**Sistema gestor de acceso a laboratorios y aulas equipadas de las instituciones educativas.**

Su función principal es permitir al administrador, a través de una aplicación web, controlar o restringir el acceso a un laboratorio o aula equipada al interior de la institución dependiendo de su aforo y de forma opcional, considerando la temperatura del usuario.

Esto representa una gran ventaja puesto que permite acceder a la aplicación desde cualquier dispositivo con un navegador web, permitiendo obtener análisis estadísticos que lleven a la institución a establecer medidas que considere pertinentes generadas por el semáforo de riesgo epidemiológico cuando exista.

**Web APP (Aplicación web)**

Una Web APP es una aplicación que se ejecuta en un servidor web y que se puede utilizar desde el navegador, como Chrome, Safari, Edge, Firefox, Opera, etc.

Ventajas y Desventajas

Una APP nativa se instala directamente en el dispositivo, ya sea teléfono, ordenador, Tablet, etc. Esta diñada para usar los recursos del dispositivo, como la memoria RAM o la CPU, y claramente necesita ser instalada para que funcione. Además, estas se desarrollan teniendo en cuenta la plataforma donde se va a instalar (Windows, macOS, Android, ¡OS, etc.), hay versiones compatibles para cada sistema.

Por el contrario, una Web APP se ejecuta en un servidor web y se accede a ella desde un a navegar, por lo que no necesitan ser instaladas en el dispositivo. Además, son desarrolladas con lenguajes propios del desarrollo web (HTML, CCS, JavaScript, etc.). por lo que son compatibles con cualquier plataforma.

Una de la más grande desventaja de usar una Web APP es la necesidad de una conexión a internet para poder funcionar. Si bien existen apps desarrolladas de modo que puedan utilizarse sin conexión de manera temporal, al final la conexión será necesaria.

**Herramientas para del desarrollo de la aplicación web**

Para la creación de la aplicación web se utiliza Nodejs, MySQL, Express y otras tecnologías de JavaScript. En esta aplicación web se contará con un CRUD de nodejs más un Login/Registro utilizando MySQL, además de proteger nuestras rutas del servidor y estilizar nuestra aplicación con CSS, Bootstrap4, y Handlebars.

A continuación, se describen los conceptos necesarios para comprender mejor el desarrollo de la página web.

**Nodejs** [**https://developer.mozilla.org/es/docs/Learn/Server-side/Express\_Nodejs/Introduction**](https://developer.mozilla.org/es/docs/Learn/Server-side/Express_Nodejs/Introduction)

Node.js es un es un entorno que trabaja en tiempo de ejecución, de código abierto, multiplataforma, que permite a los desarrolladores crear toda clase de herramientas de lado servidor y aplicaciones en JavaScript. La ejecución en tiempo real está pensada para usarse fuera del contexto de un explorador web (es decir, ejecutarse directamente en una computadora o sistema operativo de servidor). Como tal, el entorno omite las APIs de JavaScript específicas del explorador web y añade soporte para APIs de sistema operativo más tradicionales que incluyen HTTP y bibliotecas de sistemas de ficheros

Desde una perspectiva de desarrollo de servidor web, Node tiene un gran número de ventajas:

* ¡Gran rendimiento! Node ha sido diseñado para optimizar el rendimiento y la escalabilidad en aplicaciones web y es un muy buen complemento para muchos problemas comunes de desarrollo web (ej, aplicaciones web en tiempo real).
* El código está escrito en "simple JavaScript", lo que significa que se pierde menos tiempo ocupándose de las "conmutaciones de contexto" entre lenguajes cuando estás escribiendo tanto el código del explorador web como del servidor.
* JavaScript es un lenguaje de programación relativamente nuevo y se beneficia de los avances en diseño de lenguajes cuando se compara con otros lenguajes de servidor web tradicionales (ej, Python, PHP, etc.) Muchos otros lenguajes nuevos y populares se compilan/convierten a JavaScript de manera que puedes también usar CoffeeScript, ClosureScript, Scala, LiveScript, etc.
* El gestor de paquetes de Node (NPM del inglés: Node Packet Manager) proporciona acceso a cientos o miles de paquetes reutilizables. Tiene además la mejor en su clase resolución de dependencias y puede usarse para automatizar la mayor parte de la cadena de herramientas de compilación.
* Es portable, con versiones que funcionan en Microsoft Windows, OS X, Linux, Solaris, FreeBSD, OpenBSD, WebOS, y NonStop OS. Además, está bien soportado por muchos de los proveedores de hospedaje web, que proporcionan infraestructura específica y documentación para hospedaje de sitios Node.
* Tiene un ecosistema y comunidad de desarrolladores de terceros muy activa, con cantidad de gente deseosa de ayudar.

Otras tareas comunes de desarrollo web no están directamente soportadas por el mismo Node. Si quieres añadir el manejo específico de diferentes verbos HTTP (ej, GET, POST, DELETE, etc.), gestionar de forma separada las peticiones por medio de diferentes direcciones URL ("rutas"), servir ficheros estáticos o usar plantillas para crear la respuesta de forma dinámica.

**Express https://developer.mozilla.org/es/docs/Learn/Server-side/Express\_Nodejs/Introduction**

Express es el framework web más popular de Node, y es la librería subyacente para un gran número de otros frameworks web de Node populares. Proporciona mecanismos para:

* Escritura de manejadores de peticiones con diferentes verbos HTTP en diferentes caminos URL (rutas).
* Integración con motores de renderización de "vistas" para generar respuestas mediante la introducción de datos en plantillas.
* Establecer ajustes de aplicaciones web como qué puerto usar para conectar, y la localización de las plantillas que se utilizan para renderizar la respuesta.
* Añadir procesamiento de peticiones "middleware" adicional en cualquier punto dentro de la tubería de manejo de la petición.

A pesar de que Express es en sí mismo bastante minimalista, los desarrolladores han creado paquetes de middleware compatibles para abordar casi cualquier problema de desarrollo web. Hay librerías para trabajar con cookies, sesiones, inicios de sesión de usuario, parámetros URL, datos POST, cabeceras de seguridad y muchos más.

**Bootstrap https://rockcontent.com/es/blog/bootstrap/#:~:text=Bootstrap%20es%20un%20framework%20front,dispositivo%20utilizado%20por%20el%20usuario.**

Bootstrap es un framework front-end utilizado para desarrollar aplicaciones web y sitios mobile first, o sea, con un layout que se adapta a la pantalla del dispositivo utilizado por el usuario.

El framework combina CSS y JavaScript para estilizar los elementos de una página HTML. Permite mucho más que, simplemente, cambiar el color de los botones y los enlaces.

Esta es una herramienta que proporciona interactividad en la página, por lo que ofrece una serie de componentes que facilitan la comunicación con el usuario, como menús de navegación, controles de página, barras de progreso y más.

Además de todas las características que ofrece el framework, su principal objetivo es permitir la construcción de sitios web responsive para dispositivos móviles.

Esto significa que las páginas están diseñadas para funcionar en desktop, tablets y smartphones, de una manera muy simple y organizada.

**Complementos de Express (Dependencias de la aplicación web)**

**express-handlebars:** es la integración del motor de plantillas handlebars en express.

**express-session:** administra las sesiones de nuestra app, necesario para autenticar a un usuario.

**mysql:** es un modulo de npm para conectarse y hacer consultas a la base de datos.

**express-mysql-session:** este módulo almacenara las sesiones en la base de datos, en lugar del servidor.

**morgan**: permite crear logs o mensajes de que es lo que las aplicaciones clientes están pidiendo al servidor.

**bcryptjs:** este módulo permite cifrar las contraseñas de los usuarios antes de guardarlos en la base de datos.

**passport:** es un modulo para autenticar, y manejar el proceso de login de un usuario en nuestra aplicación (Facebook, Google, etc.).

**passport-local:**  es un complemento de passport para autenticar a los usuarios con nuestra propia base de datos.

**connect-flash:** permite mostrar mensajes de error y éxito cuando el usuario realice una operación.

**express-validator:** es un módulo para validar los datos del usuario que nos envía desde la aplicación cliente.

**nodemon:**  es una herramienta que ayuda a desarrollar aplicaciones basadas en node.js reiniciando automáticamente la aplicación node cuando se detecten cambios de archivos en el directorio.

**GABANNI - Aplicación Web**

GABANNI es una aplicación web que se ejecuta en un servidor, y se puede acceder a ella desde un navegador web, lo que facilita el acceso a los usuarios por ser una aplicación multiplataforma.

Esta aplicación web se compone de dos partes:

La primera consta de un dashboard, que no ayuda a hacer un seguimiento de los principales funcionamientos del circuito. Este nos servirá para tomar decisiones que optimicen la estrategia de los espacios académicos donde se implementen (Módulo de procesamiento de los datos recolectados y control de aforos).

La segunda parte consta de la gestión de datos de nuestra base de datos, en la cual se almacena la información de los usuarios que consumen el sistema (Alumnos, Profesores, Personal académico, etc.), y de los registros generados por el circuito. La aplicación web nos permite realizar las operaciones fundamentales como crear, leer, actualizar y borrar registros.

A continuación (ver figura #), se describe la estructura general del código fuente de la aplicación web GABANNI.

Imagen que contiene Gráfico

Descripción generada automáticamente

Figura 0. Estructura del código fuente de la aplicación Web (GABANNI).

A continuación, se describe con más detalles cada uno de los archivos que componen la aplicación web,

|  |  |
| --- | --- |
| **Proyecto de Integración APP** | |
| **database**  Aquí se almacena los códigos para la creación de la base de datos. |  |
| **src**  Aquí se almacena toda la estructura en general de la aplicación web, |  |
| **node\_modules**  Aquí se almacenas todos los componentes necesarios para el buen funcionamiento de los módulos externos, de los cuales depende la aplicación web para su buen funcionamiento. |  |
| **package**  es un archivo generado automáticamente. Se coloca toda la información que se conoce sobre la aplicación web |  |
| **package-lock.**  es un archivo generado automáticamente cuando se instalan paquetes o dependencias en el proyecto |  |

|  |  |
| --- | --- |
| **database** | |
| **cerradura**  Aquí se guarda la estructura general de la base de datos. |  |
| **cerradura\_datos**  Aquí se guarda la estructura general de la base de datos con datos de prueba. |  |

|  |  |
| --- | --- |
| **src** | |
| **lib**  Aquí se almacenan los archivos JS que contienen funciones que son ocupadas en ocasiones por los archivos que dan la funcionalidad a la aplicación web. | **Imagen que contiene firmar, exterior, tren, fondo  Descripción generada automáticamente** |
| **public**  Aquí se almacenan los archivos públicos (Estilo e Imágenes). | **Imagen que contiene firmar, exterior, tren, fondo  Descripción generada automáticamente** |
| **profile**  Aquí se almacenas todos los archivos que le dan funcionalidad a la página, estos están estructurados para tener una mejor comprensión de su funcionamiento y uso en la aplicación. | **Imagen que contiene firmar, exterior, tren, fondo  Descripción generada automáticamente** |
| **Views**  Aquí se almacenan las carpetas donde en cada una de ellas se encuentran las vistas de la aplicación web. estos están estructurados para tener una mejor comprensión de su funcionamiento y uso en la aplicación. | **Imagen que contiene firmar, exterior, tren, fondo  Descripción generada automáticamente** |
| **keys**  En este archivo se almacenan la información para hacer posible la conexión a la base de datos. |  |
| **index**  Este archivo es el principal, pues aquí es donde se inicializa todos los componentes de la aplicación, además se declaran variables globales y se dan las direcciones de las rutas donde se encentra cada parte del código en general. |  |
| **Database**  En este archivo se establece la conexión con la base de datos. |  |

|  |  |
| --- | --- |
| **lib** | |
| **auth**  En este archivo se encuentran las funciones para comprobar si un usuario esta logueado o no. | **Icono  Descripción generada automáticamente** |
| **handlebars**  En este archivo se encuentran las funciones para convertir cadenas de texto true/false leídas previamente desde la base de datos a activado o desactivado respectivamente. | **Icono  Descripción generada automáticamente** |
| **helpers**  En este archivo se encuentran las funciones para el cifrado de contraseñas de los usuarios cuando se registran en el sistema, Además también se encuentra la función encargada para comprobar el “match” de las contraseñas cuando un usuario desea acceder. | **Icono  Descripción generada automáticamente** |
| **Passport**  En este archivo se encuentran las funciones que sirven para cuando un usuario se desea loguear, aquí se comprueba que la información escrita sea la misma almacenada en la base de datos. También se encuentra la función para registrar un nuevo usuario con su contraseña ya encriptada. | **Icono  Descripción generada automáticamente** |

|  |  |
| --- | --- |
| **public** | |
| **css**  En esta carpeta se encuentra el archivo CSS para dar diseño a la aplicación web. | **Imagen que contiene firmar, exterior, tren, fondo  Descripción generada automáticamente** |
| **img**  En esta carpeta se encuentra imágenes que son usadas en la aplicación web para darle diseño a la misma. | **Imagen que contiene firmar, exterior, tren, fondo  Descripción generada automáticamente** |

|  |  |
| --- | --- |
| **css** | |
| **styles**  En este archivo se encuentra todo el diseño que se le da a la pagina en general como color de fondo, tamaño, etc. |  |

|  |  |
| --- | --- |
| **profile** | |
| **administradores**  Este archivo es el encargado de recibir todas las peticiones http (GET() y POST() ), de las vistas de los administradores, aquí mismo es donde se hacen la consultas hacia la base de datos. | **Icono  Descripción generada automáticamente** |
| **alumnos**  Este archivo es el encargado de recibir todas las peticiones http (GET() y POST() ), de las vistas de los alumnos, aquí mismo es donde se hacen las consultas hacia la base de datos. | **Icono  Descripción generada automáticamente** |
| **authentication**  Este archivo es el encargado de recibir todas las peticiones http (GET() y POST() ), de las vistas sigUp y sigIn, aquí mismo es donde se hacen las consultas hacia la base de datos. | **Icono  Descripción generada automáticamente** |
| **index**  Este archivo es el encargado de recibir todas las peticiones http (GET() y POST() ), de las vista index, aquí mismo es donde se hacen las consultas hacia la base de datos. | **Icono  Descripción generada automáticamente** |
| **Laboratorios**  Este archivo es el encargado de recibir todas las peticiones http (GET() y POST() ), de las vista de los laboratorios aquí mismo es donde se hacen las consultas hacia la base de datos. | **Icono  Descripción generada automáticamente** |
| **Profesores**  Este archivo es el encargado de recibir todas las peticiones http (GET() y POST() ), de las vista de los profesores, aquí mismo es donde se hacen las consultas hacia la base de datos. | **Icono  Descripción generada automáticamente** |

|  |  |
| --- | --- |
| **views - administradores** | |
| **edit**  Este archivo contiene la vista para la edición de la información de los administradores registrados. |  |
| **historial**  Este archivo contiene la vista para la eliminación de información de la base de datos. |  |
| **list**  Este archivo contiene la vista principal para la búsqueda, edición y eliminación de la información de cada administrador registrado. |  |

|  |  |
| --- | --- |
| **views - alumnos** | |
| **edit**  Este archivo contiene la vista para la edición de la información de los alumnos registrados. | **Icono  Descripción generada automáticamente** |
| **add**  Este archivo contiene la vista para agregar un nuevo registro de alumno a la base de datos. | **Icono  Descripción generada automáticamente** |
| **list**  Este archivo contiene la vista principal para la búsqueda, edición y eliminación de la información de cada alumno registrado. | **Icono  Descripción generada automáticamente** |

|  |  |
| --- | --- |
| **views - profesores** | |
| **edit**  Este archivo contiene la vista para la edición de la información de los profesores/personal registrados. | **Icono  Descripción generada automáticamente** |
| **add**  Este archivo contiene la vista para agregar un nuevo registro de profesores/personal a la base de datos. | **Icono  Descripción generada automáticamente** |
| **list**  Este archivo contiene la vista principal para la búsqueda, edición y eliminación de la información de cada profesor/personal registrado. | **Icono  Descripción generada automáticamente** |

|  |  |
| --- | --- |
| **views - auth** | |
| **signin**  Este archivo contiene la vista para el inicio de sesión (Login), de los usuarios administradores. | **Icono  Descripción generada automáticamente** |
| **signup**  Este archivo contiene la vista para poder registrar un nuevo registro de los usuarios administradores. | **Icono  Descripción generada automáticamente** |

|  |  |
| --- | --- |
| **views - partials** | |
| **navigation**  Este archivo contiene la vista de la barra de navegación que será visible en todas las vistas. | **Icono  Descripción generada automáticamente** |
| **message**  Este archivo contiene la vista de los mensajes de error/Satisfacción que se le muestran al usuario al realizar operaciones con resgistros. | **Icono  Descripción generada automáticamente** |

|  |  |
| --- | --- |
| **views - layouts** | |
| main  Este archivo contiene la estructura general de todas las vistas de la aplicación web, en ella se declaran los scripts necesarios para su funcionamiento, además es aquí donde se incrusta cada una de las diferentes vistas de la aplicación web. | **Icono  Descripción generada automáticamente** |

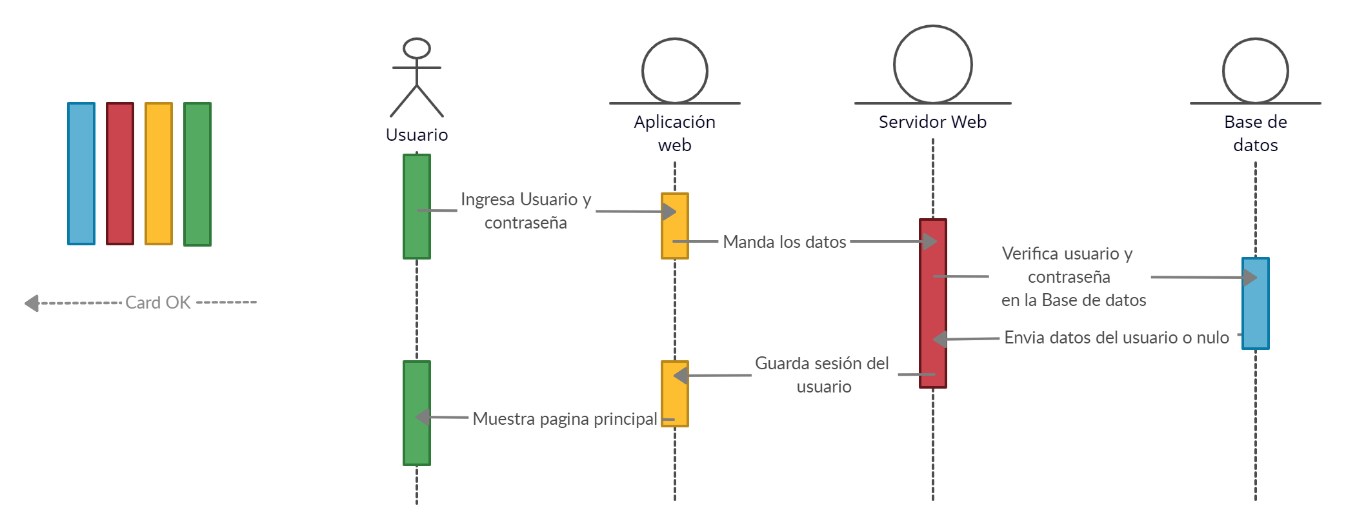
|  |  |
| --- | --- |
| **views - laboratorios** | |
| **edit**  Este archivo contiene la vista para la edición de la información de los laboratorios/aulas registradas. |  |
| **add**  Este archivo contiene la vista para agregar un nuevo registro de laboratorios/aulas a la base de datos. |  |
| **list**  Este archivo contiene la vista principal para la búsqueda, edición y eliminación de la información de cada laboratorio/aulas registrado. |  |
| **dash**  Este archivo contiene la vista general para cada panel de control (Node-RED) correspondiente a cada uno de los laboratorios registrados. |  |
| **dashAlumos**  Este archivo contiene la vista de la información de los alumnos que entraron y salieron de cada laboratorio, además nos muestra gráficamente cual es el número de asistencias de los últimos 5 días, también muestra gráficamente el número de asistencias del día por hora |  |
| **dashProfesores**  Este archivo contiene la vista de la información de los profesores/personal que entraron y salieron de cada laboratorio, además nos muestra gráficamente cual es el número de asistencias de los últimos 5 días, también muestra gráficamente el número de asistencias del día por hora |  |
| **TempAlumnos**  Este archivo contiene la vista de la información de los alumnos que intentaron ingresar al laboratorio con una temperatura superior a 37.5 grados centígrados, además nos muestra gráficamente cual es el número de temperaturas altas registradas los últimos 5 días, también muestra gráficamente el número de temperaturas altas registradas al día por hora. |  |
| **TempProfesores**  Este archivo contiene la vista de la información de los profesores/personal que intentaron ingresar al laboratorio con una temperatura superior a 37.5 grados centígrados, además nos muestra gráficamente cual es el número de temperaturas altas registradas los últimos 5 días, también muestra gráficamente el número de temperaturas altas registradas al día por hora. |  |

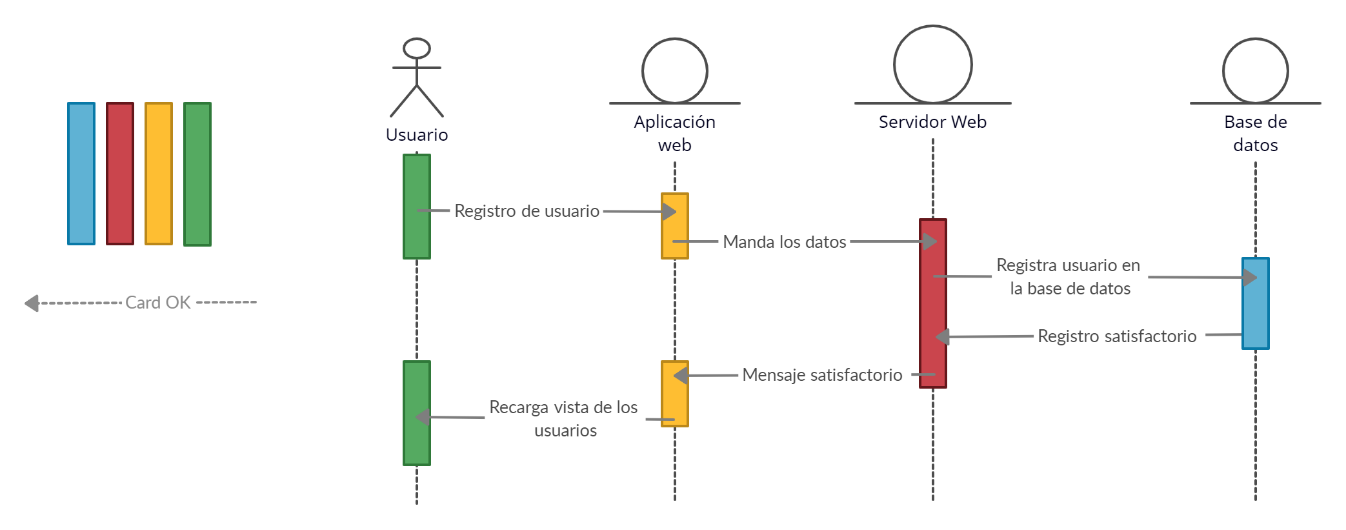
**Arquitectura del sistema de información web**

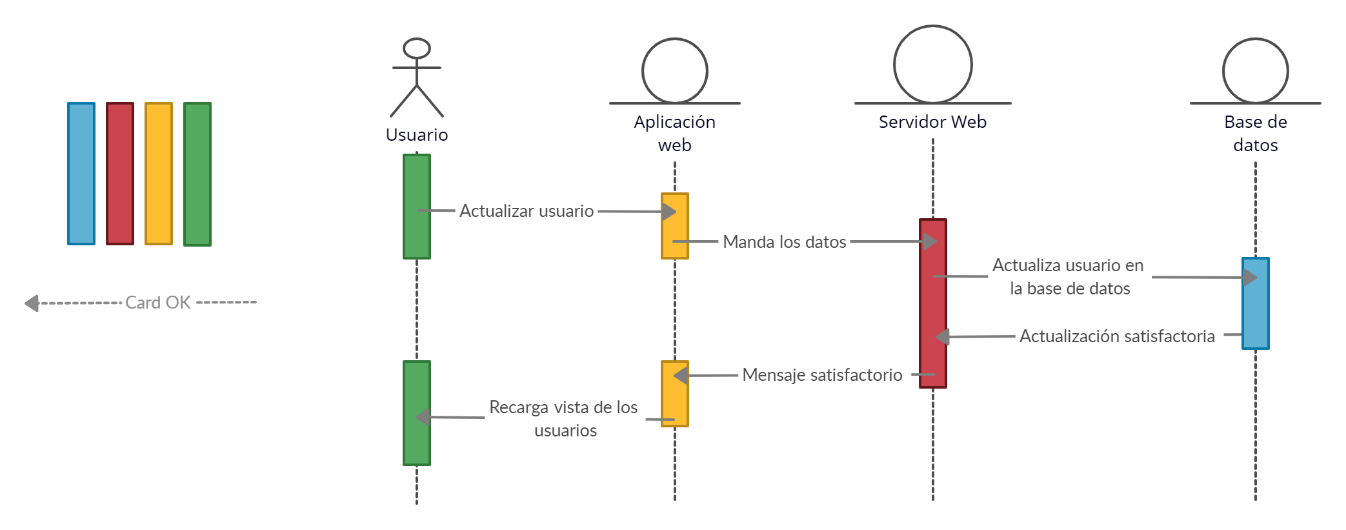
La Figura 5 muestra la arquitectura general del sistema web: la base de datos, los módulos que lo componen y la interacción de los actores.

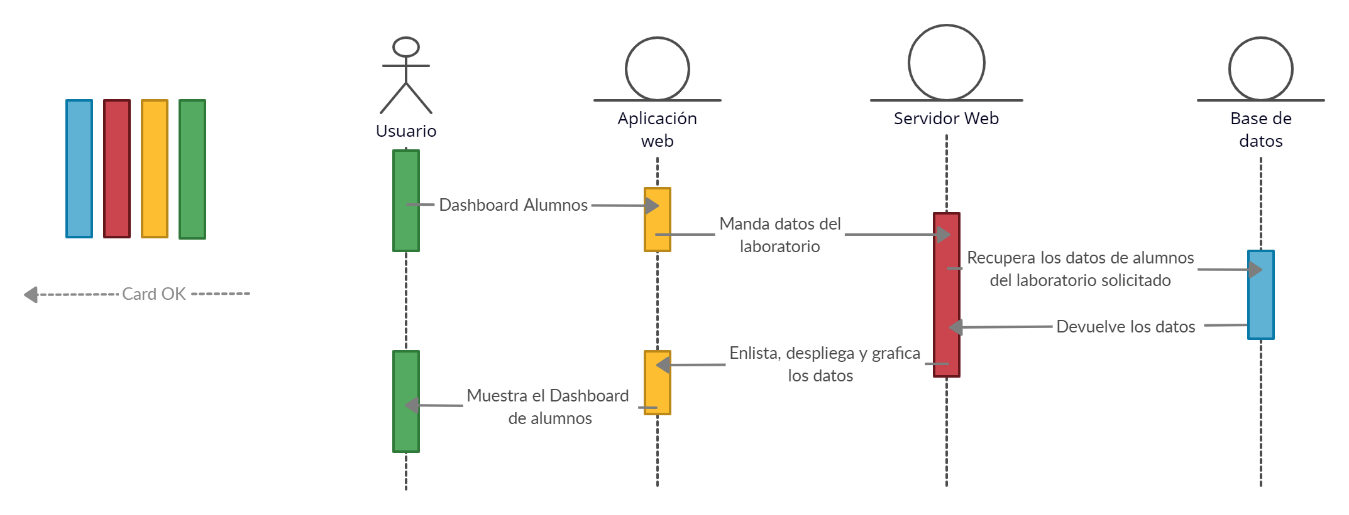
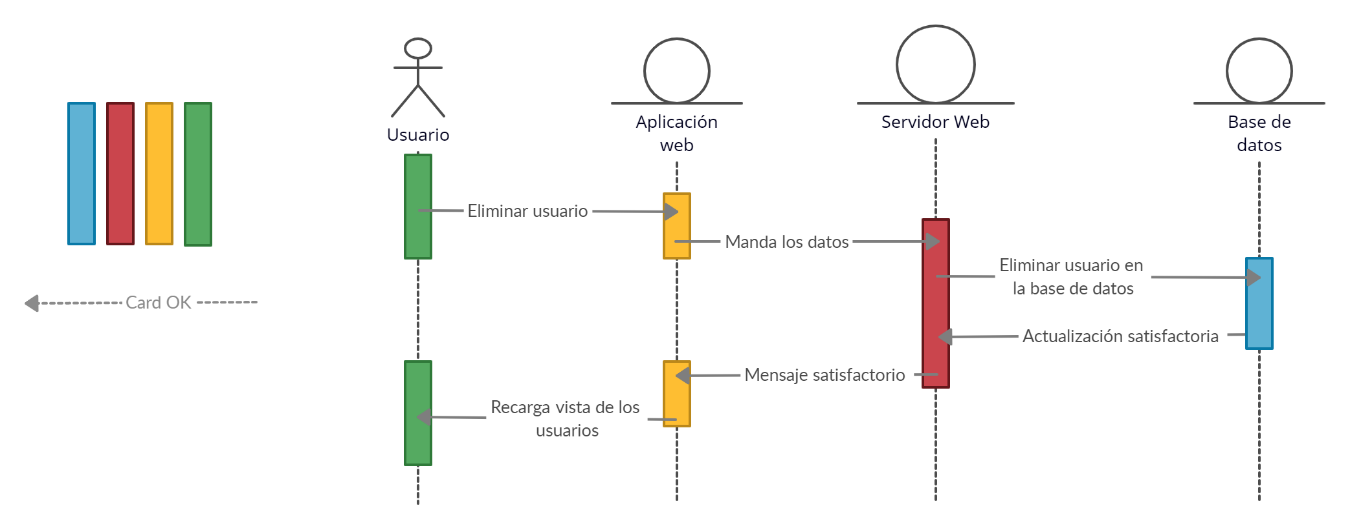
**Diagramas de secuencia de la aplicación web**

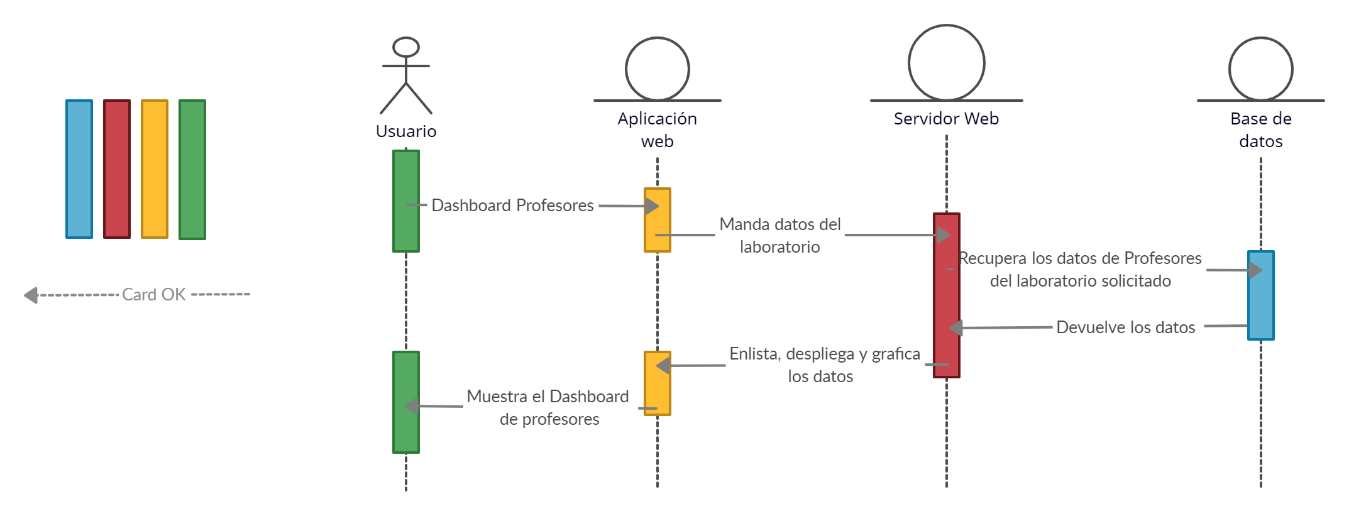
A continuación, se muestran los diagramas de secuencia del sistema, para obtener un panorama más amplio del diseño.

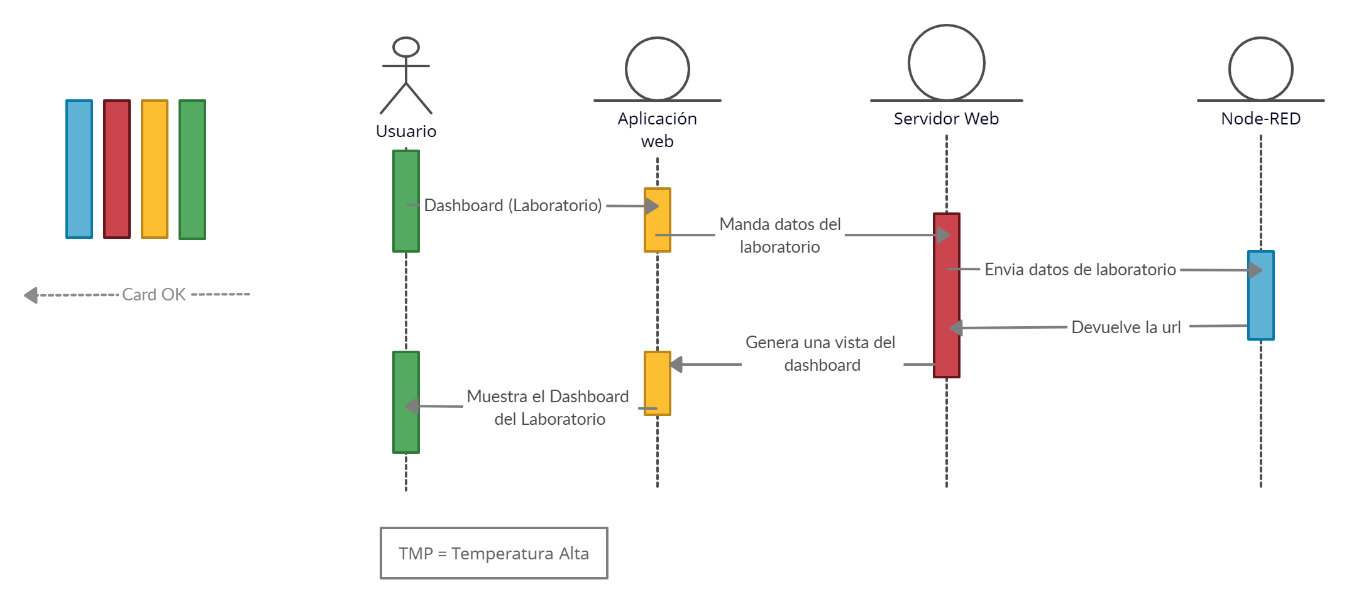




****

****

**Diagrama

Descripción generada automáticamente**