Proceso de desarrollo

**Introducción**

En este documento quedan documentadas las decisiones plasmadas para el desarrollo de la prueba backend de PHP.

**Concepto**

Se decidió iniciar 2 carpetas de proyecto debido a que su naturaleza es en funcionamiento es diferente: Api donde se pondrá toda la lógica del backend y las peticiones que tengan una solicitud que requiera de comunicación con la base de datos y la parte de Frontend que se encarga de procesar únicamente de la vista.

**Lenguaje**

Para el backend, se escoge Php 8 y para Frontend Html y JS con la librería JQuery.

**Objetivo**

Dados los requisitos presentados en la prueba, se pone como objetivo el desarrollo de un software que permita a un usuario registrarse con su número de teléfono e identificación que será corroborado, y allí empiece con un sistema de registro propio de sus usuarios (que no será accesible desde otro usuario). Se identificarán algunas bases de datos para la extracción de datos a partir de documentos de identidad.

**Planteamiento**

Decidí empezar el desarrollo dividiendo el proyecto en módulos según su funcionalidad y en los modelos que lo componen:

1. Modulo Usuarios
   1. Usuario
   2. Registros asociados
   3. Token de ingreso
2. Módulo de Notificaciones
   1. Notificación
   2. Log
3. Módulo de Estadísticas
   1. Estadística

**Planeación**

Una vez identificado el objetivo y planteado los componentes de la aplicación, procedí a plantear el desarrollo en varias etapas del proceso con ASANA.

**Desarrollo**

En el desarrollo del ejercicio se empezó por crear 2 carpetas de trabajo diferentes para cada uno de los objetivos (backend y frontend) y serán documentados individualmente.

1. Backend:
   1. Se incorpora una librería de creación de rutas para las diversas peticiones y que puedan funcionar con un .htaccess y así dedicarnos a tener una estructura adecuada y un mejor orden para cada petición. Dentro de la búsqueda en librería encontré “nikic/fast-route”
   2. La revisión de la documentación de la librería “catfan/medoo” toma una evaluación fuerte al poder esta configurarse para hacer modificación de modelos generales un poco más rápido
   3. Con la ruta y el sistema de conexión con la base de datos, se empezó a hacer la conexión general de lo que iba a necesitar: Modelo Base, un Controlador Base y un Boot Base. Se decide entonces crear unas clases que me ayudaran a servir como Middleware para aquella solución a peticiones repetitivas como la del usuario actual.
   4. Una librería de gestión de comandos seria de mucha utilidad para el procesamiento de la instalación del sistema, asi decidí importar “splitbrain/php-cli” y enfocarme en desarrollar 2 comandos, el de alterar la configuración del sistema y el de migrar la base de datos.
   5. Decidí hacer un test sobre los sms para probar su funcionamiento al enviarse y hacer test de conexión con el API de Cellvoz.
   6. Una vez listas las conexiones se retorna el token de seguridad usando JWT y guardando su vencimiento en la base de datos.
   7. Se generó un modelo de logs que permite imprimir en archivos separados los logs de cada servicio o instrucción independiente.
2. Frontend:
   1. Debido a que no se puede usar frameworks, decidí instaurar el package.json del npm pero no instalar en él nada para no cargar el repositorio, todas las librerías serán cargadas por CDN.
   2. El enfoque principal se lo llevo el sistema de login, donde deje instaurado el sistema de notificaciones de CellVoz en función con el registro.
   3. Se hizo que cada una de las vistas se cargaran de manera independiente, permitiendo así tener un mejor control de los scripts en cada una de los fragmentos necesarios.

**Consideraciones**

1. Se hizo una evaluación rápida de las librearías de uso para no generar mucha carga en los archivos generales
2. Hubo un problema de conexión con los datos iniciales por lo cual se decidió enfocar el proyecto a algo similar con los mismos objetivos de la prueba, pero con datos mas reales.
3. No ha sido posible probar hasta el momento fuera de la maquina local.