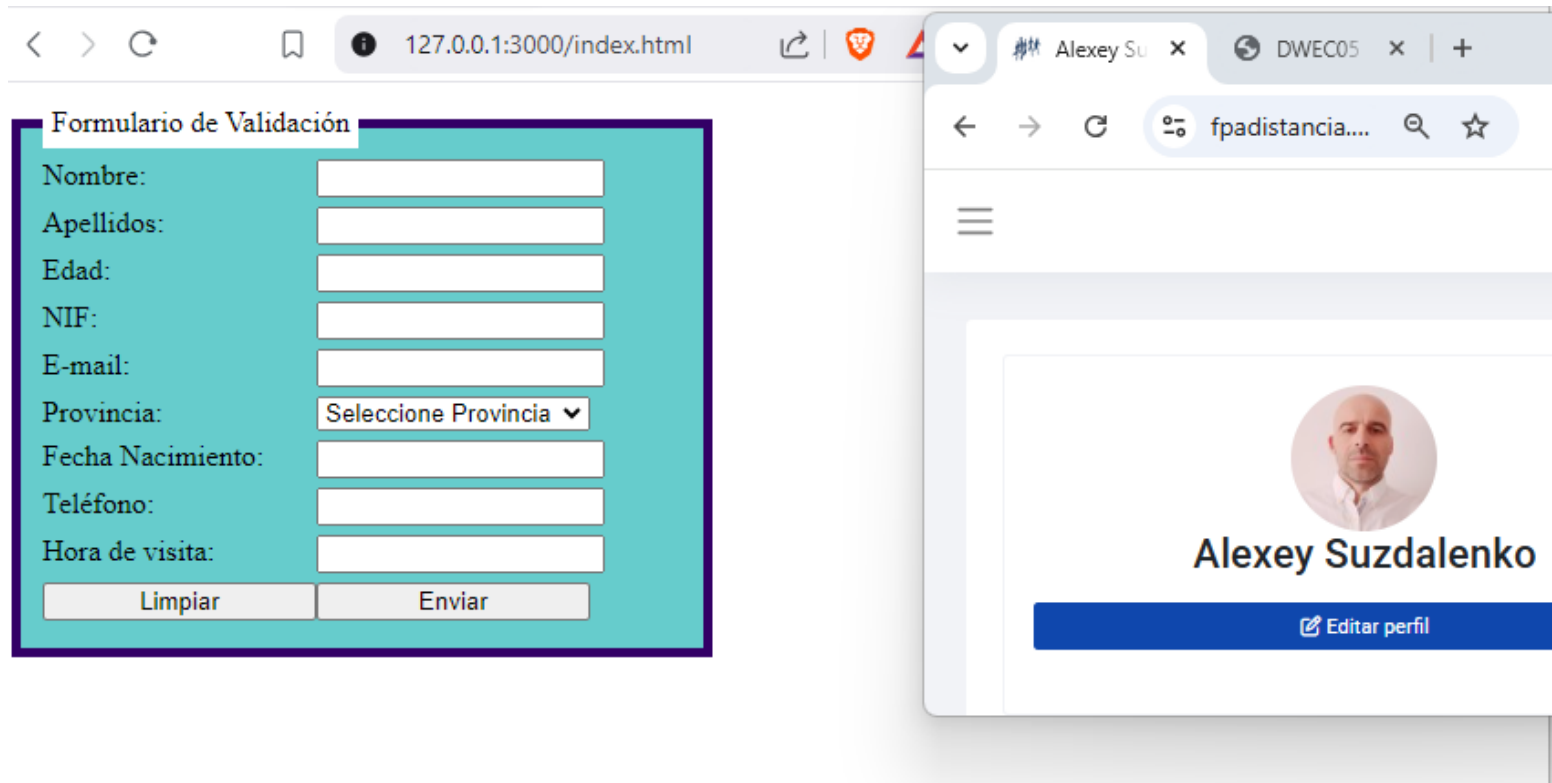
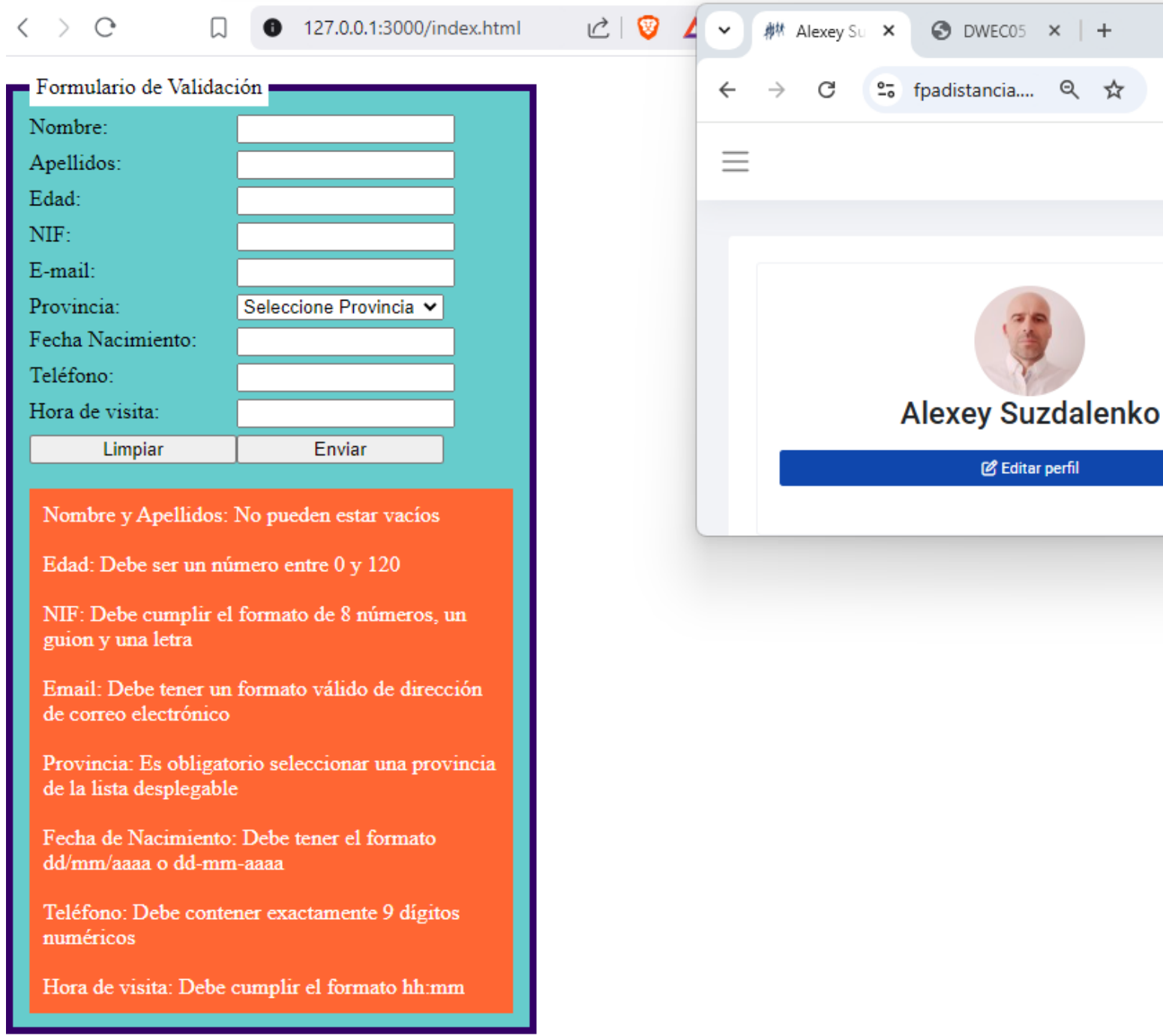


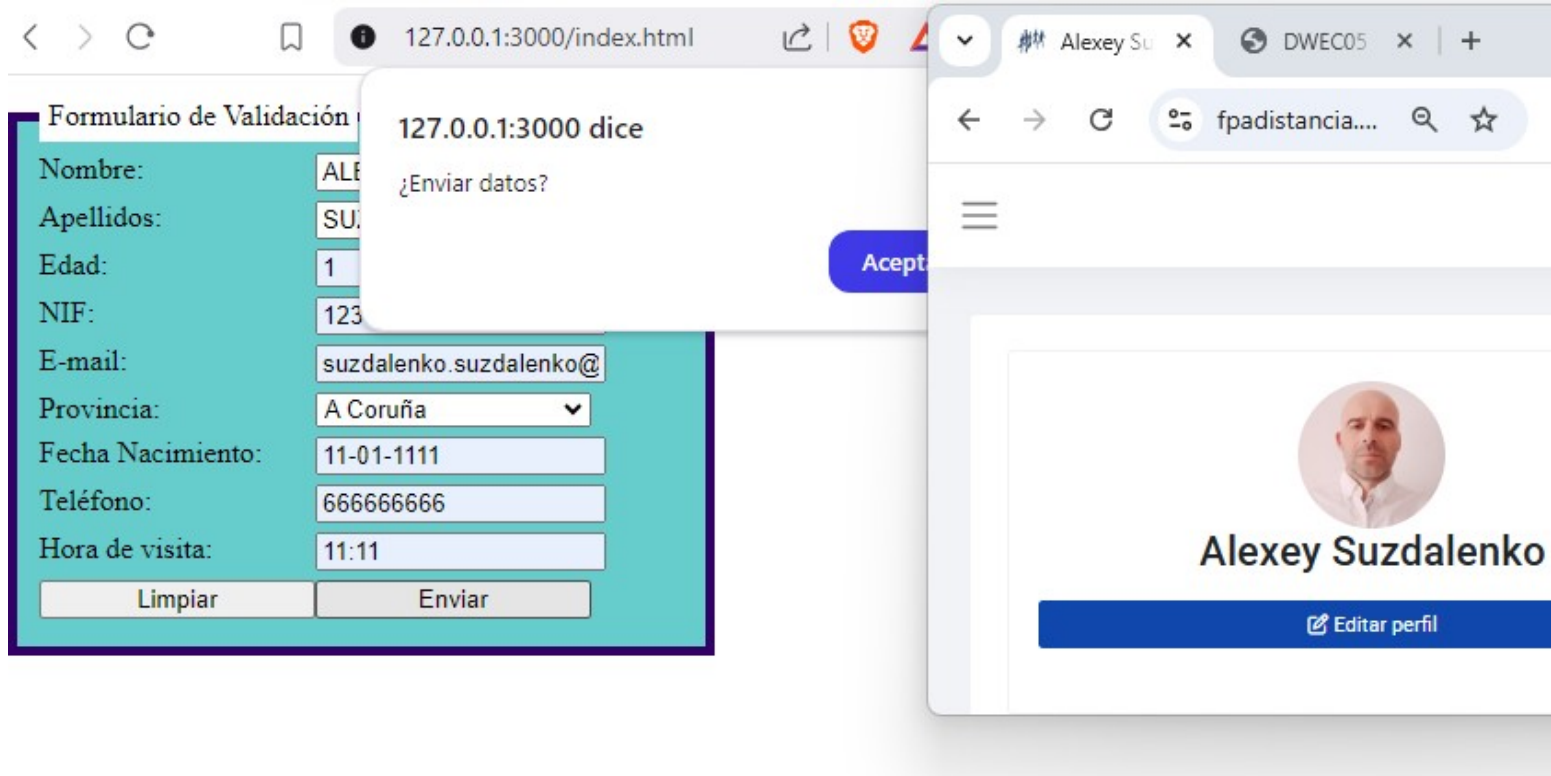
Captura de pantalla según enciendo la aplicación:



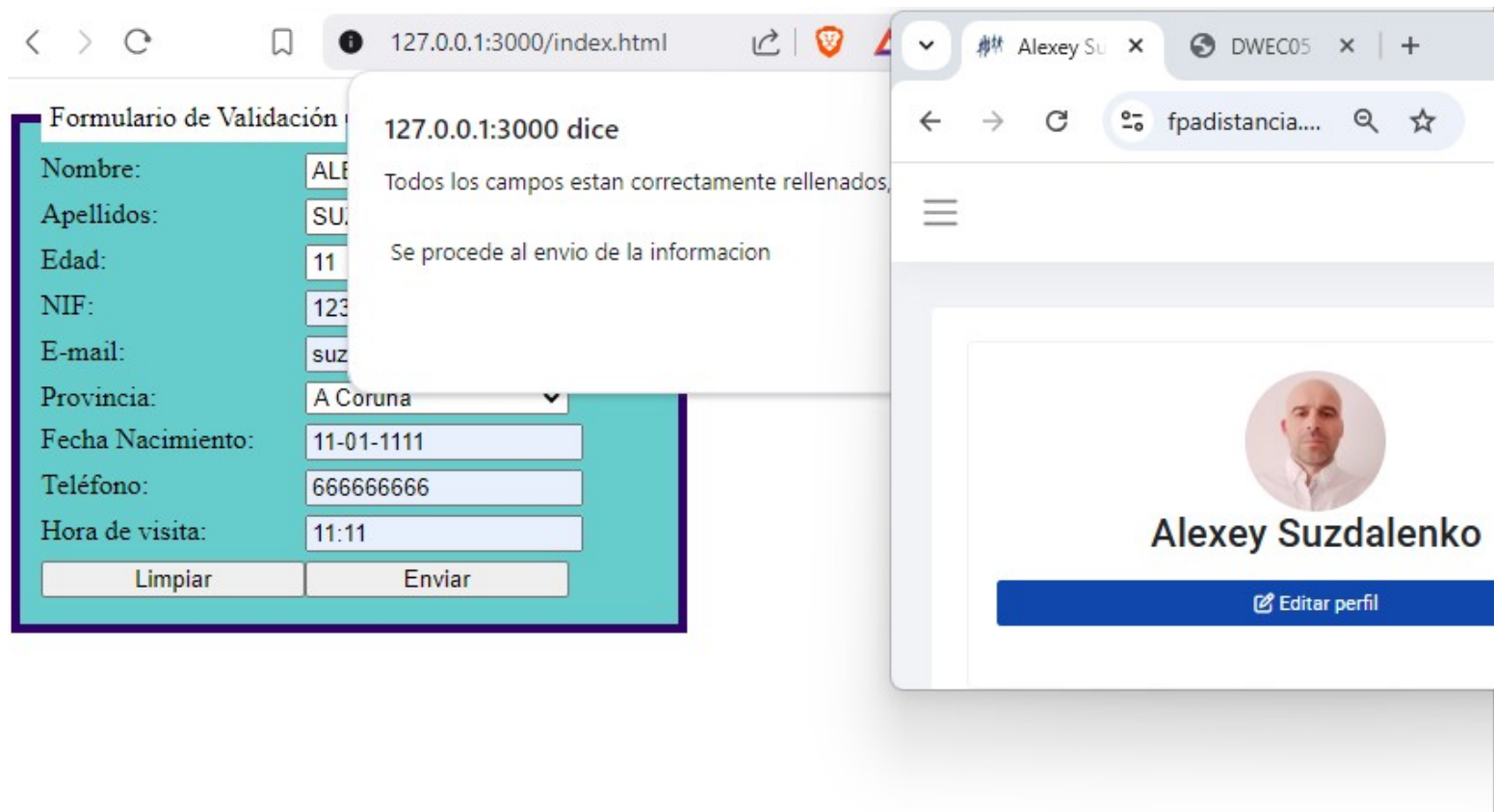
·
Captura de pantalla para ver los errores, en el caso si no ingreso ningún dato puedo ver el máximo de los errores



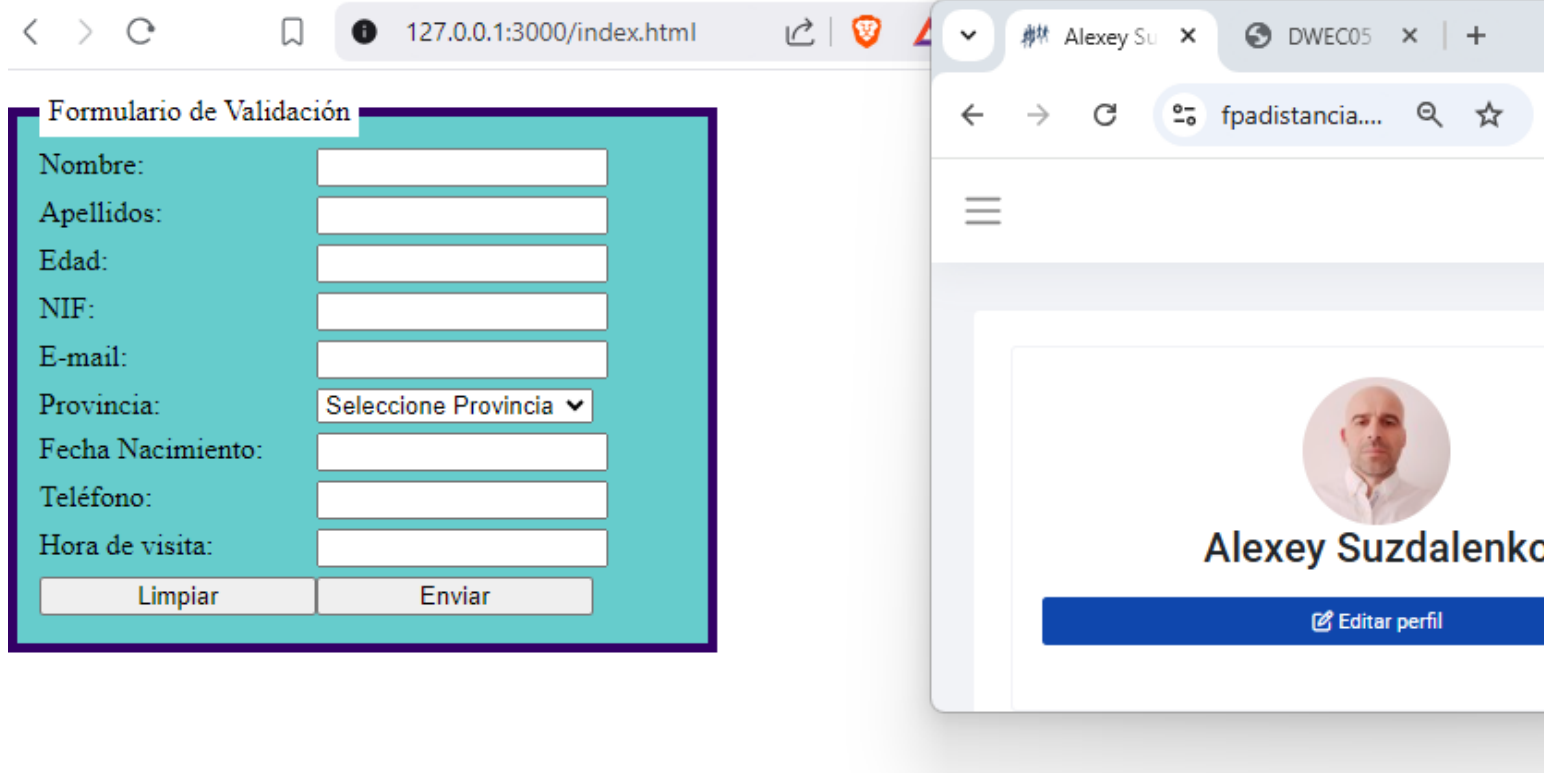
Captura de pantalla si ingreso los datos **correctamente** y presiono el botón “Enviar”



Captura de pantalla si ingreso todos los datos correctamente y confirmo el envío:



Captura de pantalla tras pulsar el botón “Limpiar”:



Ahora mostrare por separado el código de las funciones que se piden en el anunciado:

Cada vez que estos campos pierdan el foco, su contenido debe convertirse a mayúsculas automáticamente

```

/* perdida de foco por parte de nombre y apellido */
nombreInput.addEventListener('blur', () => { nameHasLostFocus(); });
apellidosInput.addEventListener('blur', () => { nameHasLostFocus(); });

```

```

function nameHasLostFocus(){
    nombreInput.value = String(nombreInput.value).toUpperCase();
    apellidosInput.value = String(apellidosInput.value).toUpperCase();
}

```

Nombre y Apellidos: No pueden estar vacíos. Si lo están, se debe mostrar un mensaje en el contenedor "errores" y enfocar el campo correspondiente

```

/* nombre de apellidos no pueden estar vacios */
function nameConNotBeEmpty(){
    if(!nombreInput.value.trim()) return true;
    if(!apellidosInput.value.trim()) return true;
    return false;
}

```

Edad: Debe ser un número entre 0 y 120. Si no cumple esta condición, se debe mostrar un error.

```

/* verificacion edad */
function verifyAge(){
    if(parseInt(edadInput.value) >= 0 && parseInt(edadInput.value) <= 120) return false;
    return true;
}

```

NIF: Debe cumplir el formato de 8 números, un guion y una letra (no es necesario validar la letra). Usa una expresión regular para esto.

```

/* validacion del NIF */
function nifValidate(){
    const regex = /^\\d{8}-[A-Za-z]$/;
    return !regex.test(nifInput.value);
}

```

Email: Debe tener un formato válido de dirección de correo electrónico (ejemplo: nombre@dominio.com)

```

/* validacion email */
function emailVerify(){
    let email = emailInput.value;
    // Verifica si incluye "@" y "."
    if (!email.includes("@") || !email.includes(".")) { return true; }
    // Obtiene las partes del correo
    let partes = email.split("@");
    // Verifica que haya exactamente una "@"
    if (partes.length !== 2) { return true; }
    let [localPart, domainPart] = partes;
    // Verifica que la parte local no esté vacía
    if (localPart.length === 0) { return true; }
    // Verifica que la parte del dominio tenga al menos un "."
    if (!domainPart.includes(".")) { return true; }
    // Verifica que el dominio tenga al menos un carácter antes y después del "."
    let dominioSplit = domainPart.split(".");
    if (dominioSplit.length < 2 || dominioSplit.some(part => part.length === 0)) { return true;}
    return false;
}

```

Provincia: Es obligatorio seleccionar una provincia de la lista desplegable.

```

/* validacion provincia */
function provinceValidation(){
    if(provinciaSelect.value == 0 || provinciaSelect.value == '') return true;
    return false;
}

```

Fecha de Nacimiento: Debe tener el formato dd/mm/aaaa o dd-mm-aaaa . Usa una expresión regular.

```

function brithDateValidate(){
    let regex = /^(0[1-9]|[12][0-9]|3[01])-(0[1-9]|1[0-2])-\\d{4}$/;
    return !regex.test(fechaInput.value);
}

```

Teléfono: Debe contener exactamente 9 dígitos numéricos.

```

function phoneValidate(){
    let telefonoStr = String(telefonoInput.value);
    if (telefonoStr.length !== 9) { return true; }
    for (let i = 0; i < telefonoStr.length; i++) {
        if (telefonoStr[i] < '0' || telefonoStr[i] > '9') {
            return true;
        }
    }
    return false;
}

```

Hora de visita: Debe cumplir el formato hh:mm .

```
function visitingTimeValidate() {
  let partes = horaInput.value.split(':');
  if (partes.length !== 2) { return true; }
  let [hh, mm] = partes;
  if (isNaN(hh) || isNaN(mm) || !Number.isInteger(+hh) || !Number.isInteger(+mm)) { return true; }
  const horas = parseInt(hh, 10);
  const minutos = parseInt(mm, 10);
  if (horas < 0 || horas > 23 || minutos < 0 || minutos > 59) { return true; }
  return false;
}
```

Confirmación antes de enviar: Antes de enviar los datos, el sistema debe mostrar una ventana de confirmación. Si el usuario cancela, el envío debe detenerse. Incluyo “setTimeout” para el que el efecto no sea instantáneo y el usuario tenga la sensación que se están procesando los datos.

```
if(errorMessage == ''){
  errorsDiv.style.display = 'none';
  setTimeout(()=>{
    let confirmEnvioData = confirmEnvio();
    if(confirmEnvioData){
      setTimeout(()=>{ alert('Todos los campos estan correctamente rellenos, \n\r Se procede al
envio de la informacion'); }, 1111);
    }
  }, 1111);
}
```

```
function confirmEnvio(){
  let request = confirm('¿Enviar datos?');
  if(request){ return true; }
  return false;
}
```

Mensajes de error: Los errores deben mostrarse en el contenedor con id errores . Debe aparecer un mensaje claro indicando qué campo está mal.

```
document.getElementById('inputSubmit').addEventListener('click', (event) => {
  event.preventDefault();
  let errorMessage = '';
  errorsDiv.style.display = 'none';

  if(nameConNotBeEmpty()){ errorMessage += 'Nombre y Apellidos: No pueden estar vacíos <br><br>'; }
  if(verifyAge()) { errorMessage += 'Edad: Debe ser un número entre 0 y 120 <br><br>'; }
  if(nifValidate()) {errorMessage += 'NIF: Debe cumplir el formato de 8 números, un guion y una
letra <br><br>'; }
  if(emailVerify()) { errorMessage += 'Email: Debe tener un formato válido de dirección de correo
electrónico <br><br>'; }
  if(provinceValidation()) { errorMessage += 'Provincia: Es obligatorio seleccionar una provincia de
la lista desplegable <br><br>'; }
  if(brithDateValidate()) { errorMessage += 'Fecha de Nacimiento: Debe tener el formato dd/mm/aaaa o
dd-mm-aaaa <br><br>'; }
  if(phoneValidate()) { errorMessage += 'Teléfono: Debe contener exactamente 9 dígitos numéricos
<br><br>'; }
  if(visitingTimeValidate()){ errorMessage += 'Hora de visita: Debe cumplir el formato hh:mm'; }

  errorsDiv.innerHTML = errorMessage;
}
```

});