

PRUEBA TECNICA DESARROLLADOR FULLSTACK - PHP

La empresa de prestamos **UThis** quiere ofrecer una renovación de crédito a sus clientes actuales, para lo cual requiere de un desarrollo que contemple lo siguiente:

Al ingresar a la página principal se muestre un inputbox que solicite la captura del RFC del cliente, y un botón Continuar que al dar click consumirá un API, en la que si el resultado es exitoso permitirá continuar al cliente a una siguiente pantalla donde mostrara los datos que el API arroje al cliente, estos datos deberán de guardarse en tablas.

Ejemplo de response del api de búsqueda:

```
{
  "success": true,
  "datos": [
    {
      "datos_personales": {
        "cliente_id": 100234,
        "nombre": "HECTOR",
        "apellido_paterno": "URBANO",
        "apellido_materno": "MAZA",
        "rfc": "UAMH880216",
        "fecha_nacimiento": "1988-02-16",
        "ingresos": 25890,
        "egresos": 12680,
        "no_dependientes": 0,
        "estado_civil": "SOLTERO",
        "genero": "MASCULINO",
        "ultimo_grado_estudios": "LICENCIATURA"
      },
      "datos_domicilio": {
        "calle": "ALVARO OBREGON",
        "no_exterior": "138",
        "no_interior": "97",
        "colonia": "SAN FRANCISCO OCOTLAN",
        "municipio": "CORONANGO",
        "estado": "PUEBLA",
        "cp": "72680"
      },
      "datos_credito": {
        "monto": 35000,
        "plazo": "18 MESES",
        "pago_mensual": 2689.77,
        "tasa_interes": "40 %"
      }
    }
  ]
}
```

Para la parte del domicilio, se deberá contemplar una funcionalidad donde mediante un checkbox/slider donde se pregunte si el domicilio precargado es el actual (por default deberá estar en Si/True), si el cliente cambia a No/False se deberá permitir la captura del nuevo domicilio en la que con la sola captura del CP se deben de precargar las colonias, estado y municipio, al final debe de contener un botón de continuar que deberá dirigir al cliente a la pantalla final que contendrá los datos de la oferta de renovación que se le asigna al cliente y contendrá dos botones que permita aceptar o rechazar la oferta.

Se quiere que el desarrollo exponga un API para poder consultar los clientes que se hayan registrado y el estatus en el que se encuentra contemplando al menos los siguientes:

Pre-Registro – Si el cliente consulta el RFC y solo se le muestran los datos precargados

Registro – Si el cliente avanza a la oferta, pero no la acepta/rechaza

Oferta Aceptada/Oferta Rechazada – Si el cliente acepta / rechaza la oferta

El Api deberá contener una respuesta en formato JSON al menos con los siguientes datos:

- Nombre completo del cliente
- RFC
- Id Cliente
- Domicilio (Calle, No Ext, No Int)
- Colonia
- CP
- Estado
- Actualizo domicilio (Si: El cliente modifico el domicilio que el servicio regresa. No: El cliente conserva el domicilio que el servicio regresa)
- Monto del crédito
- Plazo
- Pago mensual
- Tasa

Puntos para tomar en cuenta:

- El proyecto se deberá subir a github/bitbucket de manera pública.
- Se comparte una colección que contiene los endpoints para realizar las consultas a los servicios de clientes y códigos postales y el de autenticación para el consumo de estos dos servicios.
- Todos los campos de captura deben de contemplar validaciones tanto en frontend como en backend (principalmente).
- Deberán de existir 3 tablas: datos_cliente, datos_domicilio, datos_prestamo. La tabla datos_cliente sería la tabla central, el estatus de la oferta se deberá guardar en esta tabla, deberán de contar con sus migrates.
- Si el API de consulta del RFC arroja que el cliente no existe deberá de mostrarse un modal con este mensaje.
- Si el cliente ya se registró y se encuentra como Oferta Aceptada/Oferta Rechazada se deberá mostrar un mensaje con la leyenda que cuenta con una Oferta Aceptada/Rechazada según sea el caso y no puede volver a registrarse hasta 1 mes después de la fecha de aceptación/rechazo (se deberá indicar en la leyenda esta fecha). Si se encuentra en Pre-Registro, deberá mostrar de nuevo los datos precargados y si se encuentra en Registro deberá mostrar la oferta.
- El Api para consulta de registros deberá de contener un mecanismo de autenticación ya sea con AUTH o JWT y deberá contener la opción para consultar a una sola persona mediante RFC o a varias mediante un rango de fechas de registro. Los parámetros de entrada deberán estar validados y el API deberá tener la capacidad de arrojar mensajes de error si consulto RFC no registrados, fechas mal capturadas, RFC inválidos, etc.

- Se deberá contar con una documentación que indique los pasos a seguir para poder realizar la instalación adecuada del proyecto en local y poder realizar las pruebas pertinentes.
- Deberá contar al menos con pruebas unitarias.
- El desarrollo se deberá realizar con la versión 6.x o 7.x de Laravel y para la base de datos se deberá utilizar mysql (considerar que el servidor donde se ejecutará la prueba cuenta con la versión 5.7)
- Para el front end pueden ocupar VueJs o JQuery.
- Plus si el desarrollo se entrega en un servidor para pruebas, la mayoría de los servicios en la nube cuentan con una capa gratuita (Cloud9, Heroku, Digital Ocean, AWS, GCP)