***GestY Presentación***

* Índice:
  + ¿Qué es GestY?
  + La infraestructura
  + Tecnologías
  + El WebService
  + Los clientes
  + Conclusión

1. *¿Qué es GestY?*

GestY es un proyecto diseñado para la gestión de proyectos y tareas asignadas a un empleado dentro de la empresa, asi como la posibilidad de hablar con los compañeros de una misma empresa a través de un chat integrado dentro de la misma aplicación o, en el caso que se necesite, una videoconferencia de forma rápida y sencilla.

GestY es más que una simple aplicación, es una Infraestructura creada a partir de 3 elementos ( WebService, MQTT Cloud y Clientes ).

La idea principal del proyecto nació a partir del uso diario de varias herramientas de trabajo y de tener que ir de una herramienta a otra para poder comunicarse con el resto de compañeros, gestionar y visualizar las tareas o crear una nueva reunion.

De esa idea nació

1. *La Infraestructura*

Como comentaba anteriormente, GestY va más allá de una aplicación, ya que, desde el primer día se buscó la posibilidad de poder acceder en cualquier momento y desde cualquier sitio a los datos, asi como poder comunicarte con los compañeros.

Para ello, se ha desarrollado la siguiente infraestructura:

* + WebService, o Servidor Central, es la base de la infraestructura, ya que, a través de él, vamos a poder acceder a los datos, asi como poder crear nuevos proyectos y tareas desde una página de administración en el mismo WebService.
  + Servicio de Mensajeria, el cual, en la idea principal del proyecto, iba a ir dentro del WebService, pero a la hora de desarrollar la idea y llevarla a un entorno de producción, se desechó ya que requería de una infraestructura muy técnica y se decidió buscar un servidor externo que tuviera la infraestructura idónea para esta herramienta.
  + Clientes, aplicaciones destinadas para los usuarios regulares, las cuales se dividen en dos tipos:
    - Aplicaciones clientes para gestión de proyectos y tareas, ideal para empleados de las empresas.
    - Aplicaciones administrativas las cuales disponen de todas las empresas, compañías y tareas de GestY, ideales para los administradores del proyecto.

1. *Tecnologías*

Una vez comentada la infraestructura, voy a hablar sobre las tecnologías utilizadas en el proyecto. Para ello, voy a seguir la misma infraestructura que mantiene el proyecto.

* + Webservice: En el WebService encontramos tecnologías como PHP, usando el framework Laravel, como página web, del cual hablaremos mas adelante y adentraremos en ella; o MariaDB, la cual va a ser nuestro sistema gestor de base de datos. Aunque en esta primera versión no este en uso, tambien encontramos un servicio de envio de correo electrónico SMTP, incorporado dentro de Laravel

* + Mensajeria: En el servicio de mensajería nos encontramos la tecnología MQTT, del nombre Mosquitto, una tecnología creada para el flujo de datos entre usuarios corrientes y dispositivos inteligentes, también denominados IoT o Internet of Things. Un ejemplo de flujo de datos con MQTT puede ser, por ejemplo, un sensor de temperatura de un reactor, el cual envía la temperatura a un movil cada 5/10 minutos. Ese flujo de datos unidireccional suele ser a través de MQTT.
  + Clientes: Las aplicaciones clientes estan divididas en dos tipos:
    - Aplicación para empleados: La aplicación para empleados esta desarrollada en Java, contando con una interfaz de usuario desarrollada en Java Swing. Cuenta también con una integración de MQTT en forma de chat bidireccional.
    - Aplicación para administradores: Desarrollada en Dart, un lenguaje de programación para moviles desarrollado por Google, el cual usa su propio framework, Flutter.

1. *El WebService*

El WebService es el servidor central de la infraestructura. En él se aloja nuestro Laravel, el cual es accesible a través de la URL <https://gesty.devf6.es/> y en el cual encontraremos la página de administración del proyecto.

Laravel va a hacer de intermediario entre las aplicaciones cliente y la base de datos, otorgando seguridad a las peticiones, y garantizando el acceso a los datos desde cualquier sitio.

Ya que Laravel va a hacer de intermediario entre las aplicaciones cliente y la base de datos, el puerto de acceso a la base de datos queda restringido a una única IP (Mostrar en prueba)

A su vez, Laravel gestiona peticiones a través de una API, la cual, gracias a un token de autentificación, es la encargada de gestionar los datos de las aplicaciones clientes.

1. *Los Clientes*

Como he comentado anteriormente, los clientes se encargan de la visualización de los datos por pantalla, ya que, quien realmente gestiona los datos, es el WebService.

Las aplicaciones cliente realizan peticiones HTTP a través del método POST, el cual nos proporciona incluir datos en el cuerpo

1. *Conclusión*

El desarrollo del ha sido muy laborioso, debido al querer incorporar varias tecnologías, asi como nuevas técnicas y buenas practicas de desarrollo. Por desgracia, el proyecto no esta lo completo que me gustaria, ya que falta depuración y diseños mas atractivos visualmente. Aun teniendo en cuenta esto, estoy orgulloso del proyecto, ya que he conseguido, funcionamente, la idea principal que tenia en mente desde el principio, como por ejemplo, un chat funcional o el flujo de datos a través de peticiones autentificadas.