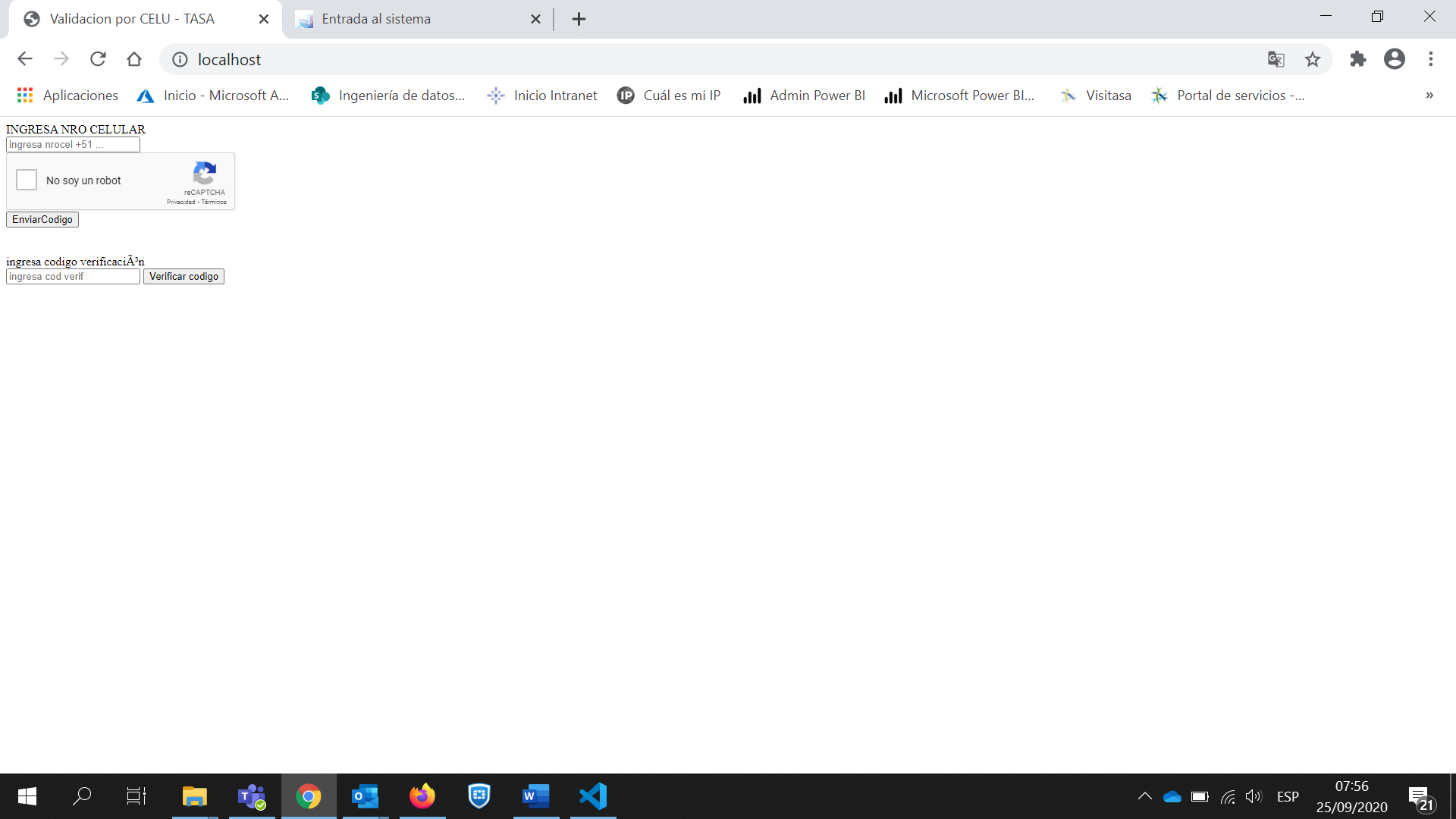
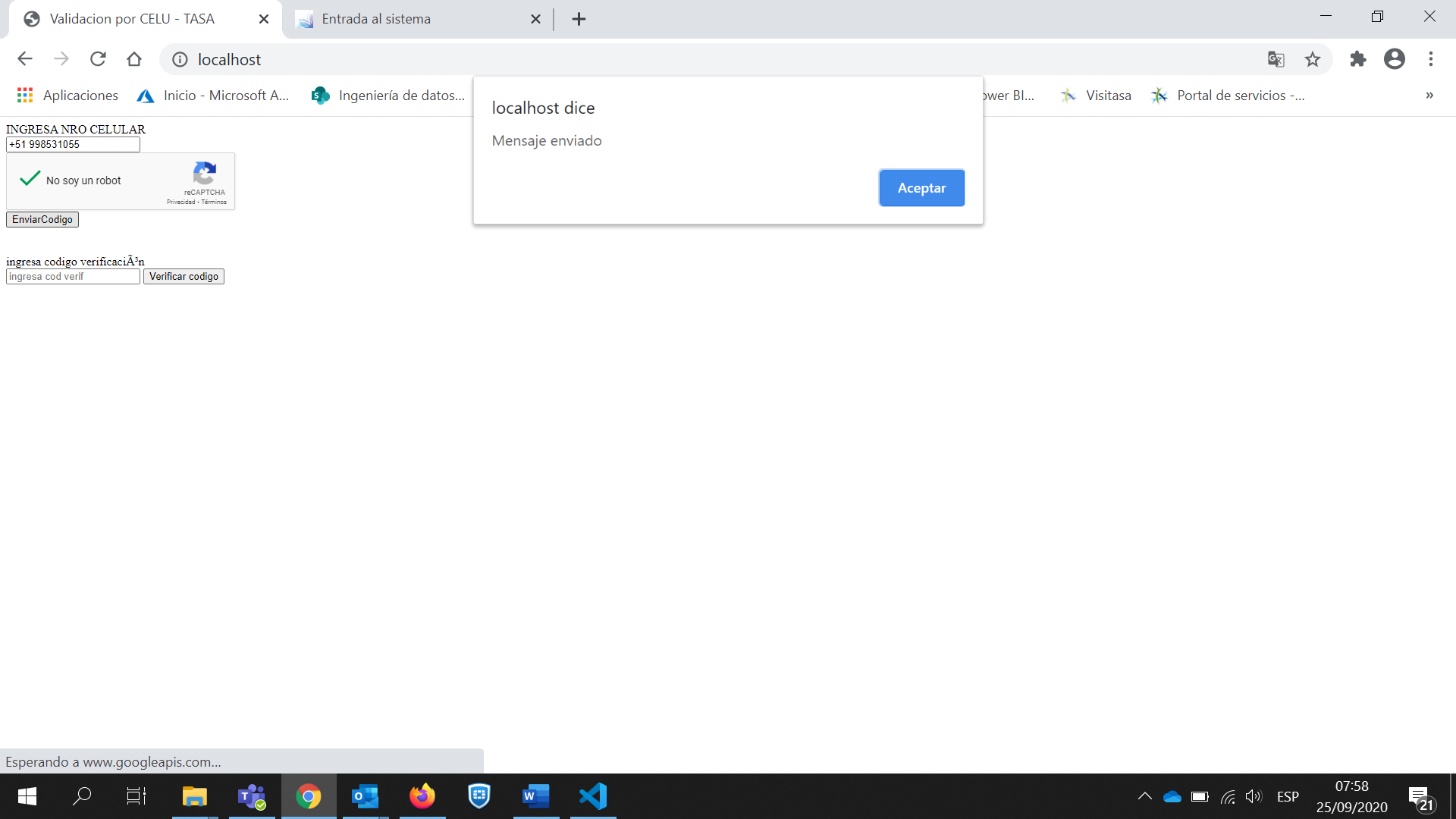
**Validación y registro de usuarios a través de Firebase**

**I.- FRONT END. Interfaz donde interactúa el usuario.**

El usuario ingresará su número de celular en el primer campo solicitado, marcará con check el captcha para validar que se trate de un usuario auténtico. Luego recibirá en unos segundos un SMS proporcionándole la clave de acceso, el cual ingresará en el segundo campo solicitado para verificar que sea correcta y así pueda registrar su acceso.

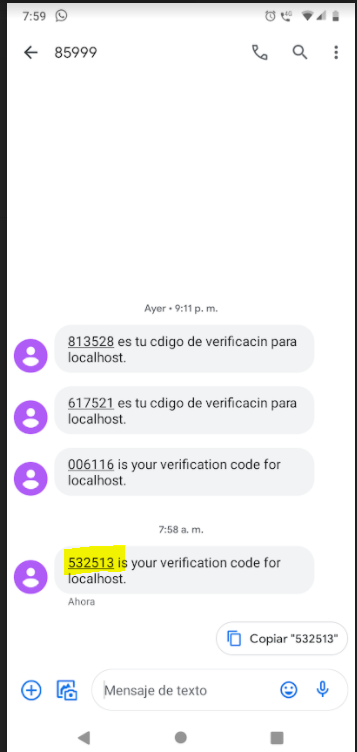


1. El usuario ingresa el número de seguridad +51 999999999, le da check al captcha (en algunos casos pedirá seleccionar imágenes específicas) y se presiona el botón EnviarCodigo.

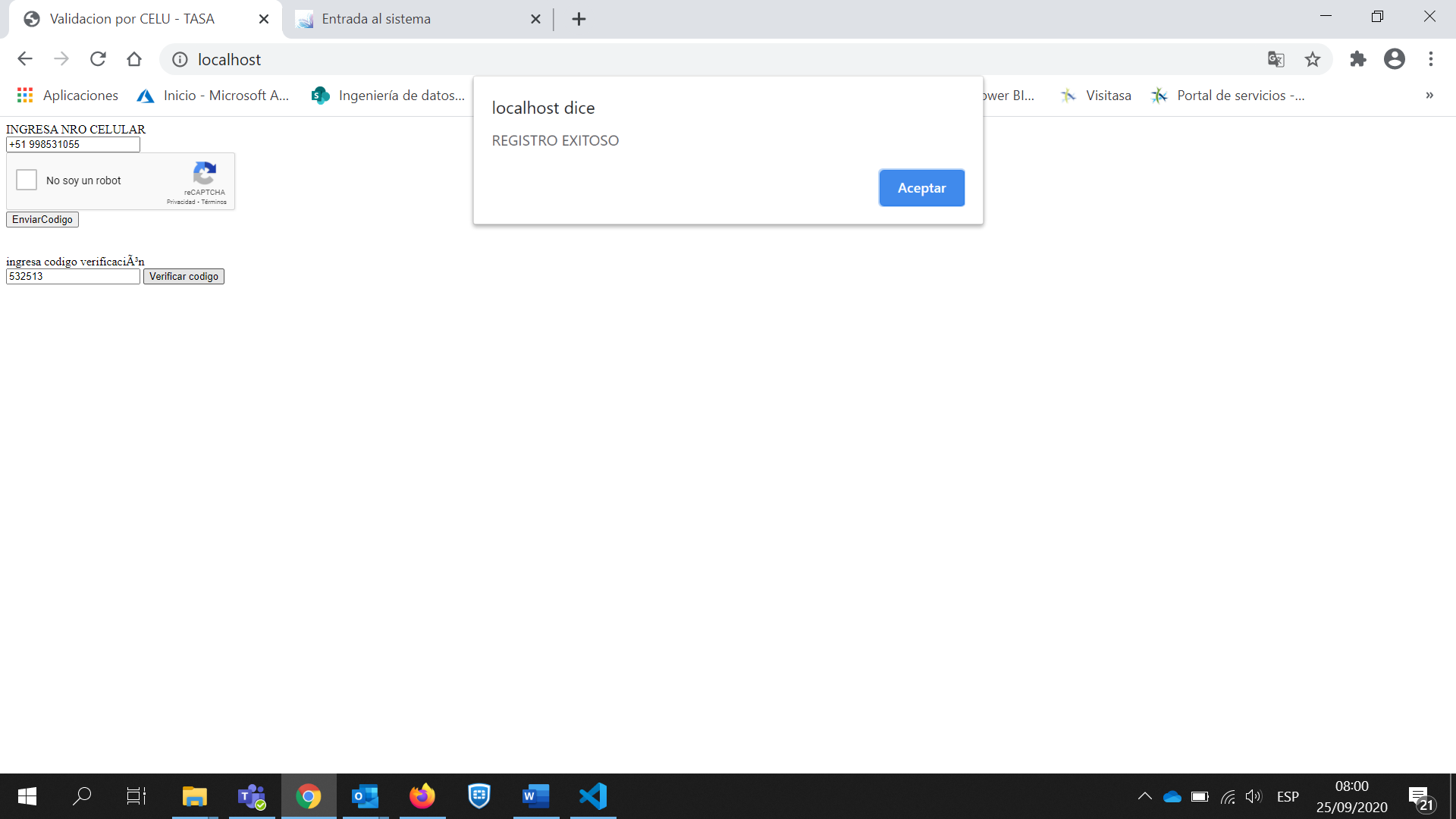


El botón Enviar Código ejecutará la función **phoneAuth( )** (Ver sección II Codigo fuentes) el cual recibe como parámetro el numero de celular capturado. Internamente genera el código de verificación y lo devuelve por medio de un SMS enviado al celular ingresado. El aplicativo adicionalmente lo notifica con un mensaje de “Mensaje enviado”

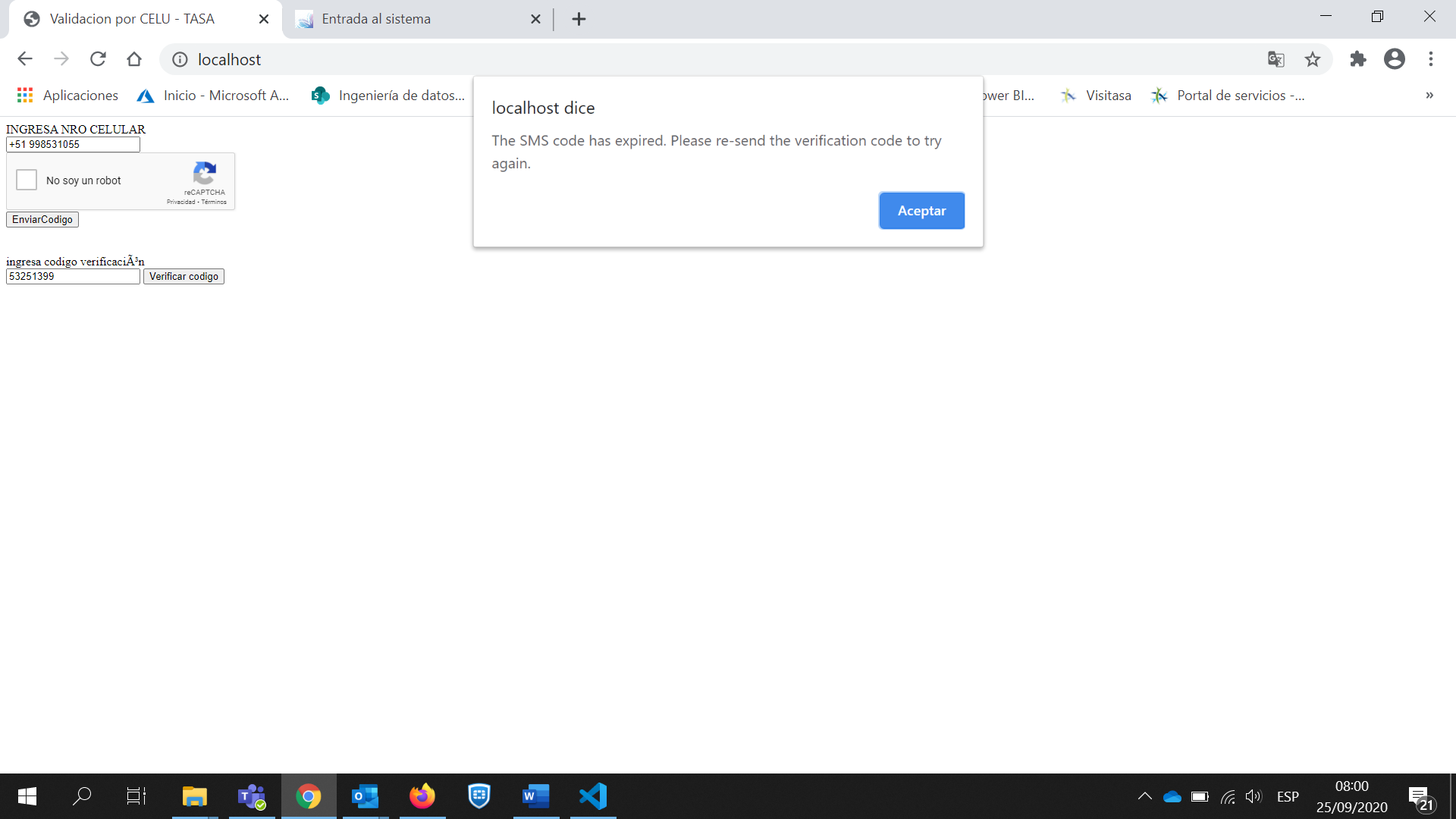
1. Mensaje SMS con el código de validación



1. El usuario ingresa el código recibido, en este ejemplo 532513, en el campo donde pide ingresar código de verificación y presionar botón “Verificar Codigo”. El botón ejecutará la función **codeverify ( )** el cual recibe como parámetro el código de verificación y realiza la validación por medio de sus librerias internas (se detallará en la sección II Codigo Fuentes). En caso el código ingresado sea válido, la aplicación web lo notifica con un mensaje de confirmación y se registra en la base de datos de Firebase.



En caso se ingrese un código incorrecto ó haya pasado un tiempo considerable, se muestra un mensaje de advertencia.



**II.- BACK END. Código fuente**

Son dos archivos de prueba: **iistart.htm,** donde está la interfaz de ingreso de datos, y **acceso\_sms.js,** en el cual están las librerías en javascript para ejecutar las funciones.

1. **Archivo iistart.htm**

La inclusión del archivo acceso\_sms.js para invocar las funciones de los formularios

<script src="acceso\_sms.js"></script>

Comandos de Firebase SDK que se autogeneran al momento de crear un proyecto en le Firebase. Dentro de este proyecto se incluye un aplicativo web el cual se está mostrando en este ejemplo. (Ver sección III Proyecto en Firebase). De este código no hay que cambiar nada.

<!-- The core Firebase JS SDK is always required and must be listed first -->

<script src="https://www.gstatic.com/firebasejs/7.21.0/firebase-app.js"></script>

<script src="https://www.gstatic.com/firebasejs/7.21.0/firebase-auth.js"></script>

<!-- TODO: Add SDKs for Firebase products that you want to use

     https://firebase.google.com/docs/web/setup#available-libraries -->

<script src="https://www.gstatic.com/firebasejs/7.21.0/firebase-analytics.js"></script>

<!--<script src="/\_\_/firebase/6.2.0/firebase-firestore.js"></script>-->

<script>

      // Your web app's Firebase configuration

      // For Firebase JS SDK v7.20.0 and later, measurementId is optional

 var firebaseConfig = {

    apiKey: "AIzaSyC5UYlZv66TnOfTIkCe0ixJyB3aGrZqt4E",

    authDomain: "t4s4sms.firebaseapp.com",

    databaseURL: "https://t4s4sms.firebaseio.com",

    projectId: "t4s4sms",

    storageBucket: "t4s4sms.appspot.com",

    messagingSenderId: "344655128766",

    appId: "1:344655128766:web:c24560eca794399100d6ef",

    measurementId: "G-5B3ZP2YHQK"

  };

      // Initialize Firebase

      firebase.initializeApp(firebaseConfig);

      //firebase.analytics();

</script>

Ingreso de nro celular. El Id es “number” e incluye el botón “Enviar Código” el cual invoca la función phoneAuth ( )

INGRESA NRO CELULAR

<br>

<form>

  <input type="text" id="number" placeholder="ingresa nrocel +51 ..."/>

  <div id="recaptcha-container"></div>

  <button type="button" onclick="phoneAuth();"> EnviarCodigo </button>

</form>

Ingreso del código de verificación. Id = verificationcode. El botón Verificar código ejecuta la función **codeverify ( )**

<form>

  ingresa codigo verificación

  <br>

    <input type="text" id="verificationcode" placeholder="ingresa cod verif">

    <button type="button" onclick="codeverify();">Verificar codigo</button>

</form>

1. **Archivo acceso\_sms.js**

Generación de Captcha para validar la presencia de un usuario.

window.onload=function()

{

    render();

}

function render()

{

    window.recaptchaVerifier  = new firebase.auth.RecaptchaVerifier('recaptcha-container');

    recaptchaVerifier.render();

}

**phoneAuth( )**: Función que recibe dos parámetros: “number” que es el número telefónico ingresado en el primer formulario; y la confirmación del Captcha. Internamente el comando firebase.auth().signInWithPhoneNumber se encarga de generar la clave de autenticación para luego enviarlo por SMS

function phoneAuth()

      {

          // captura nrocel

          var number = document.getElementById('number').value;

          var appVerifier = window.recaptchaVerifier;

          firebase.auth().signInWithPhoneNumber(number, appVerifier).then(function (confirmationResult) {

              window.confirmationResult = confirmationResult;

              coderesult = confirmationResult;

              console.log(coderesult);

              alert("Mensaje enviado");

          }).catch(function(error){

              alert(error.message);

          })

      }

**codeVerify( ):** Función que recibe como parámetro el código de seguridad ingresado en el segundo formulario. LA función confirmationResult.confirm(code) se encarga de validar dicho código y confirmar su registro.

function codeverify()

      {

          var code = document.getElementById('verificationcode').value;

          confirmationResult.confirm(code).then(function (result){

              alert("REGISTRO EXITOSO");

              var user = result.user;

              console.log(user);

          }).catch(function (error){

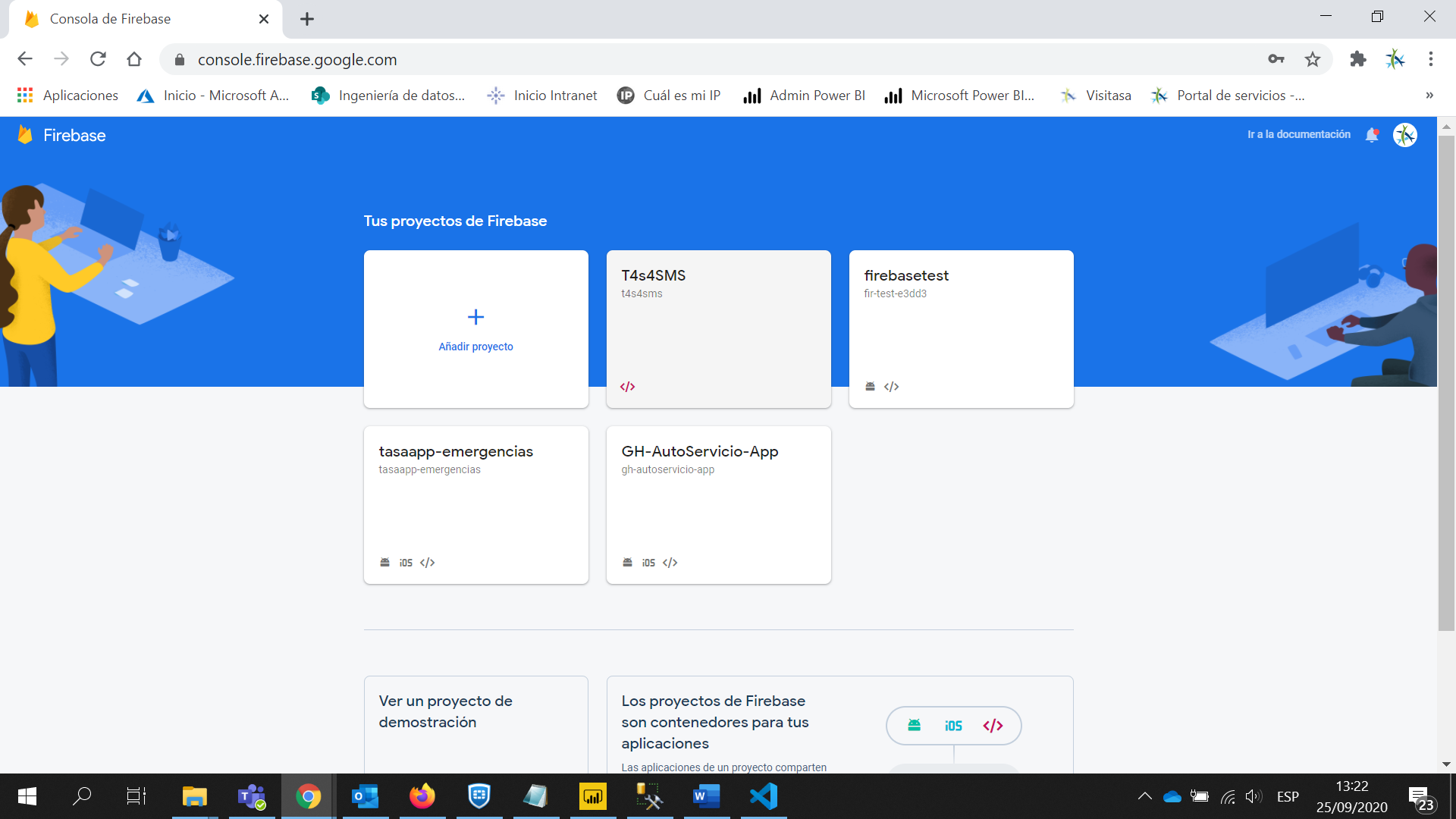
              alert(error.message);

          })

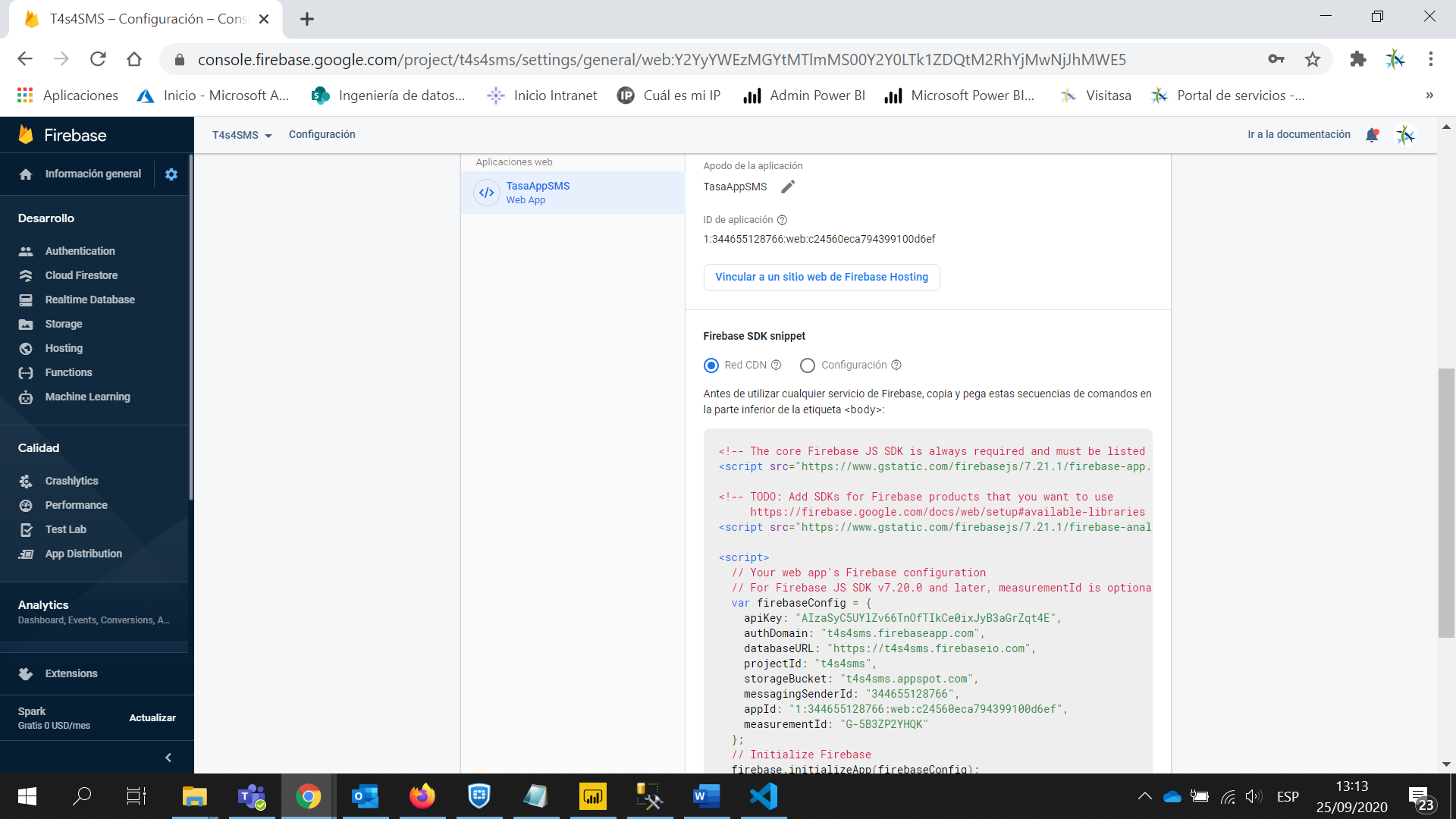
      }

**III. Proyecto en Firebase**

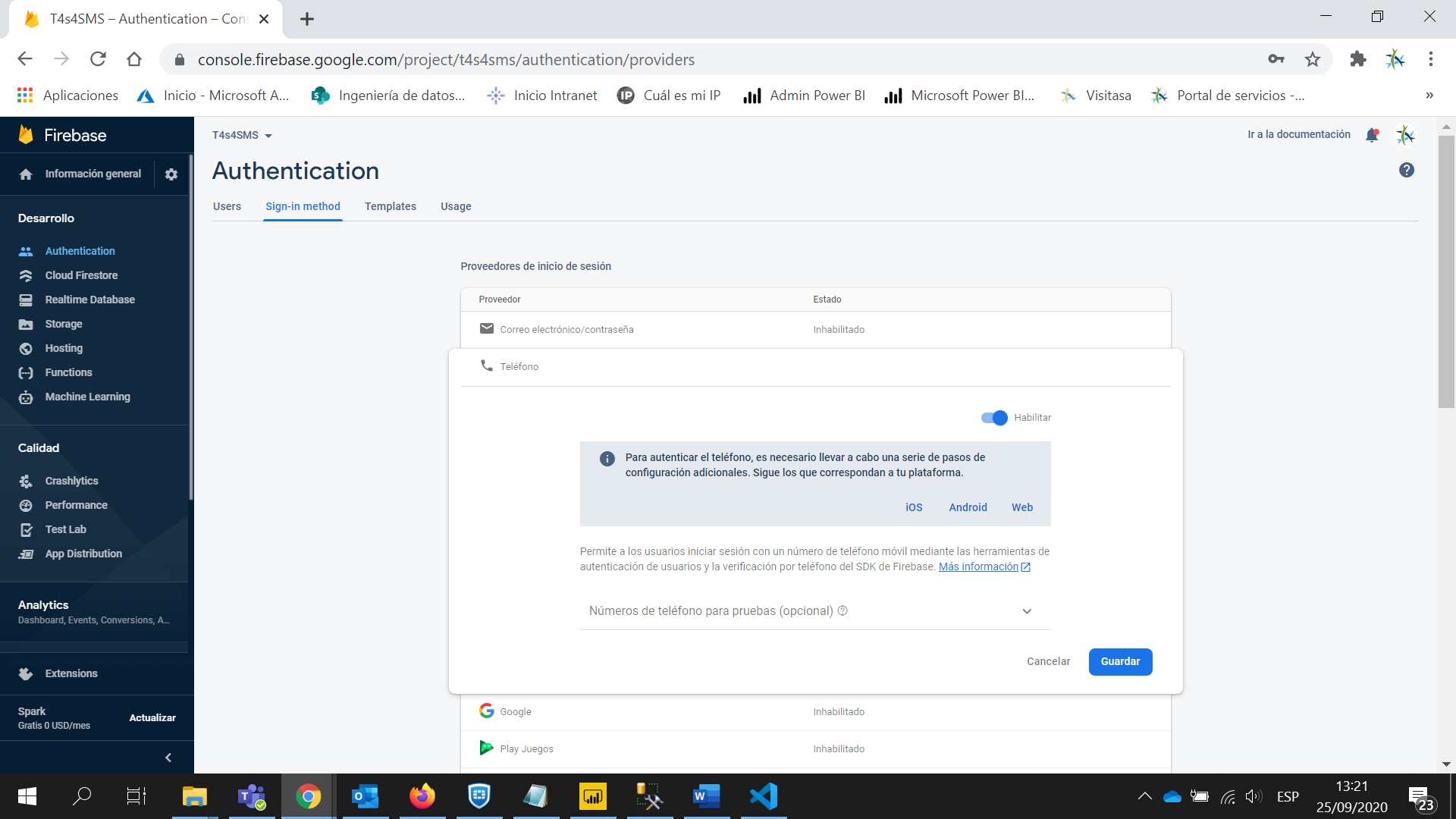
Para este caso se consideró el proyecto T4s4SMS, el cual contiene un aplicativo web llamado TasaAppSMS

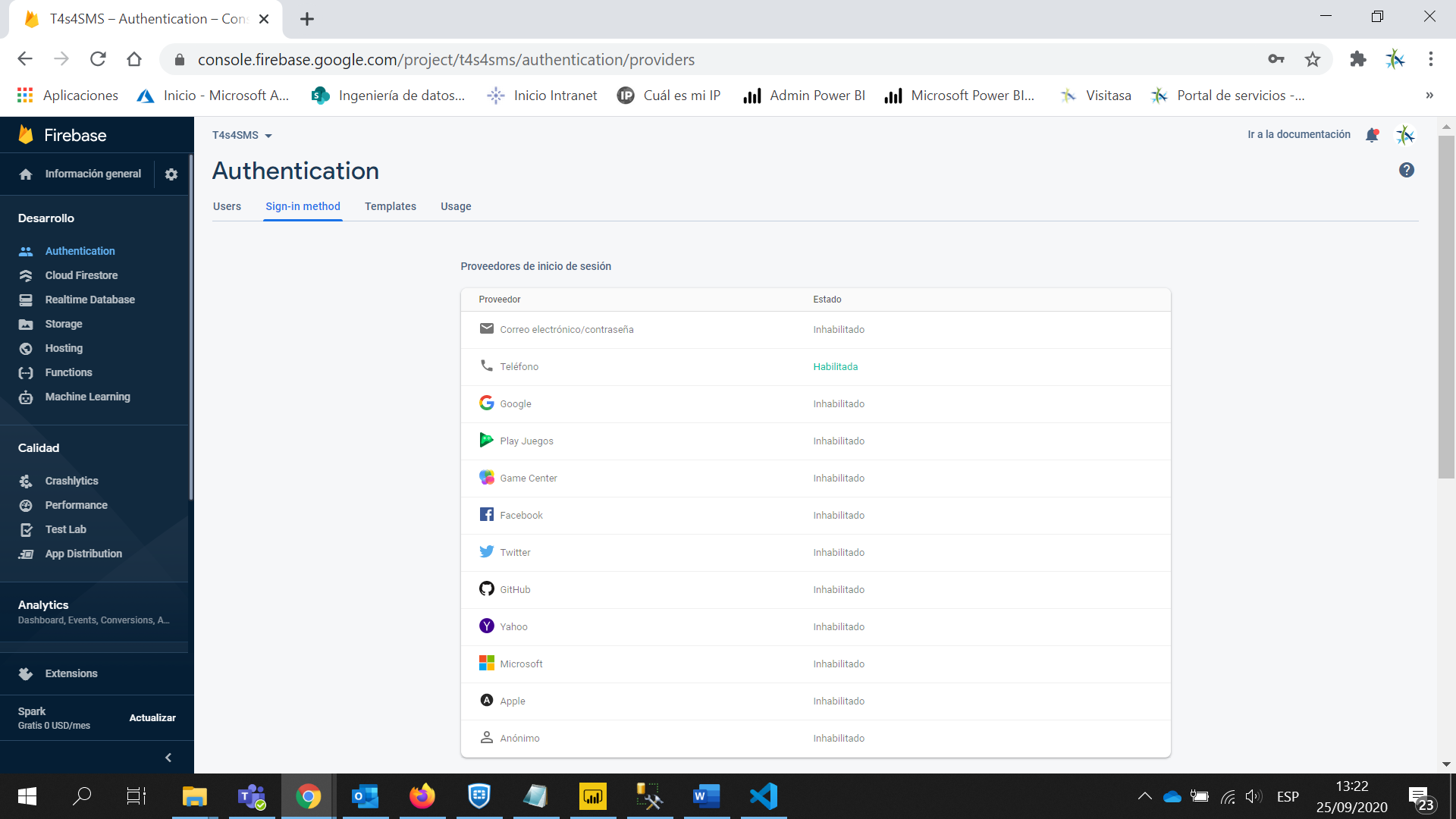


Aplicación TasaAppSMS: Este aplicativo es el que permitirá la autenticidad de accesos por medio de SMS. El Firebase SDK snippet que se muestra son las líneas de comandos autogenerados que deben utilizarse para conectar el aplicativo web con los servicios que incluyen el proyecto en Firebase (ver sección II Código Fuente). Por cada aplicativo que se crea en el proyecto se generan distintos SDK’s



Métodos de autenticación: En este caso se habilita el acceso telefónico





Autenticación de usuarios: Los números que se van registrando en la interfaz del aplicativo se registran en esta base de datos.

