

23-25 Agosto 2022

# Turing-IA

Conocimiento en Front-End y Back-End  
Evidencia.



**Luis Fernando Del oso Garcia**  
Postulante becario de desarrollo de software

## Evidencia en Front-End.

```
index.html > html > head
You, 5 hours ago | 1 author (You)
1 <html lang="es">
2
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>AI-Soft</title>
8   <!-- esto es el css de bootstrap --> You, 6 hours ago • Add files
9   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0-beta1/dist/css/bootstrap
10  integrity="sha384-0evHe/X+R7YkIZDRvuzKMRqM+OrBnVFBL6DOitfPri4tjfhXaWutUpFmBp4
11  <!-- JavaScript de bootstrap -->
12 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0-beta1/dist/js/bootstrap
13 integrity="sha384-pprn3073KE6t6bjs2QrFaJGz5/SUsLqktiwsUTF55Jfv3qYSDhgCecCxMW!
14 crossorigin="anonymous"></script>
15 <!-- esta es laparte donde importo los estilos css -->
16 <link rel="stylesheet" href="style.css">
17 <!-- este es el script para los iconos de awesome -->
18 <script src="https://kit.fontawesome.com/223aad6dab.js" crossorigin="anonymous"><
19
20 <!-- Script principal -->
21 <script src="main.js" type="module"></script>
22 <script src="scroll.js"></script>
23 </head>
24
25 <body>
26   <!-- esto es el menu -->
27   <nav class="navbar navbar-expand-lg w-100 position-fixed" id="navbar-fixed">...
51   </nav>
52   <!-- video del menu -->
53   <div class="video-container mb-5">...
65   </div>
66
67   <div class="container" id="ours">...
83   </div>
84
85   <!-- este es el contenedor principal -->
86   <div class="container-fluid no-gutter mt-5">...
159  </div>
160   <!-- esta es la parte del footer -->
161   <div class="container-fluid container-footer">...
217  </div>
218 </body>
219
220 </html>
```

Este es el index echo por HTML.

```
style.css > body
...
1 body { You, 6 hours ago • create style.css
2   margin: 0;
3   padding: 0;
4 }
5 .navbar-collapse {
6   flex-grow: unset;
7 }
8 .navbar-expand-lg {
9   padding-left: 5%;
10  padding-right: 5%;
11 }
12 .video-container {
13   background-color: #222222;
14   height: auto;
15   position: relative;
16   width: 100%;
17 }
18 .container-textStart {
19   color: white;
20   z-index: 50;
21   position: absolute;
22   left: 5%;
23   bottom: 30%;
24 }
25 .txtS1 {
26   font-size: 80px;
27 }
28 .txtS2 {
29   font-size: 38px;
30   font-weight: 700;
31 }
32 .container-arrow {
33   font-size: 50px;
34   color: white;
35   z-index: 50;
36   position: absolute;
37   left: 50%;
38   bottom: 20px;
39   cursor: pointer;
40 }
41 .video-header {
42   height: 90%;
43   object-fit: cover;
44   width: 100%;
45 }
46 .overlay {
47   background-color: rgba(0, 0, 0, 0.3);
48   height: 100%;
```

Este código contiene los estilos creados por CSS.


```

main.js > ...
You, 6 hours ago | 1 author (You)
1 import { services, contact } from "../utils/data.js";
2 import { insertData } from "../utils/insertCustomer.js";
3
4 const setServices = () => {
5   // map regresa un nuevo arreglo
6   const columns = services
7     .map(({ img, title, description }) => {
8     // desestructuramos los objetos que están dentro del arreglo
9     // para meter una variable dentro de un string usamos las comillas invertidas !
10    return `
11      <div class="col-md-6 col-lg-4 col-sm-12">
12        <div class="card">
13          
14          <div class="card-body text-trun">
15            <h6 class="card-title my-3">${title}</h6>
16            <p class="card-text">${description}</p>
17          </div>
18          <div class="card-footer">
19            <button class="btn-vm">Ver más</button>
20          </div>
21        </div>
22      </div>`;
23    })
24    .join("");
25    const tag = document.getElementById("services"); // buscamos un elemento HTML por ID
26    tag.innerHTML = columns; // inserta código HTML
27  };
28
29 const setContact = () => {
30   // sacamos las propiedades o keys del arreglo de objetos...
31   const columns = contact
32     .map(({ title, description, icon }) => {
33     return `
34       <div class="col-6">
35         <p class="h5"><i class="${icon}"></i>${title}</i></p>
36         <p>${description}</p>
37       </div>`;
38     })
39     .join("");
40   // n
41   const tag = document.querySelector("#form");
42   // insertamos en el dom el html construido de columns
43   tag.innerHTML = columns;
44 };
45
46 // esta función almacena dos funciones flecha...
47 const functions = () => {
48   setServices();

```

En este código importamos los servicios que usa HTML con el lenguaje JavaScript.

```

utils >  datajs > ...
You, 6 hours ago | 1 author (You)
1 export const services = [ You, 6 hours ago • Add files
2 {
3   id: 1,
4   img: "https://www.turing-ia.com/assets/img/balog.jpg",
5   title: "IMPLEMENTACIÓN DE PROYECTOS (BUSINESS ANALYST)",
6   description:
7     "Nuestra metodología de servicios se encuentra orientada en los estándares del I
8 },
9 {
10  id: 2,
11  img: "https://www.turing-ia.com/assets/img/code.jpg",
12  title: "DESARROLLO DE APLICACIONES",
13  description:
14    'En un mundo moderno, con la cantidad de problemas y su complejidad cada vez es
15 },
16 {
17  id: 3,
18  img: "https://www.turing-ia.com/assets/img/datad.jpg",
19  title: "WORKSHOP (BLUE PRINT)",
20  description:
21    "Su estrategia debe abarcar la gestión de personas, procesos y cambios, incluid
22 },
23 {
24  id: 4,
25  img: "https://www.turing-ia.com/assets/img/caplog.jpg",
26  title: "CAPACITACIÓN",
27  description:
28    "Nuestra gama de cursos de Tableau incluye: Tableau Prep Tableau Desktop Visual
29 },
30 {
31  id: 5,
32  img: "https://www.turing-ia.com/assets/img/vhar.jpg",
33  title: "VENTA DE SOFTWARE",
34  description:
35    "Nos dedicamos a la venta de licencias de software, podemos recomendarle los si
36 },
37 {
38  id: 6,
39  img: "https://www.turing-ia.com/assets/img/vhar.jpg",
40  title: "VENTA DE HARDWARE",
41  description:
42    "Trabajamos con los mejores proveedores de México y tenemos la capacidad de cubi
43 },
44 ];
45
46 > export const contact = [ ...
72 ];

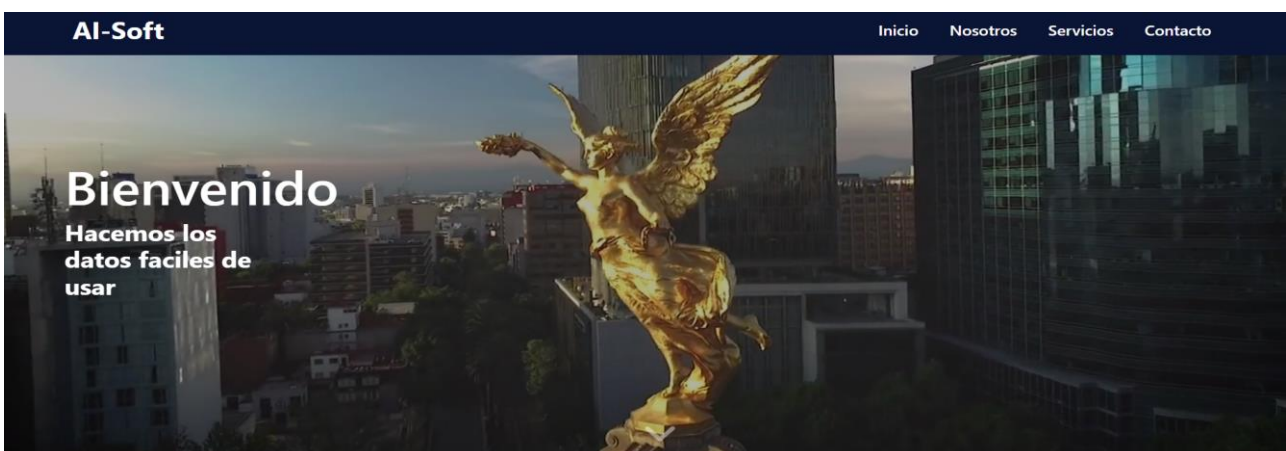
```

Estos son algunos de los servicios que exportamos echo en JavaScript.

```
scroll.js > ...
You, 6 hours ago | 1 author (You)
1 // esto escucha el evento scroll en la vewntana You, 6 hours ago • Add files
2 window.addEventListener("scroll", () => {
3     // en esta parte buscamos un elemento por su id...
4     const navbar = document.querySelector("#navbar-fixed");
5     //en esta parte obtenemos el scroll en el eje de las Y y lo almacenamos en una con:
6     const scroll = window.scrollY;
7
8     if (scroll > 0) {
9         //agregarnos la clase navbar-blue cuando el scroll es mayor a 0
10        navbar.classList.add("navbar-blue");
11    } else {
12        //agregarnos la clase navbar-blue cuando el scroll es igual o menor 0
13        navbar.classList.remove("navbar-blue");
14    }
15 });
16
```

En este script usamos un propiedad para que cuando escuche el evento 'scroll' mediante addEventListener se active así dándole una mejora a la página principal.

Resultado en pagina principal:



## Turing-IA, tu mejor opción

Somos una empresa Consultora de Big Data en México, desde hace 13 años proporcionamos servicios en Business Analytics mediante Knowledge Discovery in Database (KDD). Somos un grupo de especialistas en proyectos de Business Analytics con más de 13 años de experiencia, realizando análisis descriptivos, predictivos y prescriptivos.



## TECNOLOGÍAS QUE UTILIZAMOS



## LO QUE TE OFRECEMOS



### IMPLEMENTACIÓN DE PROYECTOS (BUSINESS ANALYST)

Nuestra metodología de servicios se encuentra orientada en los estándares del Project Management Institute (PMI) & a

[Ver más](#)

### DESARROLLO DE APLICACIONES

En un mundo moderno, con la cantidad de problemas y su complejidad cada vez es mayor, el desarrollo de aplicaciones, una

[Ver más](#)

### WORKSHOP (BLUE PRINT)

Su estrategia debe abarcar la gestión de personas, procesos y cambios, incluida la identificación de casos de uso comercial

[Ver más](#)

Solo podemos ver poco del futuro, pero lo suficiente para darnos cuenta de que hay mucho que hacer...

— Alan Turing



## CONTACTA CON NUESTRAS OFICINAS

Si tienes alguna pregunta sobre nuestro servicios, estamos disponibles...

**Dirección**

Av. Insurgentes Sur 601,  
Nápoles, Benito Juárez, 03810  
Ciudad de México, México

**Horario**

Lunes a Viernes  
7 a.m. - 7 p.m. UTC-6

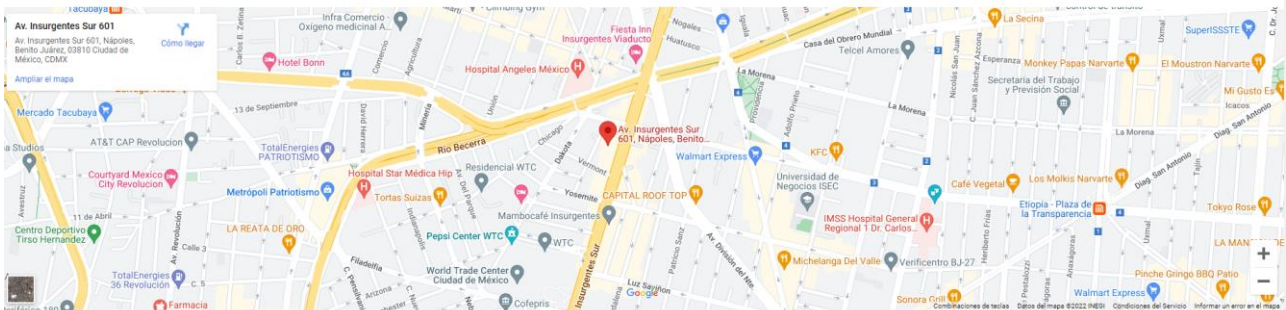
**Teléfono**

+52 (722) 533-6638

**E-mail**

contacto@turing-ia.com

## PONTE EN CONTACTO CON NOSOTROS

**AI-Soft**

Av. Insurgentes 601 Col. Nápoles,  
Benito Juárez Ciudad de México,  
México

Teléfono de contacto:  
+52 (722) 936-96-65

Correo electrónico:  
contacto@turing-ia.com

**Nuestro servicio**

Implementación de Proyectos  
Workshop (Blue Print)  
Capacitación  
Venta de Software  
Venta de Hardware  
Desarrollo de Aplicaciones  
Balanced Scorecard

**Mapa de sitio**

Inicio  
Nosotros  
Servicios

**Inscríbete a nuestro newsletter****Síguenos en nuestras redes**



## Evidencia Back-End

```
backend > turing.sql
You, 6 hours ago | 1 author (You)
1 DROP DATABASE IF EXISTS bdturing;
2 CREATE DATABASE bdturing;
3 USE bdturing;
4
5 CREATE TABLE clientes(
6     id_clientes TINYINT(2) NOT NULL PRIMARY KEY AUTO_INCREMENT,
7     nombre_completo VARCHAR(20),
8     email VARCHAR(20),
9     telefono VARCHAR(20),
10    empresa VARCHAR(20),
11    mensaje VARCHAR(20)
12 );
13
14 CREATE TABLE email(
15     id_email TINYINT(2) NOT NULL PRIMARY KEY AUTO_INCREMENT,
16     email VARCHAR(20)
17 );
18
19 DELIMITER $$
20
21 CREATE PROCEDURE `almacenado` ()
22 BEGIN
23     SELECT * FROM clientes;
24 END$$
25
26 CREATE PROCEDURE `correosE` ()
27 BEGIN
28     SELECT * FROM email;
29 END$$
30
31 DELIMITER ;
32
33 call almacenado();
34 call correosE();
```

Aquí podemos ver las estructura SQL con las que se hizo las base de datos.

```
backend > apiPhp > config > db.php > ...
You, 6 hours ago | 1 author (You)
1 <?php
2
3 You, 6 hours ago | 1 author (You)
4 class Database {
5     private $host = "localhost";
6     private $database_name = "bdturing";
7     private $username = "root";
8     private $password = "";
9     public $conn;
10
11     public function getConnection(){
12         $this->conn = null;
13         try{
14             $this->conn = new PDO("mysql:host=" . $this->host . ";dbname=" . $this->d.
15             $this->conn->exec("set names utf8");
16         }catch(PDOException $exception){
17             echo "Error de conexion con la base de datos: " . $exception->getMessage(
18         }
19         return $this->conn;
20     }
21 }
```

En este scrip vemos el código echo en PHP donde se hace la conexión as la BD.

backend > apiPhp > class > customers.php > ...

You, 6 hours ago | 1 author (You)

```
1 <?php You, 6 hours ago * add backend
```

2

You, 6 hours ago | 1 author (You)

```
3 class Customer
```

```
4 {
```

```
5     private $conn;
```

6

```
7     // Propiedades
```

```
8     public $nombreCompleto;
```

```
9     public $email;
```

```
10    public $telefono;
```

```
11    public $empresa;
```

```
12    public $mensaje;
```

13

14

```
15    // Metodo constructor
```

```
16    public function __construct($db)
```

```
17    {
```

```
18        $this->conn = $db;
```

```
19    }
```

20

```
21    // metodo para crear clientes
```

```
22    public function createCustomer()
```

```
23    {
```

```
24        //Asiganamos el Query de insert into
```

```
25        $sql_query = "INSERT INTO clientes
```

```
26            SET
```

```
27                nombre_completo = :nombre_completo,
```

```
28                email = :email,
```

```
29                telefono = :telefono,
```

```
30                empresa = :empresa,
```

```
31                mensaje = :mensaje
```

```
32            ;";
```

33

34

```
35        //preparamaos la conexion con los datos que se insertaran desde create
```

```
36        $statement = $this->conn->prepare($sql_query);
```

```
37        $statement->bindParam(':nombre_completo', $this->nombreCompleto);
```

```
38        $statement->bindParam(':email', $this->email);
```

```
39        $statement->bindParam(':telefono', $this->telefono);
```

```
40        $statement->bindParam(':empresa', $this->empresa);
```

```
41        $statement->bindParam(':mensaje', $this->mensaje);
```

42

```
43        // ejecutamos el query y retornamos un true o false con un ternario
```

```
44        return $statement->execute() ? true : false;
```

```
45    }
```

```
46 }
```

47

En este script asignamos el query (consulta) de la BD con las propiedades que se solicitan.

```

backend > apiPhp > create.php > ...
You, 6 hours ago | 1 author (You)
1  <?php      You, 6 hours ago * add backend
2  // Headers para el endpoint
3  header("Access-Control-Allow-Origin: *");
4  // datos en json
5  header("Content-Type: application/json; charset=UTF-8");
6  // Permitimos el metodo post
7  header("Access-Control-Allow-Methods: POST");
8  header("Access-Control-Allow-Headers: Content-Type, Access-Control-Allow-Headers, Autl
9
10
11 //Importamos la clase Database para utilizar su metodo getConnection()
12 include_once './config/db.php';
13
14
15 //importamos la clase Customer para utilizar su metodo constructor y createCustomer()
16 include_once './class/customers.php';
17
18 // Creamos una nueva variable de tipo de Database
19 $database = new Database();
20 $connection = $database->getConnection();
21
22 // Creamos una nueva variable de tipo Customer
23 $customer = new Customer($connection);
24
25 // decodificamos el json de los datos recibidos
26 $data = json_decode(file_get_contents("php://input"));
27
28 // Asignamos los valores traídos a cada propiedad correspondiente
29 $customer->nombreCompleto = $data->nombreCompleto;
30 $customer->email = $data->email;
31 $customer->telefono = $data->telefono;
32 $customer->empresa = $data->empresa;
33 $customer->mensaje = $data->mensaje;
34
35 //Ejecutamos la función createCustomer() e imprimimos una respuesta
36 echo $customer->createCustomer() ? 'ok': 'failed';
37

```

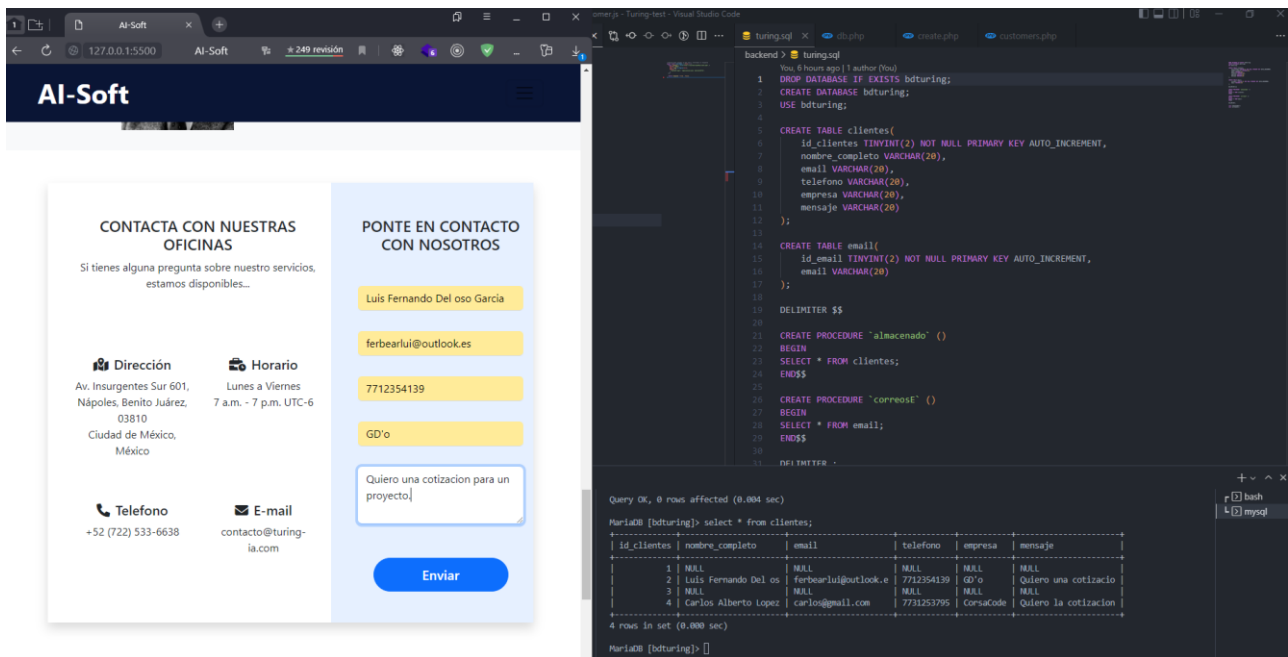
**En este script montamos un Api Rest echo en PHP.**

```

utils > insertCustomer.js > ...
You, 2 hours ago | 1 author (You)
1 //funcion para consumir la api rest y retornamos un respuesta
2 export const insertData = (form) => {
3   const response = fetch("http://localhost/apiPhp/create.php", {
4     method: "POST",
5     body: JSON.stringify(form),
6     headers: {
7       "Content-type": "application/json; charset=UTF-8",
8     },
9   });
10
11   return response ? true : false;
12 };

```

**Por ultimo consumimos la Api Rest por medio de un endpoint echo en JavaScript para el consumo total de la Base de Datos de la pagina web...**



En este captura se muestra el funcionamiento del Front-End y Back-End con el llenado del formulario a la izquierda y capturando datos en la derecha...  
Fin del proyecto.

Proyecto entero en GitHub:

<https://github.com/LFDeLoso/Turing-test.git>