar que el equipo pueda trabajar eficientemente con la nueva arquitectura.

8. Requerimientos y Criterios de Aceptación

8.1 Requerimiento: Arquitectura de Microservicios.

- **Criterio de Aceptación:** El sistema debe estar diseñado e implementado como una arquitectura de microservicios utilizando Kubernetes para la gestión y despliegue de los servicios.

8.2 Requerimiento: Escalabilidad independiente.

- **Criterio de Aceptación:** Cada servicio debe poder escalar de manera independiente según sus necesidades de rendimiento y carga.

8.3 Requerimiento: Entrega Continua y Despliegue Rápido.

- **Criterio de Aceptación:** Se debe establecer un proceso de CI/CD que permita la entrega continua de nuevas funciones y correcciones con tiempos de implementación reducidos.

8.4 Requerimiento: Alta Disponibilidad.

- **Criterio de Aceptación:** La arquitectura debe garantizar la alta disponibilidad de la aplicación, minimizando el tiempo de inactividad no planificado.

8.5 Requerimiento: Monitorización y Mantenimiento.

- **Criterio de Aceptación:** Se deben implementar protocolos de monitorización y mantenimiento para asegurar el buen funcionamiento de los servicios en la plataforma Kubernetes.

8.6 Requerimiento: Formación del Equipo.

- **Criterio de Aceptación:** Se debe proporcionar formación adecuada al equipo de desarrollo sobre cómo trabajar con Kubernetes y desarrollar microservicios.

9. Marco Regulatorio

- El proyecto debe cumplir con las regulaciones y leyes de protección de datos y privacidad aplicables en la región donde opera EcoAdventure. Se debe garantizar que los datos del usuario están protegidos y se cumplan todas las normativas relacionadas con el uso de información personal.
- Seguridad: La aplicación debe implementar medidas de seguridad robustas para proteger la información del usuario y prevenir accesos no autorizados. Se deben emplear protocolos de cifrado y métodos de autenticación seguros para proteger las cuentas de los usuarios y cualquier información sensible almacenada en el sistema.

10. Frecuencia y Volumen de Uso Esperado

- El sistema debe ser capaz de manejar un alto volumen de usuarios y transacciones, especialmente durante picos de demanda en momentos de alta actividad turística o promociones especiales. Se espera un uso frecuente y constante de la aplicación móvil y sus servicios, por lo que la infraestructura debe ser escalable para manejar la carga.

11. Interesados

Interesados	
Equipo de Desarrollo	Encargados de diseñar, desarrollar e implementar la infraestructura de microservicios con Kubernetes y mantenerla en funcionamiento.
Gerentes y Stakeholders	Interesados en la mejora de la aplicación y la eficiencia del desarrollo, así como en los beneficios empresariales que se obtendrán con la nueva arquitectura.
Usuarios Finales	Beneficiarios directos de una aplicación más rápida, confiable y con nuevas funciones implementadas con mayor agilidad.
Equipo de Operaciones	Responsables de la monitorización y el mantenimiento del sistema para garantizar su buen funcionamiento y disponibilidad.
Entidades Reguladoras	Asegurarse de que el proyecto cumpla con las regulaciones y leyes aplicables, especialmente en lo que respecta a la protección de datos y privacidad del usuario.

12. Matriz de Interesados y Procesos Impactados

Interesado	Procesos Impactados
Usuarios de la aplicación	Búsqueda de destinos, selección de actividades y planificación de viajes.
Desarrolladores de la aplicación	Diseño, desarrollo, pruebas y mantenimiento de la página de inicio.
	Diseño: Andres Jacome, Alisson Clavijo.
	Desarrollo: Andres Jacome, Adrian Iza.
	Pruebas: Gabriel Aguirre.
	Mantenimiento : Deniss Yaguarshungo.
Proveedores de servicios turísticos	Promoción de sus servicios y actividades en la página de inicio.

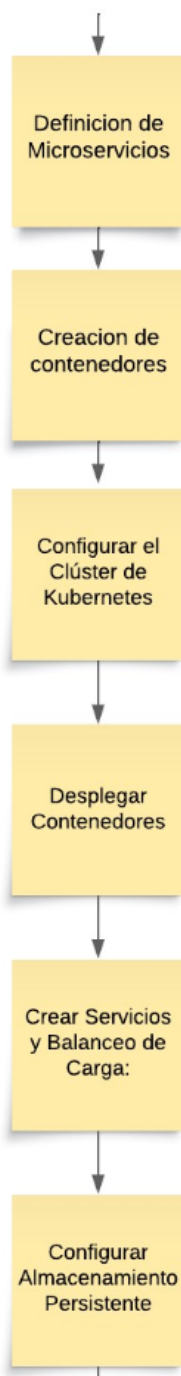
13. Flujo de actividades

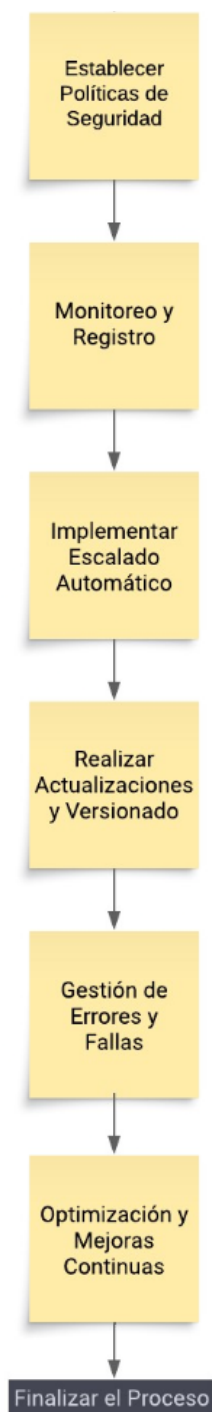
13.1 Diagrama de Gantt

Prd5 Semana 6					
<input type="checkbox"/>	Tarea		Persona	Estado	Fecha
<input type="checkbox"/>	Informacion General Descripcion	+		Listo	jul. 11
<input type="checkbox"/>	Metas, Alcance	+	GA	Listo	jul. 11
<input type="checkbox"/>	Riesgos y Problemas a aResolver	+		Listo	jul. 12
<input type="checkbox"/>	Requerimientos, criterios de acept...	+		Listo	jul. 12
<input type="checkbox"/>	Marco Regulatorio	+		Listo	jul. 13
<input type="checkbox"/>	Frecuencia y volumen de uso esper...	+		Listo	jul. 13
<input type="checkbox"/>	Interesados, matrix de interesados,...	+	GA	Listo	jul. 13
<input type="checkbox"/>	flujo de actividades, diagramas de ...	+		Listo	jul. 14
<input type="checkbox"/>	Interesados, matrix de interesados,...	+	GA	Listo	jul. 13
<input type="checkbox"/>	flujo de actividades, diagramas de ...	+		Listo	jul. 14
<input type="checkbox"/>	flujo de tareas y datos e informacion	+	GA	Listo	jul. 14
<input type="checkbox"/>	Analitica, comunicacion	+		Listo	jul. 15
<input type="checkbox"/>	Glosario, anexos, aprobacion	+		Listo	jul. 15
<input type="checkbox"/>	+ Agregar Tarea				

14. Diagrama de proceso de negocios

Inicio del proceso





15. Flujos de tareas



16. Datos e información

Datos	Información
Base de datos	Destinos sostenibles y actividades relacionadas.
Regulaciones y leyes pertinentes	El turismo sostenible y la conservación del medio ambiente.
Datos de uso y análisis	Sobre la interacción de los usuarios con la página de inicio y los destinos seleccionados.
Información sobre los proveedores.	Los proveedores de servicios turísticos sostenibles y sus ofertas.

17. Analítica

La implementación de esta sólida infraestructura de microservicios con Kubernetes proporcionará una base ideal para incorporar capacidades analíticas avanzadas en EcoAdventure. Aprovechando la arquitectura modular y escalable, será factible integrar herramientas de análisis que permitan recopilar datos valiosos sobre el comportamiento de los usuarios y el rendimiento general de la aplicación.

Dentro de la arquitectura de microservicios, la interfaz del perfil de usuario puede ser el punto focal para la implementación de funciones de análisis. Mediante el uso de librerías de terceros o el desarrollo de módulos personalizados dentro de la capa de backend, se podrán habilitar capacidades de análisis que monitoreen sistemáticamente cómo los usuarios interactúan con la aplicación.

Las funcionalidades analíticas podrían recopilar datos sobre preferencias de viaje, actividades más buscadas, destinos frecuentemente seleccionados y el uso específico de características de la interfaz. Estos datos recopilados de manera efectiva serán almacenados de forma adecuada y utilizados para tomar decisiones informadas con el objetivo de mejorar el rendimiento general de la aplicación y enriquecer la experiencia del usuario.

Gracias a un análisis detallado, se podrán identificar patrones y tendencias de comportamiento, lo que permitirá optimizar la oferta de destinos y actividades personalizadas y pertinentes para cada usuario. La información

obtenida a partir de estas herramientas de análisis permitirá al equipo de desarrollo tomar decisiones basadas en datos para mejorar continuamente la aplicación y responder a las necesidades y preferencias cambiantes de los usuarios.

18. Comunicación

Comunicación	Objetivo
Con los usuarios a través de la aplicación	<p>Agilizar la entrega de información pertinente y actualizada a los usuarios sobre destinos sostenibles.</p> <p>Además, se garantizará una continua provisión de actualizaciones sobre opciones de viaje y actividades relacionadas, aprovechando la modularidad de los microservicios para implementar cambios de manera rápida y efectiva..</p>
Con los desarrolladores de la aplicación	<p>Impulsar un desarrollo constante y mejoras continuas en la página de inicio, con el objetivo de optimizar la experiencia del usuario.</p> <p>Adicionalmente, se habilitará la implementación ágil de mejoras y actualizaciones para enriquecer tanto la funcionalidad como la usabilidad de la aplicación. La arquitectura de microservicios permitirá trabajar de manera independiente en distintos componentes, lo que facilitará el proceso de desarrollo y asegurará una entrega rápida de las mejoras.</p>

19. Glosario

- **Infraestructura de Microservicios:** Es una arquitectura de software que descompone una aplicación en pequeños servicios independientes y autónomos, cada uno ejecutándose en su propio contenedor, lo que permite una mayor flexibilidad y escalabilidad.
- **Microservicio:** Es un componente individual de una aplicación que se enfoca en una función específica. Los microservicios se pueden desarrollar, probar, desplegar y escalar de manera independiente.
- **Kubernetes:** Es una plataforma de orquestación de contenedores de código abierto que facilita la gestión, el despliegue y la escalabilidad de aplicaciones basadas en microservicios.
- **Contenedor:** Es una unidad de software que encapsula una aplicación y todas sus dependencias para garantizar la portabilidad y la consistencia entre diferentes entornos de ejecución.
- **Orquestación:** Es el proceso de gestionar y coordinar automáticamente la implementación, el escalado y la gestión de contenedores en un clúster de servidores.
- **Clúster de Kubernetes:** Es un conjunto de nodos (servidores) que ejecutan el software de Kubernetes y trabajan juntos para administrar y programar contenedores.
- **Despliegue:** Es el proceso de implementar y ejecutar un contenedor o microservicio en un clúster de Kubernetes para que esté disponible para su uso.
- **Escalabilidad:** Es la capacidad de una infraestructura para adaptarse al aumento o disminución de la demanda, ajustando automáticamente la cantidad de recursos asignados a los contenedores o microservicios.
- **Autoescalado:** Es una característica de Kubernetes que permite que los pods (instancias de un microservicio) se escalen automáticamente según la carga de trabajo, para garantizar el rendimiento óptimo.
- **Alta disponibilidad:** Es la capacidad de una infraestructura para mantener un servicio en funcionamiento incluso si se producen fallos en algunos componentes, asegurando la continuidad del servicio.

-
- **Autorreparación:** Es una característica de Kubernetes que monitorea constantemente el estado de los pods y los reinicia automáticamente en caso de que fallen, para mantener la estabilidad del sistema.
 - **Servicio de Kubernetes:** Es un recurso de Kubernetes que define una abstracción lógica para exponer aplicaciones ejecutándose en un conjunto de pods como un servicio de red.
 - **Carga de Trabajo:** Es la cantidad de trabajo que una aplicación o sistema necesita realizar, lo que determina la cantidad de recursos que se asignan a los microservicios para garantizar un rendimiento óptimo.
 - **Despliegue Rolling:** Es una estrategia de implementación en la que las nuevas versiones de los microservicios se implementan gradualmente, reemplazando progresivamente las versiones anteriores, para minimizar el tiempo de inactividad.
 - **Namespace:** Es una forma de organizar y aislar recursos en un clúster de Kubernetes, lo que permite la compartimentación de aplicaciones y microservicios para evitar conflictos y mejorar la seguridad.
 - **Flutter:** Un framework de código abierto desarrollado por Google para crear aplicaciones móviles multiplataforma.
 - **Dart:** El lenguaje de programación utilizado en Flutter.
 - **Node.js:** Entorno de ejecución de JavaScript que permite ejecutar código JavaScript en el lado del servidor.
 - **Express:** Una biblioteca JavaScript utilizada en el lado del servidor junto con Node.js para crear aplicaciones web y API.
 - **JavaScript:** Un lenguaje de programación muy utilizado para desarrollar aplicaciones web interactivas.
 - **Patrón de diseño N-Tier:** Un patrón de diseño utilizado para separar las responsabilidades de una aplicación en capas distintas.
 - **Frontend:** Se refiere a la capa de la aplicación responsable de la interfaz de usuario y la presentación visual.
 - **Backend:** Se refiere a la capa de la aplicación que maneja la lógica de negocio y el acceso a los datos.
 - **Modularidad:** Capacidad de dividir una aplicación en módulos independientes con interfaces definidas.

- **Escalabilidad:** La capacidad de una aplicación para gestionar un aumento de la carga de trabajo y del número de usuarios.
- **Mantenibilidad:** La facilidad con la que una aplicación puede mantenerse y evolucionar con el tiempo.
- **Marco normativo :**Un marco normativo es el conjunto de leyes, normas, decretos, reglamentos, etc., de carácter obligatorio o indicativo que rigen en un país, estado o institución; siendo en este caso con perspectiva de género.
- **El patrón de diseño Decorador:** es un patrón estructural que permite añadir funcionalidades adicionales o responsabilidades a objetos individuales de forma dinámica, sin modificar su estructura básica. Se basa en la composición, donde se agregan objetos dentro de otros objetos para extender sus capacidades

20. Anexos

- La Ley Orgánica de Protección de Datos Personales (LOPD) en Ecuador tiene por objeto regular el tratamiento de los datos personales de los ciudadanos y garantizar su derecho a la privacidad y protección de la información. En el marco del desarrollo y operación de la aplicación ECOADVENTURE, es fundamental cumplir con los principios y obligaciones establecidos en esta ley para asegurar la seguridad y confidencialidad de la información de los usuarios.

LEY ORGÁNICA DE PROTECCIÓN DE DATOS PERSONALES(n.d.). Edu.ec.

Retrieved July 20, 2023 from:

https://www.finanzaspopulares.gob.ec/wp-content/uploads/2021/07/ley_organica_de_proteccion_de_datos_personales.pdf

- Análisis de los preceptos constitucionales y legales del turismo sostenible en el Ecuador.

Vista de El Turismo Sostenible en Ecuador: reflexiones constitucionales y legislativas. (n.d.). Edu.ec. Retrieved July 11, 2023, from

<https://revistas.uteq.edu.ec/index.php/csye/article/view/553/670>

- Se deben tener en cuenta las disposiciones establecidas en la Ley de Telecomunicaciones de Ecuador. Esta ley, que regula las comunicaciones electrónicas en el país, puede contener aspectos relevantes para la protección de la privacidad de los usuarios y la seguridad de la información en el contexto de la aplicación.

LEY ORGÁNICA DE TELECOMUNICACIONES.(n.d.). Retrieved July 20, 2023 from:

<https://www.telecomunicaciones.gob.ec/wp-content/uploads/downloads/2016/05/Ley-Org%C3%A1nica-de-Telecomunicaciones.pdf>

21. Aprobación

Nombre	firma	Fecha
Elaborado: AGUIRRE GUANOLUISA PATRICIO GABRIEL Alumno		12/07/2023
Elaborado: ALISSON NICOLE CLAVIJO GUTIÉRREZ Alumno		12/07/2023
Elaborado: BRYAN ADRIÁN IZA OÑA Alumno		12/07/2023
Elaborado: ANDRÉS JONATHAN JÁCOME NAVARRETE Alumno		12/07/2023
Elaborado: BRYAN DENISS YAGUARSHUNGO AVENDAÑO Alumno		12/07/2023

