

**UNIVERSIDAD TÉCNICA DE MACHALA**

Maestría en Software

**Asignatura:**  
Diseño arquitectónico en ambientes cloud  
  
  
**Tema:**

**PROYECTO FINAL**

Docente: Ing. Armando Cabrera

Estudiantes:

Ing. Fernando Castillo

Ing. Carlos Quezada

Ing. Esteban Gonzabay

Ing. Jorge Miranda

Ing. Leonardo Caraguay

2021-2022

Contenido

[Introducción 3](#_Toc82109227)

[Desarrollo de la propuesta 4](#_Toc82109228)

[Metodología Tradicional – Scrum 4](#_Toc82109229)

[1. Identificación de roles del proyecto. 4](#_Toc82109230)

[2. Identificación de módulos del sistema. 4](#_Toc82109231)

[3. Identificación de requerimientos funcionales. 4](#_Toc82109232)

[4. Identificación de Requerimientos no funcionales. 4](#_Toc82109233)

[5. Diagramas de casos de Usos 5](#_Toc82109234)

[DEVOPS 1](#_Toc82109235)

[1. Descomposición funcional (Cadena de valor, flujos de valor y capacidades) 1](#_Toc82109236)

[2. Arquitectura orientada a microservicios. 2](#_Toc82109237)

[Justificación de la selección de los ambientes cloud. 1](#_Toc82109238)

[Bibliografía 1](#_Toc82109239)

# Introducción

La gran mayoría de las empresas de software funcionan manteniendo a los departamentos aislados unos de otros es decir por un lado están los **desarrolladores**, que deben, valga la redundancia, desarrollar un producto a la mayor velocidad posible, sin dejar de lado la innovación y guiados por una gran tendencia al cambio. Por otro lado, los **profesionales de operaciones** que deben implementar, configurar y optimizar recursos para el correcto funcionamiento del código, buscando constantemente la estabilidad del sistema.

Conforme los desarrolladores van terminando los entregables, son enviados a los de operaciones para que los implementen en un entorno de producción. Durante este proceso, es bastante común que aparezcan errores, lo que termina en discusiones entre los miembros de los equipos del proyecto.

Mientras operaciones soluciona la incidencia, desde desarrollo se sigue enviando nuevo código a operaciones, que trata de ralentizar el proceso para estabilizar y optimizar al máximo el sistema. Este desequilibrio termina generando conflictos y retrasando la entrega del proyecto, lo que supone un coste económico para el negocio.

Dada a esta problemática fue que nació DEVOPS, DevOps surge para fomentar la comunicación entre departamentos, buscando intereses comunes de todas las partes, automatizando procesos y facilitando una integración continua.

# Desarrollo de la propuesta

## Metodología Tradicional – Scrum

### Identificación de roles del proyecto.

|  |  |
| --- | --- |
| **Nombres y apellidos** | **Roles** |
| Leonardo Caraguay | Product Manager |
| Carlos Quezada | Analista |
| Esteban Gonzabay | Analista |
| Jorge Miranda | Diseñador |
| Fernando Castillo | Arquitecto de software |

### Identificación de módulos del sistema.

* Módulo de gestión de usuarios.
* Modulo de gestión de levantamiento de datos.
* Modulo de gestión de mapas.
* Modulo de gestión de catálogos.

### Identificación de requerimientos funcionales.

* Gestión de usuarios.
* Gestión de reportes.
* Gestión de formularios.
* Gestión de fotos geolocalizadas.
* Gestión de catálogos.
* Registro de coordenadas del GPS del dispositivo móvil.
* Captura de fotos desde dispositivo móvil.
* Registro de mapas online/offline.
* Registro de historial de información.
* Gestión de waypoints.
* Visualización simultánea de pantallas para comparación de información de mapas.
* Funcionalidad para lectura de polígonos, líneas, puntos en formato .shp y .gpx.
* Asignar y editar atributos alfanuméricos a vectores.
* Cálculo de áreas y distancias por recorrido.
* Bajar / subir datos manualmente (sincronización) al SAF o SEMOP, según corresponda.

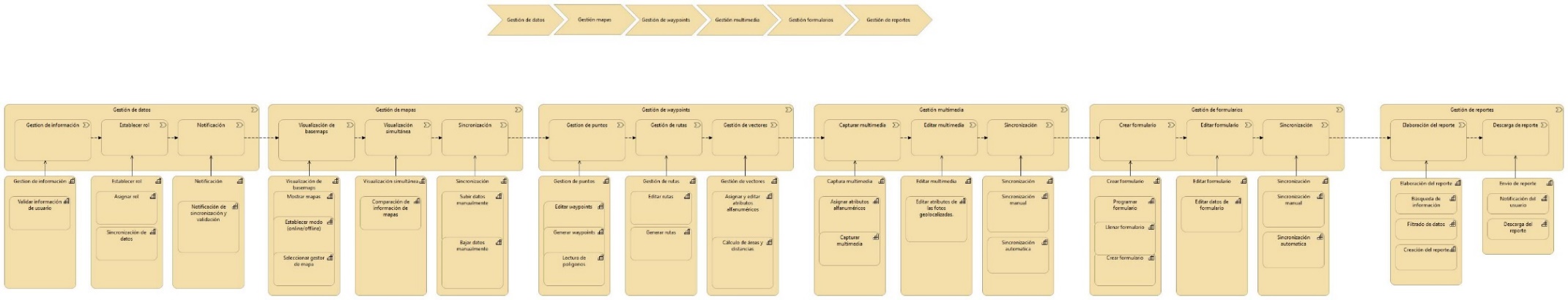
### Identificación de Requerimientos no funcionales.

* Bases de datos relacional PosgreSQL,
* Geodatabase PostGis,
* Servidor de servicios geomáticos Geoserver,
* Lenguajes de desarrollo Java y PHP.
* Dar la función de navegación (go to).
* Toma de fotografías con la cámara integrada al dispositivo móvil.
* Disponer offline la visualización de información de servidores web: Google Maps, Bing basemap, Planet Scope, Streetview.
* El sistema debe operar a través de Internet y contar con un aplicativo para los dispositivos móviles con sistema operativo Android, la captura de información debe realizarse Online y Offline.
* El sistema debe soportar el acceso concurrente de usuarios conforme a su rol previamente asignado.
* El aplicativo móvil debe realizar la captura de información de forma ágil y remitir los datos al SAF o SEMOP (sincronización), según corresponda, cuando se disponga de conexión a internet.
* El aplicativo móvil puede ser utilizado sin conexión a internet Offline, para luego sincronizarse con el SAF o SEMOP.
* El sistema debe contar con las correspondientes validaciones de datos, necesarias para la carga de información desde el aplicativo móvil de manera que se asegure la calidad de los mismos.
* El sistema debe consolidar los datos capturados en una base de datos centralizada.
* El sistema debe generar mensajes de confirmación de la carga de información. En caso de fallo en la carga, el sistema deberá permitir cargar nuevamente la información asegurando su integridad.
* Diseño del sistema de acceso concurrente de usuarios en función a roles.
* La autenticación para el acceso al sistema será vía usuario/clave.
* El menú del usuario debe ser personalizado de acuerdo al rol asignado.
* Los aplicativos móviles se deberán implementar utilizando la plataforma Open Data Kit

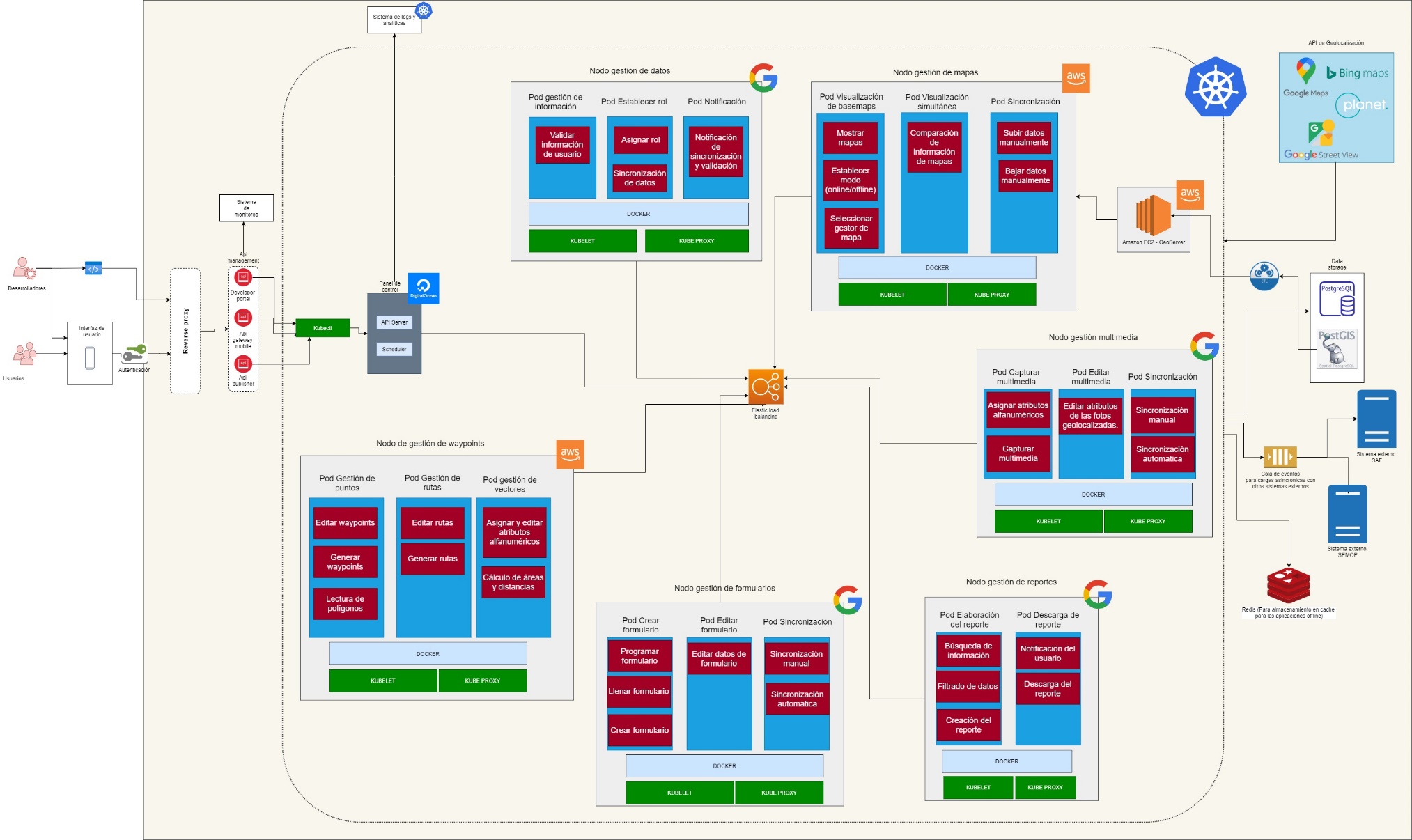
### Diagramas de casos de Usos

## DEVOPS

### Descomposición funcional (Cadena de valor, flujos de valor y capacidades)



### Arquitectura orientada a microservicios.

****

### Justificación de la selección de los ambientes cloud.

Para la implementación del panel de control, debido a que este panel de control es básicamente el desarrollo de un sistema web, se optó por utilizar **Digital Ocean** debido que es una plataforma enfocada en desarrolladores de software y esto sumado a su bajo coste lo hace ideal para esta tarea.

Para la gestión de mapas, waypoints e instalación del entorno de postgis, geoserver y otras herramientas afines se optó por utilizar instancias de **Amazon EC2** (Elastic Compute Cloud) debido a que para geoserver ya se encuentra disponible una instancia EC2 completamente implementada en el marketplace de AWS denominada *OpenGeo Suite on Amazon Web Services* (<https://aws.amazon.com/marketplace/pp/prodview-fglqvc7pwsm4e>) esto sumado a que tiene un coste de algunos pocos centavos de dólar por hora o incluso gratis cuando se trata de realizar pruebas. En conclusión, se obtendrá una arquitectura muy robusta con los mejores herramientas open source (PostGIS + GeoServer + GeoExplorer +GeoWebCache + OpenLayers), el servidor y el soporte de AWS en un único paquete que funcionarán armónicamente con las otros dos nodos (mapas y waypoints) implementadas también bajo una arquitectura AWS.

Para los nodos de gestión de datos, gestión de formularios y gestión multimedia, se optó por utilizar **Google Cloud Platform (GCP)** debido a que estos nodos no requerirán de mucho consumo de hardware o software por tal motivo no es necesario gastar mucho dinero en estos servidores y en el aspecto de precios, GCP es el más económico en comparación a las otras plataformas cloud mencionadas anteriormente y eso sin contar de la excelente disponibilidad y velocidad ofrecida.

# Bibliografía