



## PRUEBA TECNICA

### Objetivo

El objetivo de esta prueba es evaluar al candidato y como resuelve un problema o caso de uso habitual de un proyecto.

### Problema

Implementar un front-end con dos pantallas: una para registro de usuarios y otra para visualizar un histórico de operaciones sobre esos usuarios.

En la primera pantalla se requiere que se solicite la siguiente información:

| Campo  | Tipo    | Formato   | Requerido |
|--|---------|---|-----------|
| Nombre   | String  | Sólo letras y espacios  | Si        |
| Apellido   | String  | Sólo letras y espacios  | Si        |
| Correo Electrónico   | String  | Correo Válido   | Si        |
| Fecha de Nacimiento  | Date    | YYYY-MM-DD  | Si        |
| Teléfono   | Number  | Sólo números  | No        |
| País de Residencia   | String  | Debe ser un dropdown con opciones de países. El país se debe guardar con su código internacional. Por ejemplo, Costa Rica como CRI (según ISO 3166-1) | Si        |
| Pregunta Sobre Contacto<br>(¿Desea recibir información? Si/No) | Boolean | 1 / True = Si, 0 / False = No   | Si        |

No se va a evaluar diseño, se evalúa funcionalidad. En cuanto a tecnología, se puede realizar la solución usando .Net, React, Angular o Flutter. No se aceptarán proyectos en PHP.

Este front-end debe conectarse a un back-end que exponga servicios RESTful para las operaciones CRUD (Create, Read, Update, Delete) para la entidad de usuarios. Este back-end se puede trabajar en .Net o en NodeJS. No se aceptarán proyectos en PHP.

El back-end se tiene que conectar a una base de datos para realizar las operaciones.

Email. [info@loymark.com](mailto:info@loymark.com)

Tel. (+506) 2521 - 6222

[www.loymark.com](http://www.loymark.com)

Diagonal a la Cámara de Comercio Barrio

Tournón, Edificio Garnier BBDO

San José, Costa Rica.



Esta base de datos va a constar de 2 tablas:

- usuarios
- actividades

Para la tabla de usuarios se tienen que generar los campos especificados arriba en el enunciado.

La tabla de actividades va a mantener un registro por cada acción realizada por el back-end. Por ejemplo: el usuario A, se creó la fecha tal. Este registro de actividades se tiene que ejecutar en cada llamado al CRUD de usuarios.

Como estructura, la tabla de actividades tiene los siguientes campos:

| Campo        | Tipo   | Formato   | Requerido |
|--------------|--------|---|-----------|
| id_actividad | Number | Autogenerado  | Si        |
| create_date  | Date   | Fecha auto asignada   | Si        |
| Id_usuario   | Number | Relación al registro del usuario con el que se está interactuando.                                    | Si        |
| actividad    | String | Observación que detalla la actividad, por ejemplo: "Creación de Usuario" o "Actualización de Usuario" | Si        |

Para el front-end de la actividad de usuarios, se requiere que se presente una lista ordenada de más reciente a más antigua (usando el campo create\_date para ese ordenamiento).

En la lista se requiere que se vean los siguientes campos:

| Columna              |
|----------------------|
| Fecha de Actividad   |
| Nombre Completo      |
| Detalle de Actividad |

Reiteramos, no se está evaluando diseño si no funcionalidad.



Para poder acceder a cada página (usuarios o actividades) se requiere que se presente un menú tipo NavBar para solución web o bien barra de botones en parte baja o alta de pantalla para soluciones Flutter Móviles.

### **Presentación y Entrega de Solución**

Se requiere que el código de la solución sea presentado en un repositorio de Github. Dentro del repositorio se esperan 3 folders:

- front-end
- back-end
- database (\*)

(\*) Para la base de datos se requiere que se entreguen los scripts para creación de la base de datos, creación de las tablas y si fuera necesario, script para creación de usuarios.

Una vez entregado el repositorio, se coordinará una llamada para que el candidato explique brevemente la solución.