

#### **ALIANZA FIDUCIARIA Y VALORES**

## PRUEBA PRÁCTICA INGENIERO FULLSTACK

Nombre:	
Cargo al que aspira:	

**Duración:** 4 – 5 horas

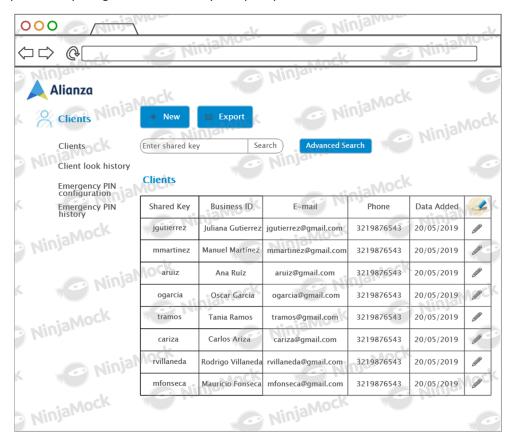
Utilizando el IDE con el cual se sienta más cómodo, desarrollar los ejercicios que se numeran a continuación.

Tenga en cuenta que será evaluado el diseño de la solución y las prácticas de codificación, no solamente el software funcionando. El tiempo también será un factor evaluado vs la completitud de la solución.

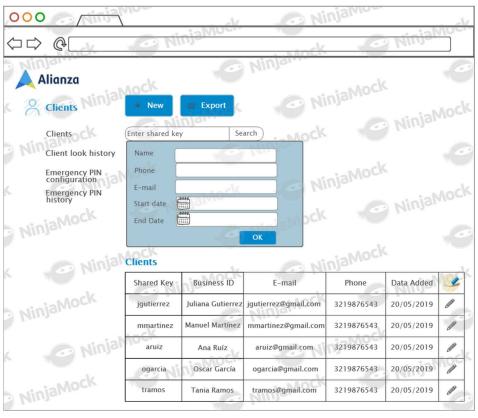
La entrega se hará por medio de un repositorio GIT de su elección (por ejemplo: github), el único requisito es que nos envíe la URL y que no esté protegido para que lo podamos clonar.

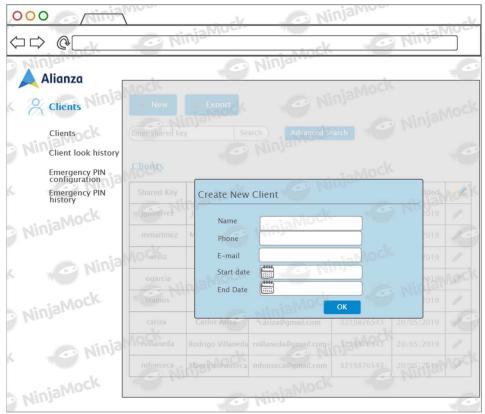
#### 1. Pantalla de administración de clientes

Crear 2 pantallas que sigan el look & feel y campos que se muestran a continuación:











Estas pantallas simulan una aplicación con un menú lateral y un espacio de trabajo a la derecha, la idea es implementar el link para la opción "Clientes" (las demás opciones no se deben implementar) de forma que permita listar los clientes, buscar los clientes (búsqueda simple por shared key, que es un campo texto) y crear un nuevo cliente.

Se deben agregar las validaciones de pantalla que considere necesarias para los tipos de campos.

Se debe hacer un servicio rest que permita consultar, buscar por shared key y crear un cliente, la implementación no necesita llevar base de datos, esto con el fin de hacer más sencillo el ejercicio (pero si le es más practico usar base de datos, lo puede hacer). Las pantallas deben interactuar con este servicio rest para cargar y enviar la información.

# Requerimientos no funcionales

- En todos los pasos se debe implementar un sistema de logs, de forma que cuando se coloque en producción ayude a dar soporte del sistema.
- Generar pruebas unitarias para los casos que considere necesarios. Puede usan un framework como junit, testNG, si no conoce ninguno puede usar clases planas en java desde un main. Para el caso del front también implementar pruebas unitarias.
- Para la capa de presentación se debe usar Angular.
- Las pantallas deben ser responsive.
- El servicio debe ser implementado en Java, se puede cualquier framework.

### 2. Puntos opcionales

NOTA: Este punto es opcional, no es necesario hacerlo, pero proporciona puntos extras.

- Implementar la funcionalidad del buscador avanzado (botón advanced search), tanto en la parte front como en el servicio.
- Implementar la opción de Exportar (ver botón en la primera pantalla) hacia un archivo csv.