Documentación Trabajo Final de Grado

ServiceCoin

Sebastià Adrover Llaneras

Tabla de contenido

1. Introducción

El proyecto que se presenta en este documento es una aplicación web destinada a ser usada por cualquier usuario con ordenador y conexión a internet.

Para hablar de todas las funcionalidades que ofrece nuestra aplicación deberemos dividir a los usuarios que la utilicen en dos grupos, los registrados con sesión iniciada y los usuarios medios sin registrar.

Para los usuarios sin registrar la aplicación permite:

- Ver todos los servicios de la aplicación de todos los usuarios registrados que ofrecen los usuarios registrados.
- Podrá filtrar esos servicios con una barra de búsqueda.
- El usuario podrá ver información más detallada de los servicios si estuviera interesado.
- El usuario podrá registrarse e iniciar sesión.

Para los usuarios ya registrados y con sesión iniciada:

- La aplicación permite que el usuario registre todos servicios a ofrecer que desee.
- Permite modificar y eliminar todos los servicios que él ha proporcionado.
- Todas las funcionalidades que se ofrecen anteriormente.

En un futuro esta aplicación ofrecerá un servicio de contratos virtuales el cual todavía no está implementado por falta de tiempo para que no haya trámites de terceros en caso de que alguien quiera contratar algún servicio ofrecido por alguno de nuestros usuarios.

Dicha funcionalidad se implementará utilizando el sistema de blockchain de Flow.

Historias de usuario:

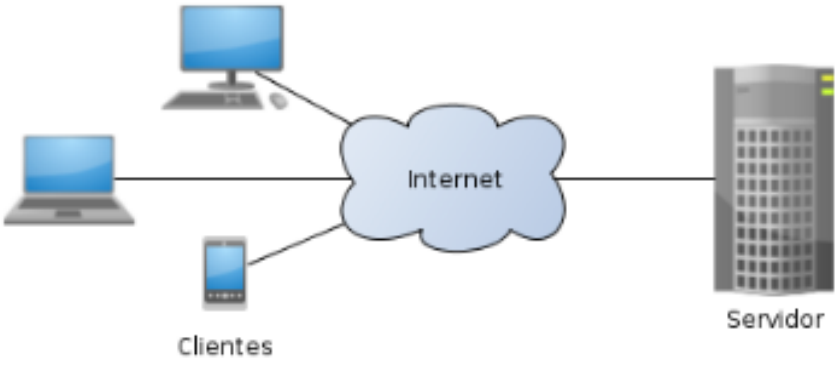
- El usuario puede ver los servicios que se ofrecen por todos los usuarios.
- El usuario puede ver los detalles de cualquier servicio.
- El usuario filtra los servicios.
- El usuario no puede registrarse. Las credenciales ya están usadas.
- El usuario se registra con otras credenciales.
- El usuario intenta iniciar sesión. Se equivoca de credenciales.
- El usuario inicia sesión con las credenciales correctas.
- El usuario registra un servicio.
- El usuario modifica el servicio creado.
- El usuario registra otro servicio.
- El usuario borra el servicio creado.

2. Descripción técnica

Las tecnologías usadas para este proyecto han sido:

- Node Js para la lógica de negocio, es decir para el servidor.
- React como Front Framework para el desarrollo de la interfaz gráfica, es decir el cliente.
- MongoDB como sistema de base de datos para almacenar todos los datos que registren los usuarios.
- Atlas MongoDB, que es un servicio que ofrece la desarrolladora de MongoDB que permite almacenar todos los datos de nuestros usuarios en la nube siendo mucho más cómodo en caso de que se quiera desplegar la aplicación.
- Mongoose que permite a nuestro servidor interaccionar con la base de datos de MongoDB.
- Express para que nuestro servidor gestione todas las peticiones HTTP de nuestro cliente.
- Chai y Mocha para testear la lógica de negocio del servidor.
- Visual Studio Code como Entorno de Desarrollo.

La aplicación implementa un modelo Cliente-Servidor:



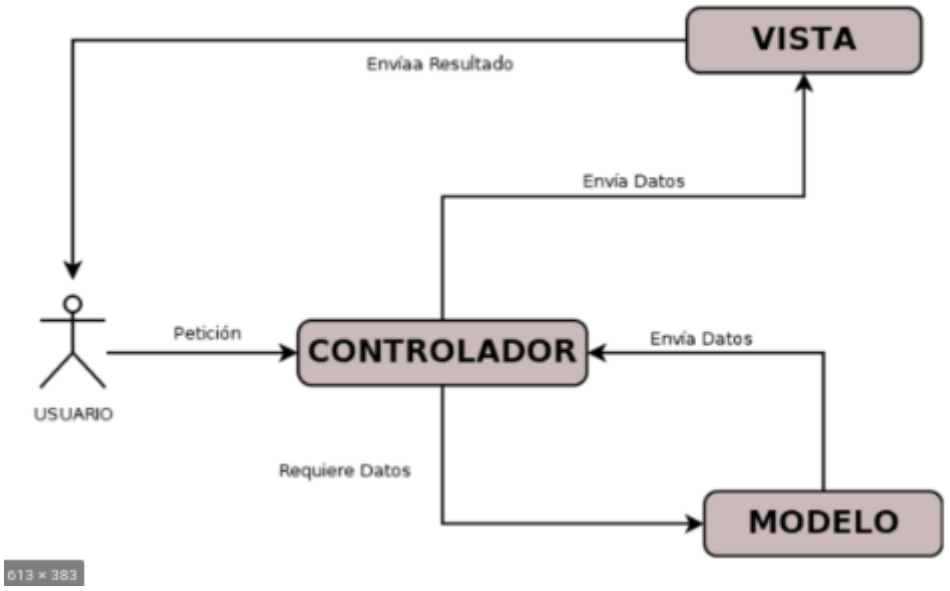
La arquitectura de cliente-servidor es un tipo de diseño de implementación software en la que las tareas se reparten entre los servidores y los clientes. El software cliente realiza peticiones al software servidor, que es quien le da respuesta.

En nuestra aplicación se ve reflejado dicho modelo ya que el servidor y el cliente están completamente divididos en repositorios distintos.

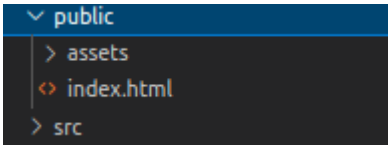
Aquí están ambos enlaces de ambos repositorios subidor en GitHub:

- Cliente
- Servidor

La aplicación también usa arquitectura Modelo-Vista-Controlador:

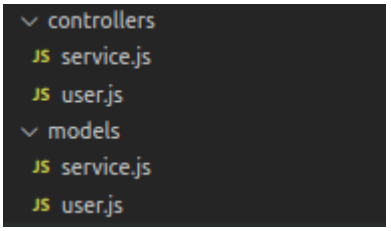


La arquitectura de MVC es un patrón de diseño que divide la aplicación en tres partes conectadas, la vista (que es la vista que ofrece nuestro cliente ya mencionado anteriormente), el controlador que gestiona todas la peticiones del cliente y el modelo que permite la conexión a la base de datos.

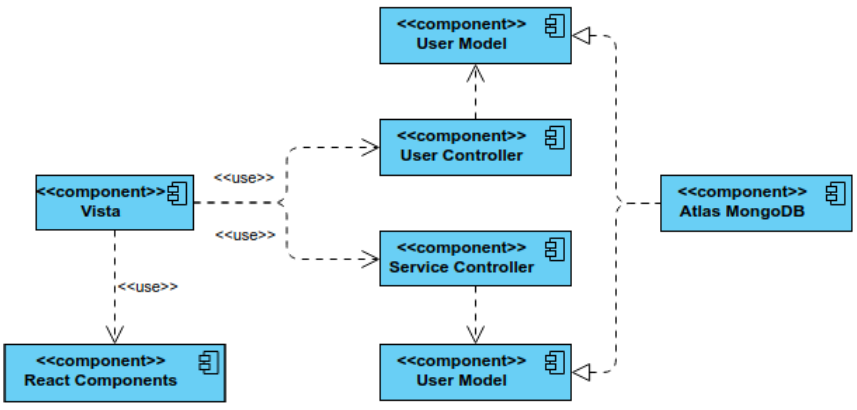


Este es el repositorio del cliente y el fichero index.html es la vista que utiliza el usuario para interactuar con el servidor.

El servidor contiene dos directorios con los modelos de los datos que requiere nuestra aplicación y los controladores que usan sus respectivos modelos para poder procesar las solicitudes de nuestro cliente.



A continuación pasamos a presentar el diagrama de componentes de nuestra aplicación:



El componente Vista es la interfaz gráfica con la que interactúa nuestro usuario. Dicha Vista usa los componentes de React para que estos sean renderizados en la Vista.

La vista se conecta al servidor y usa los componentes controladores para que estos puedan gestionar las peticiones de la Vista en función de lo que solicite el usuario.

Los controladores dependen de los Modelos que permiten conectarse a la Base de Datos y obtener toda la información que el cliente solicite.

3. Metodologías de desarrollo

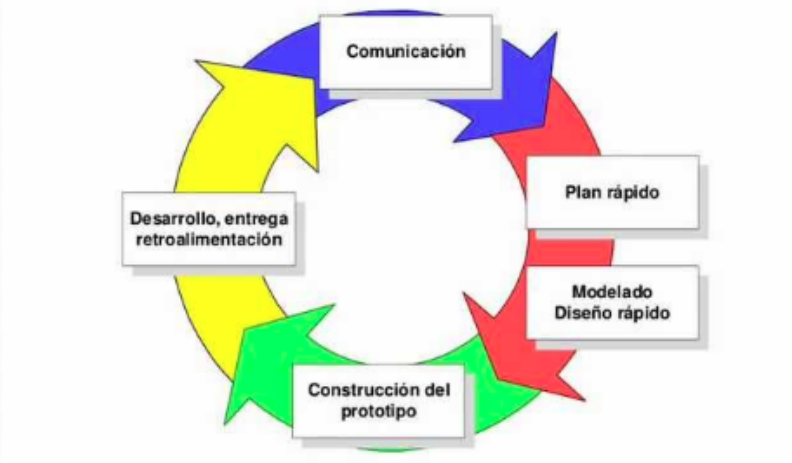
Para el desarrollo de la aplicación se han utilizado dos metodologías ágiles muy útiles para aplicaciones como la nuestra:

Metodología GitFlow:

Para el desarrollo se han ido creando ramas en el repositorio, tanto del backend como en el frontend, en función de las tareas que iban surgiendo. Todas estas ramas han terminado convergiendo en una rama de apoyo llamada “Develop”, para que esta se haya podido mergear correctamente en la main branch llamada “Master”.

Esto permite estructurar bien cada implementación que se realice en el proyecto, evita la pérdida de commits y que surjan conflictos a la hora de mergear ramas.

Metodología de desarrollo prototipado:

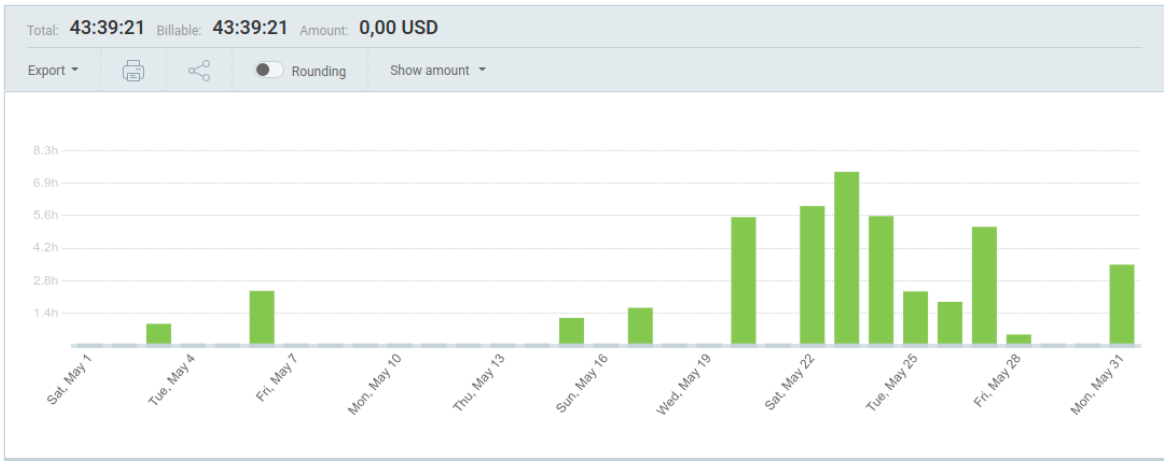


Esta metodología es muy versátil perfecta para aplicaciones donde no se tiene muy claro qué tipo de datos van a entrar y que tipo de datos van a salir.

Consiste en ir desarrollando distintos prototipos de nuestra aplicación para ir añadiendo distintas funcionalidades. Las funcionalidades se van acumulando hasta obtener el prototipo de nuestra aplicación más completo. Dicha metodología es muy utilizada en el desarrollo de aplicaciones web. Cada prototipo de nuestra aplicación contaba con una funcionalidad más que la anterior.

4. Desarrollo de la aplicación

Aquí tenemos todas las horas invertidas en el mes de mayo:



Gran parte de todas las horas invertidas durante el mes de mayo consisten en el estudio y análisis de las tecnologías que se han usado para el desarrollo de la aplicación.

Y aquí tenemos las del mes de junio:

Durante el mes de junio se ha ido poniendo en práctica todos los conocimientos que se iban asumiendo durante el mes pasado, y por lo tanto, el desarrollo de la aplicación creció exponencialmente durante este último mes.

Aquí tenemos todas las herramientas de software y hardware que se han utilizado para el desarrollo de la aplicación:

A. Análisis
B. Diseño
C. Implementación
D. Redacción de manuales

Nombre	Descripción	A	B	C	D
Visual Studio Code	IDE de desarrollo software			X	
Visual Paradigm	Herramienta de modelado de UML		X		X
Google Drive Docs	Herramienta para redactar documentación	X			X
Ubuntu 18.04	Sistema Operativo	X	X	X	X
Portatil	Portatil Asus	X	X	X	X

Teniendo en cuenta todas las horas invertidas y el coste de las herramientas usadas esté es el presupuesto:

Coste por horas invertidas como analista junior: 27 horas · 25 € = 675 €
Coste por horas invertidas como programador: 80 horas · 15 € = 1200 €
Coste por herramientas usadas (ninguna de pago): 0 €

Coste total del proyecto = 1975 €

5. Conclusiones

Posibles mejoras:

- Ya se ha comentado durante la documentación, pero la versión final de este proyecto pretende conectarse con el Blockchain de Flow para poder gestionar contratos virtuales entre los usuarios de la aplicación.
- La paleta de colores no es la adecuada para una aplicación dedicada a ofrecer oportunidades de trabajo.
- Al no conocer completamente React, quizá la implementación de componentes se podría refactorizar.

Las principales dificultades encontradas:

- Adaptarse a tecnologías que no había usado antes.