**Conocer la arquitectura de la plataforma Dory**

**Objetivo:** Conocer como está compuesta la plataforma Dory y mostrar en que servicios están desplegados sus componentes.

Usaremos tres diagramas para mostrar en diferentes niveles la plataforma. Y de manera similar a como usamos Google maps iremos haciendo zoom en cada uno de ellos para ir conociendo más detalles.

En primer lugar, usaremos el diagrama de nivel más alto llamado “Diagrama de contexto” como se muestra en la *Imagen 1 Diagrama de contexto de la plataforma Dory.* Este nos mostrará cómo encaja la plataforma Dory con su entorno y los elementos con los que interactúa(personas, clientes software y sistemas externos.

Luego haremos zoom en el rectángulo azul claro de la Imagen 1 Diagrama de contexto de la plataforma Dory para poder ver cómo está compuesto la plataforma Dory(aplicaciones, almacenamiento de datos, etc. El resultado del hacer este zoom podremos verlo en la Imagen 2 Diagrama de contenedores. Cuando hablamos de contenedores estamos haciendo referencia a todo por lo que esta compuesto la plataforma Dory, esto es una aplicación web, una aplicación de escritorio, una aplicación móvil, una base de datos, un Api-rest, etc. También este diagrama nos muestra las relaciones de estos contenedores con su entorno.

Para estos diagramas hemos usado rectángulos para diferenciar los elementos de la plataforma Dory y los elementos del entorno. Cada imagen tiene al pie de esta un recuadro donde que contiene una muestra de cada rectángulo usado en el diagrama con su respectivo color y una etiqueta de su rol en el diagrama.

**Recursos necesarios:**

* Diagrama de contexto de la plataforma
* Diagrama de contenedores
* Diagrama de despliegue

**Pasos:**

**1. Conocer el contexto de la plataforma Dory (Nivel 1(**

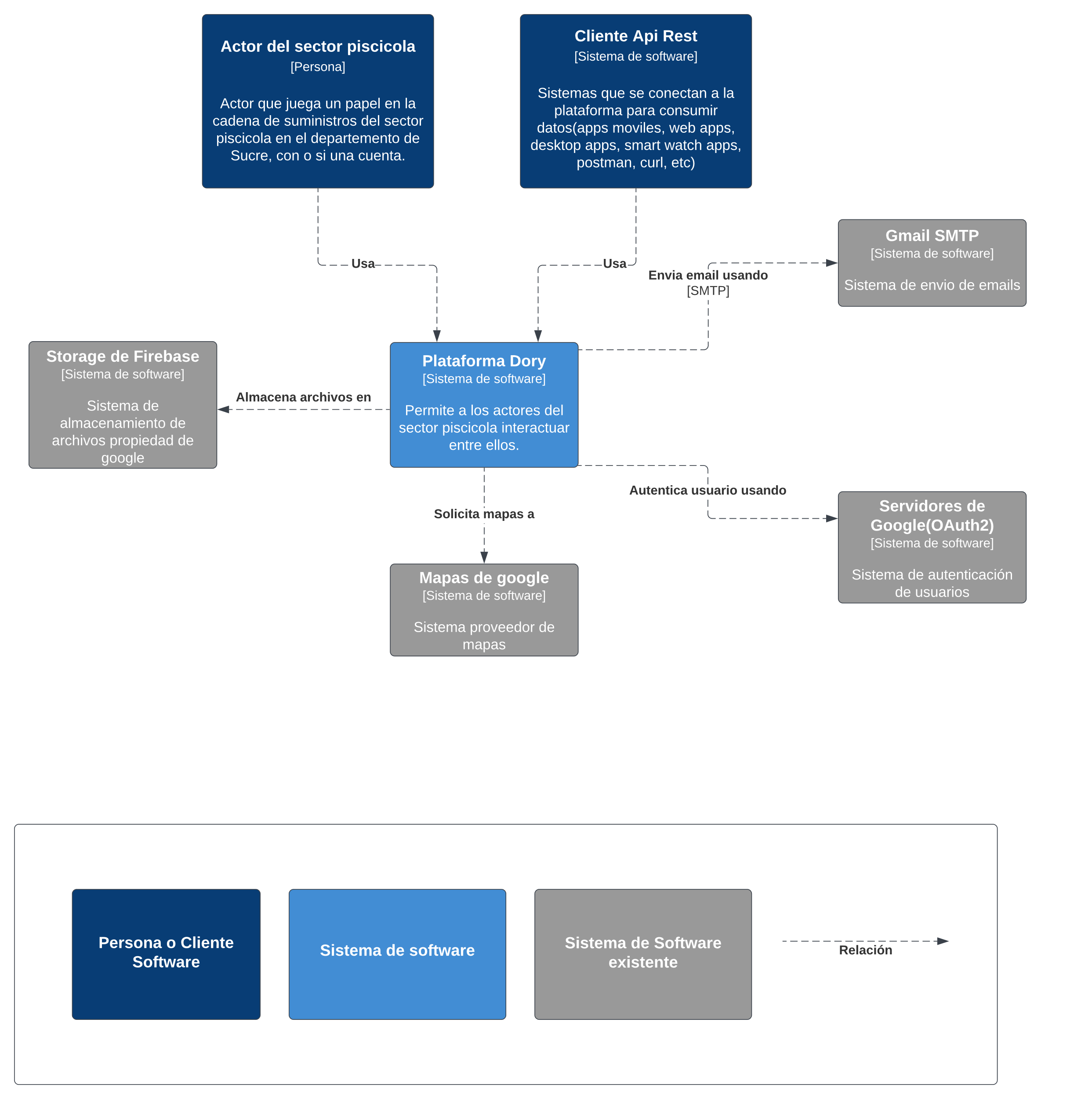


Imagen 1 Diagrama de contexto de la plataforma Dory

La Imagen 1 Diagrama de contexto de la plataforma Dory nos muestra los sistemas externos (rectángulos grises) con los que se relaciona la plataforma Dory (rectángulo azul claro). Podemos ver que esta hace uso del Storage de Firebase para almacenar las imágenes o archivos que los usuarios almacenan desde su panel de administración como son sus fotos de perfil, fotos de sus granjas, fotos de sus productos etc. También hace uso de Mapas de google(Google maps) para mostrar la ubicación de las granjas en el servicio de Geolocalización. Además, usa el servicio de autenticación de google para que los usuarios puedan ingresar con su cuenta (Google) en la plataforma. Y por último usa el sistema de SMTP de Gmail para enviar un correo de bienvenida y verificación de la cuenta de correo una vez que el usuario se registra.

Por otra parte, están los rectángulos azules oscuros que son otros elementos externos que interactúan con la plataforma Dory, estos pueden ser personas(usuarios) o sistemas de software que pueden usar los servicios que les brinda esta.

Cada rectángulo en el diagrama tiene una descripción del rol que juega.

Puede acceder al diagrama a través del siguiente enlace:

[**https://ginelect.net/dory/manual-tecnico/diagrama-contexto-sistema-dory.svg**](https://ginelect.net/dory/manual-tecnico/diagrama-contexto-sistema-dory.svg)

**2. Conocer los contenedores de la plataforma Dory (Nivel 2)**

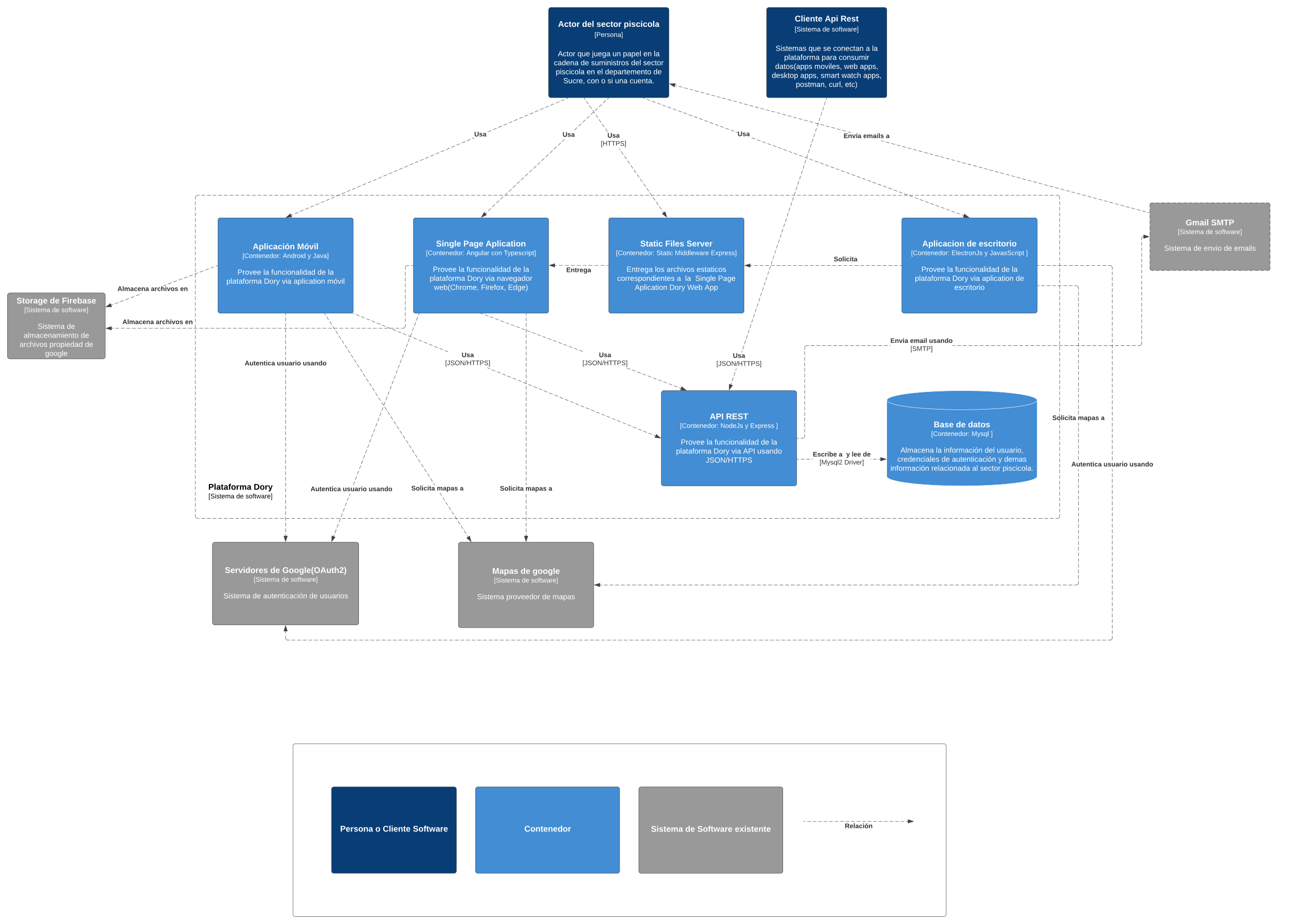


Imagen 2 Diagrama de contenedores

Este diagrama (Imagen 2 Diagrama de contenedores) es el resultado de hacer “zoom” en rectángulo azul claro que aparece en el diagrama de la *Imagen 1 Diagrama de contexto de la plataforma Dory* que representa la plataforma Dory. Los contenedores pueden ser aplicaciones móviles, aplicaciones web, bases de datos, etc.

En este diagrama podemos ver que la plataforma esta compuesta por un API-REST que es su corazón y que soporta un conjunto de servicios que son consumidos por otros contenedores como la aplicación móvil, La Single Page Aplication o aplicación web, la aplicación de escritorio e incluso sistemas de software externo como son los clientes API-REST.

La aplicación móvil, La single Page aplication o aplicación web, la aplicación de escritorio son clientes que permiten usuario final (piscicultor, pescador, proveedor, etc.) acceda a los servicios de valor que ofrece la plataforma vía un celular, navegador y aplicación de escritorio.

Otro contenedor en el diagrama es el Static Files Server este es un servidor web que sirve los archivos js, css y html de la aplicación web cuando el usuario ingresa a la url de la aplicación web en el navegador.

Por último, otro contenedor que hace parte de la plataforma es la base de datos MySQL. Esta almacena la información del usuario, credenciales de autenticación, granjas, productos y demás información relacionada con el sector piscícola.

Puede acceder al diagrama a través del siguiente enlace:

**<https://ginelect.net/dory/manual-tecnico/diagrama-de-contenedores-dory.svg>**

**3. Conocer el despliegue de los contenedores en los servicios (Nivel 2)**

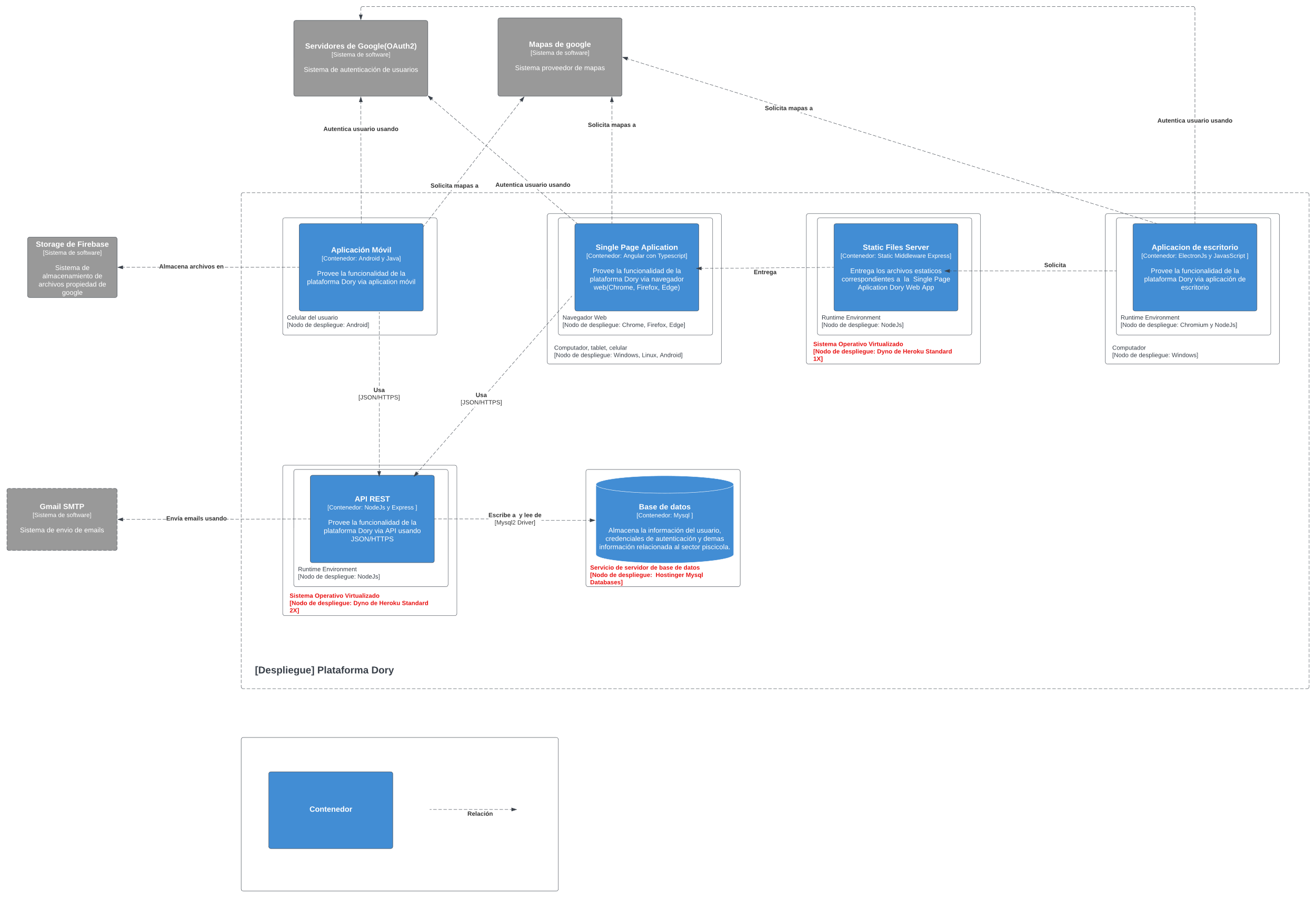


Imagen 3 Diagrama de despliegue

Este diagrama nos muestra donde se están ejecutando físicamente cada uno de los contenedores que componen a la plataforma Dory. Por ejemplo, la aplicación se ejecuta sobre un móvil con sistema operativo Android, la aplicación de escritorio se ejecuta sobre Chromium y Node.js y asi sucesivamente.

En este contexto cuando hablamos de contenedor no se hace referencia a un contenedor Docker en cambio un contenedor puede ser una aplicación, una base de datos, etc.

A continuación, se listarán cada uno de los nodos de despliegue (donde se ejecutan físicamente) de cada uno de los contenedores:

* Aplicación móvil

Nodo de despliegue: Sistema operativo Android

* Aplicación de escritorio

Nodo de despliegue:  
Chromium y Node.js

La aplicación de escritorio se desarrolló usando el framework Electron el cual utiliza el motor de renderizado de Chromium, el mismo motor detrás del navegador Google Chrome, para representar y mostrar la interfaz de usuario de la aplicación. Esto permite que las aplicaciones creadas con Electron tengan capacidades avanzadas, similares a las de un navegador web moderno. Esto lo combina con el entorno de tiempo de ejecución de Node.js lo que proporciona la capacidad de ejecutar código JavaScript tanto en el proceso principal (back-end) como en procesos de renderizado (front-end). Por ejemplo Node.js permite que las aplicaciones Electron accedan a funciones del sistema operativo, archivos locales, red y otras capacidades de nivel del sistema.

* Single Page Aplication (aplicación web Dory)

Nodo de despliegue:

Chrome, Firefox, Edge

* Static File Server

Nodo de despliegue:

Node.js sobre un Dyno Standard 1X en el servicio Heroku

(<https://www.heroku.com/>)

* API-REST

Nodo de despliegue:

Node.js sobre un Dyno Standard 2X en el servicio Heroku

(<https://www.heroku.com/>)

* Base de datos (MySQL)

Nodo de despliegue:

Hostinger MySQL Databases

(<https://www.hostinger.com/>)

Puede acceder al diagrama a través del siguiente enlace:

[**https://ginelect.net/dory/manual-tecnico/diagrama-despliegue-dory.svg**](https://ginelect.net/dory/manual-tecnico/diagrama-despliegue-dory.svg)

**4. Conocer el diagrama de componentes de la Single Page Application (SPA) (Nivel 3)**

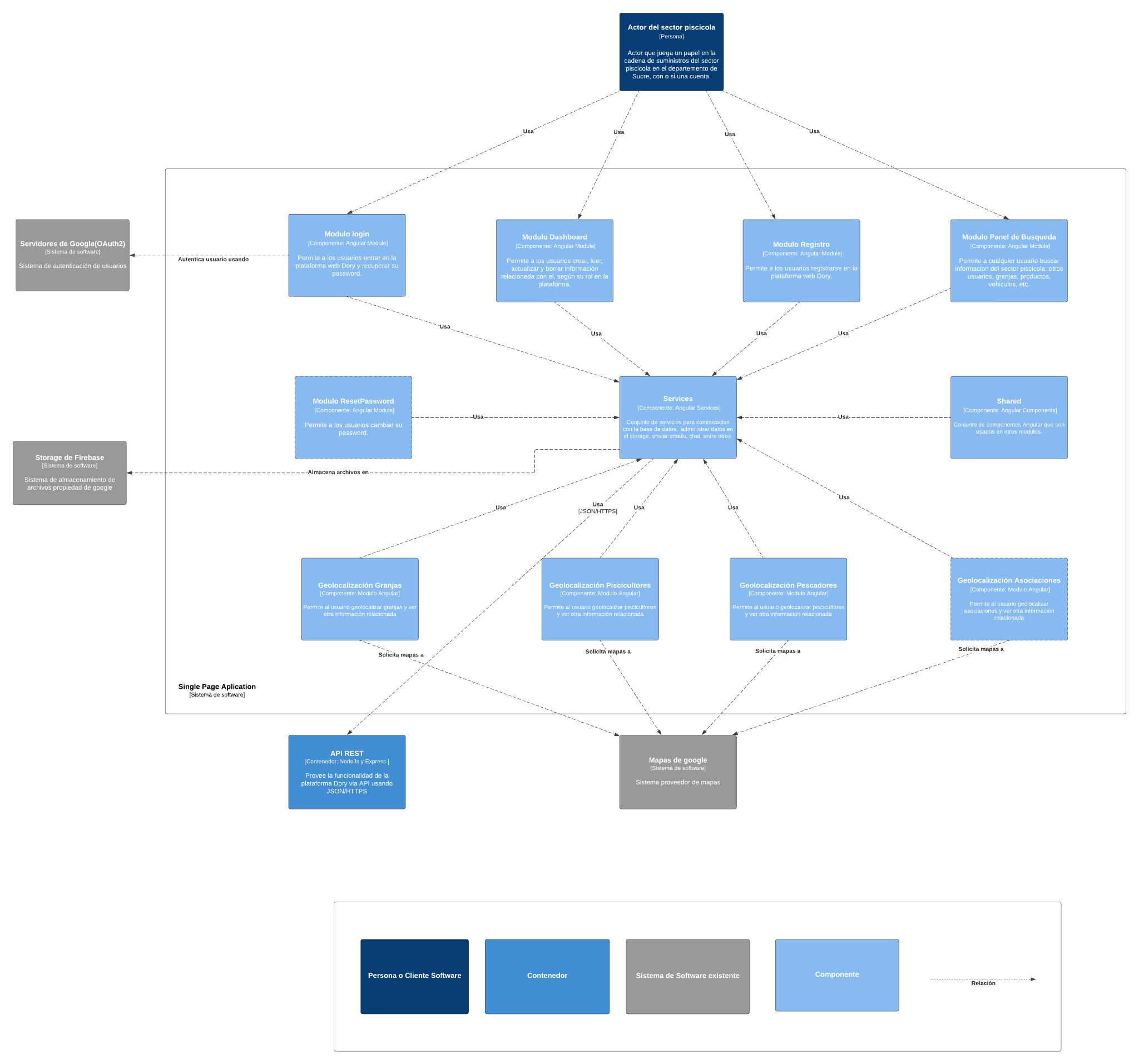


Imagen Diagrama de componentes Single Page Application (SPA)

La SPA se compone de módulos Angular que agrupan diferentes funcionalidades. Estos hacen uso de servicios inyectables angular para acceder a información y archivos proporcionados por el API REST y del Storage de Firebase respectivamente. También algunos módulos se comunican con otros sistemas externos directamente. Como es el caso del Módulo de Login que se comunica con los servidores de Google (OAuth2) para realizar la autenticación de un usuario con cuenta de Google o los módulos de geolocalización que hacen uso de los Mapas de Google.

Puede acceder al diagrama a través del siguiente enlace:

[**https://ginelect.net/dory/manual-tecnico/diagrama-de-componentes-spa.svg**](https://ginelect.net/dory/manual-tecnico/diagrama-de-componentes-spa.svg)

**5. Conocer el diagrama de componentes de la Aplicación API (API REST) (Nivel 3)**

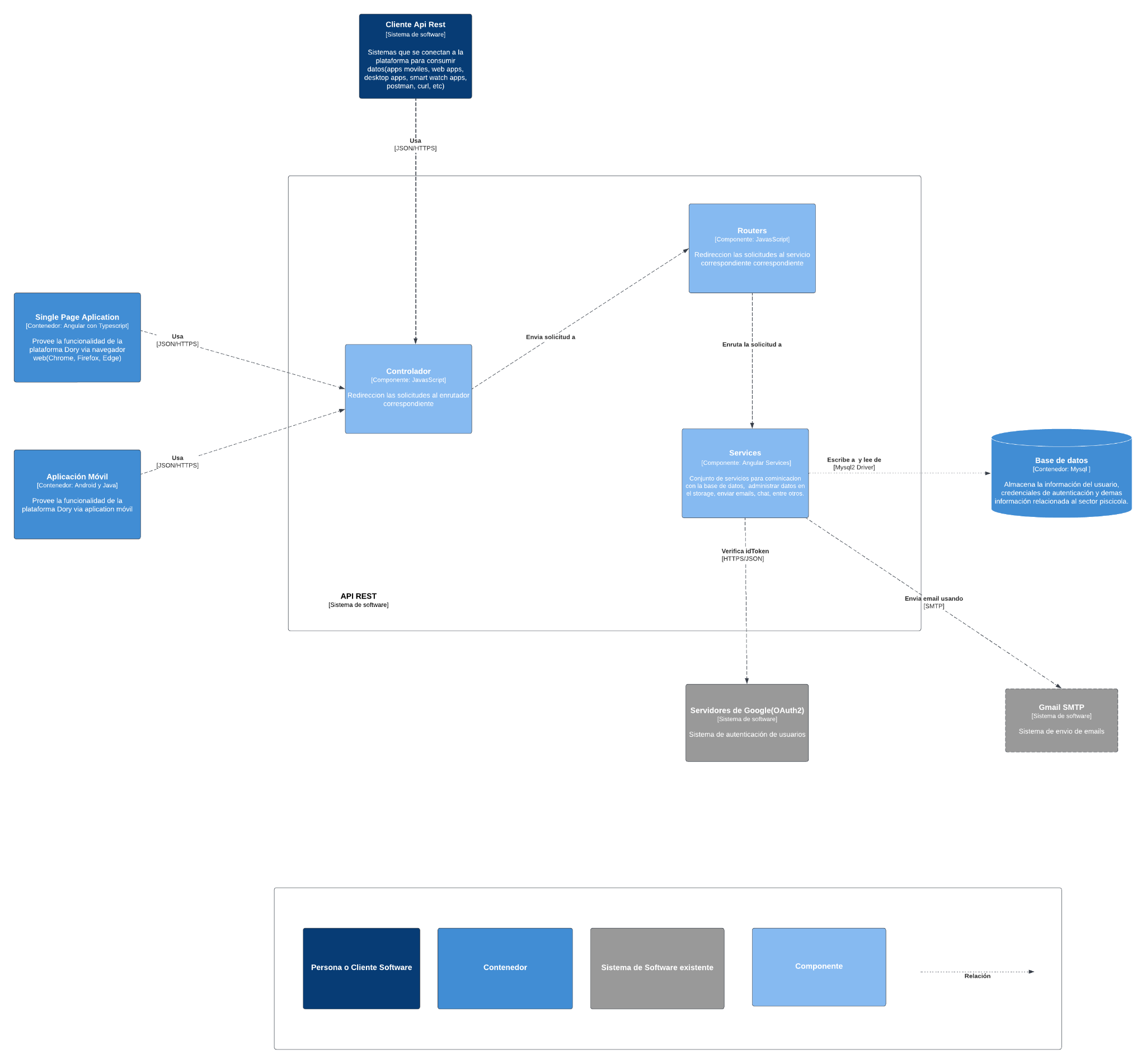


Imagen Diagrama de componente de la API-REST

El API REST se compone de un Controlador, Routers y Servicios. El controlador recibe solicitudes JSON/HTTPS a través de un endpoint específico y con base en esto envía la solicitud al Router correspondiente. Este último sabe qué servicio usará para resolver la solicitud y se la comunica. El servicio realiza una consulta (query) a la base de datos usando el driver Mysql2 y espera la respuesta. Cuando la recibe arma un objeto JSON y envía la respuesta al cliente a través del protocolo HTTPS.

Puede acceder al diagrama a través del siguiente enlace:

[**https://ginelect.net/dory/manual-tecnico/diagrama-de-componentes-api-rest.svg**](https://ginelect.net/dory/manual-tecnico/diagrama-de-componentes-api-rest.svg)

**6. Conocer el diagrama de componentes aplicación móvil (Nivel 3)**

La App móvil Android se compone de vistas o pantallas (Activitys/Fragments) que componen la interfaz de usuario. Estas vistas son manejadas por controladores que se comunican con el modelo de datos para acceder a información y archivos proporcionados por el API REST y del Storage de Firebase respectivamente. También algunas vistas se comunican con otros sistemas externos directamente. Como es el caso de la vista de Login que se comunica con los servidores de Google(OAuth2) para realizar la autenticación de un usuario con su cuenta de Google o las vistas de geolocalización que hacen uso de los Mapas de Google.

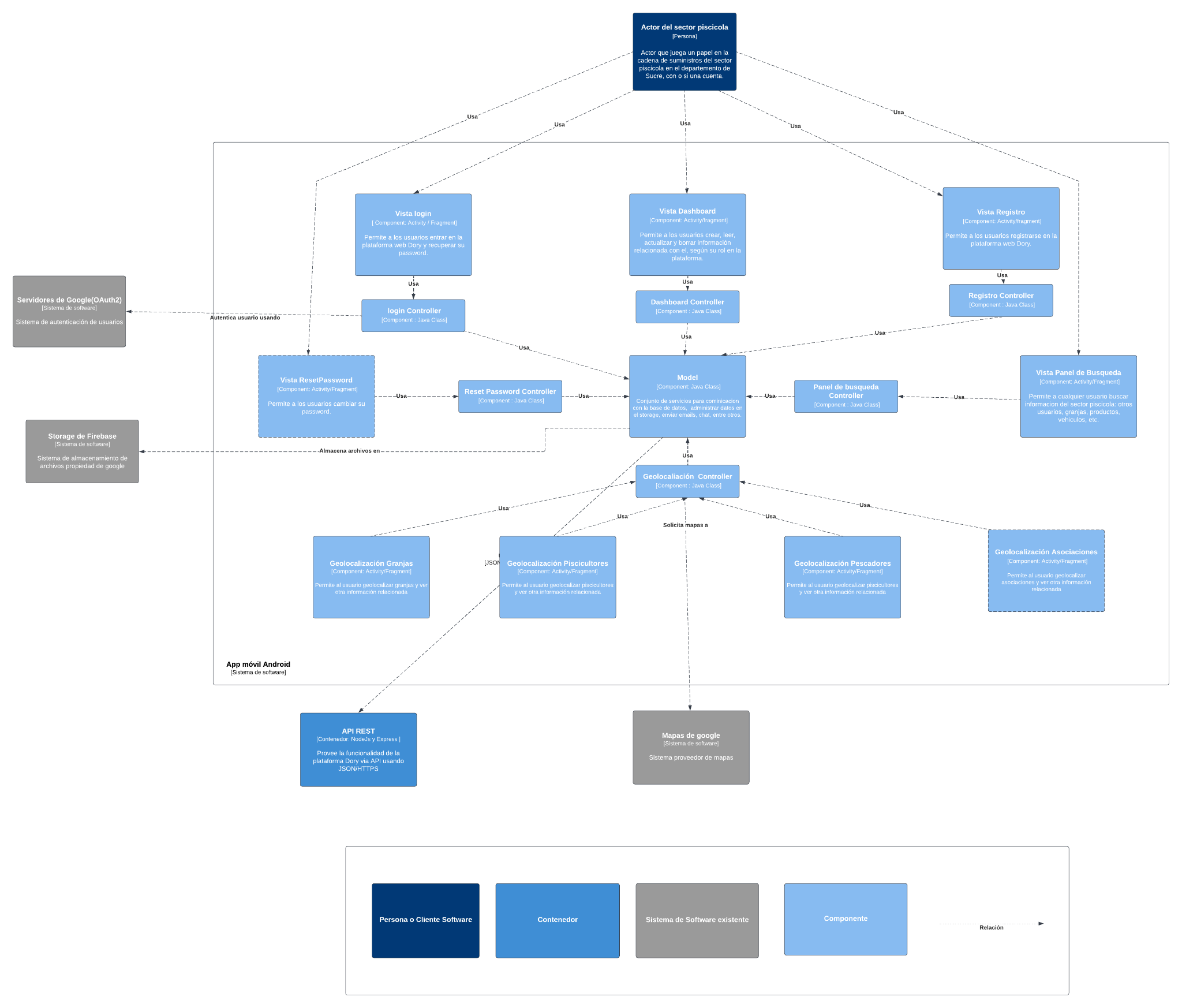


Imagen Diagrama de componentes de la app móvil.

Puede acceder al diagrama a través del siguiente enlace:

[**https://ginelect.net/dory/manual-tecnico/diagrama-componentes-mobile.svg**](https://ginelect.net/dory/manual-tecnico/diagrama-componentes-mobile.svg)

**7. Conocer el diagrama de componentes aplicación de escritorio**

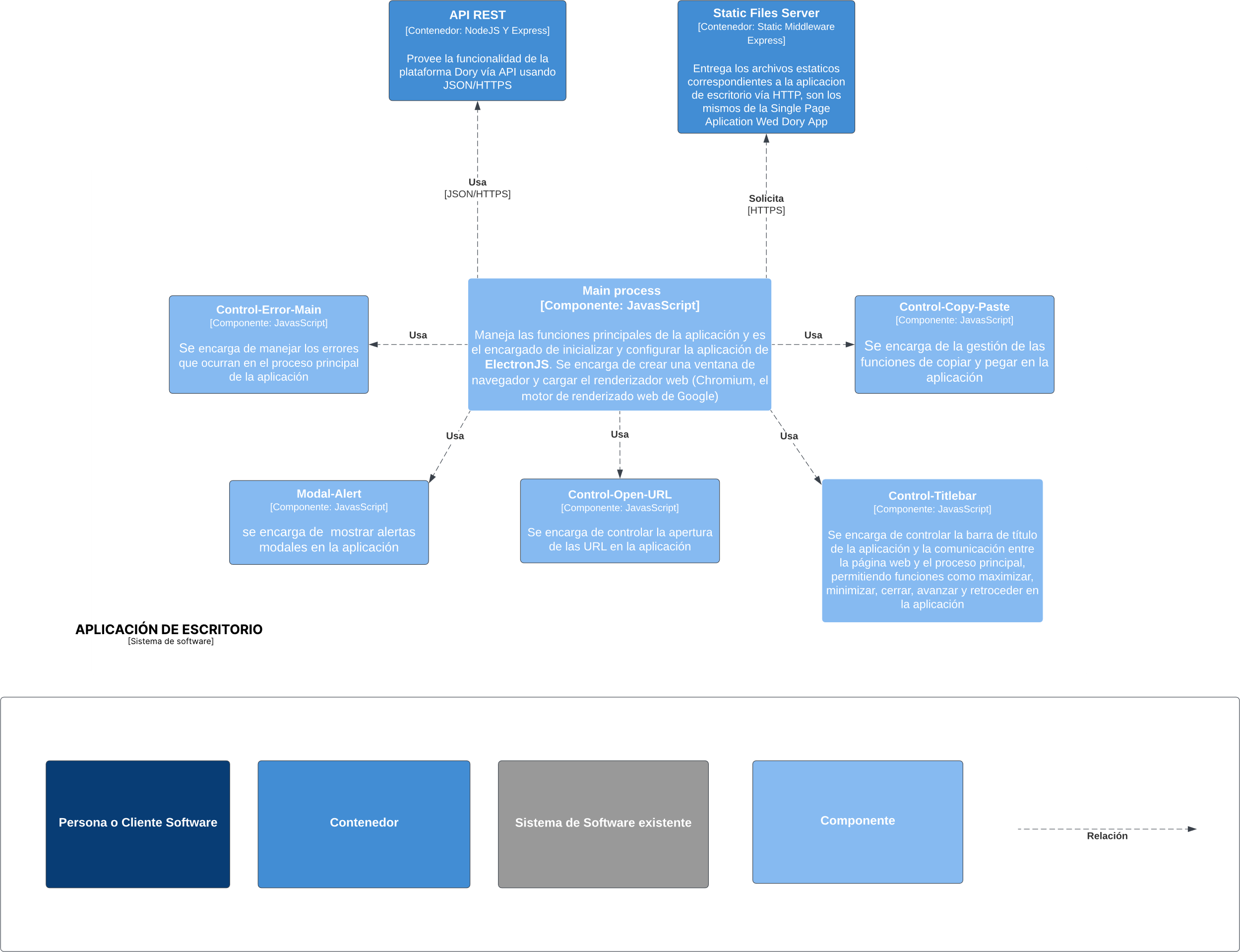


Imagen Diagrama de componentes de la app de escritorio.

Puede acceder al diagrama a través del siguiente enlace:

[**https://ginelect.net/dory/manual-tecnico/diagram-componentes-desktop.svg**](https://ginelect.net/dory/manual-tecnico/diagram-componentes-desktop.svg)