

UNIVERSIDAD DE SONORA



**"El saber de mis hijos
hará mi grandeza"**

DIVISIÓN DE INGENIERÍA

Sistema Bitácora Procesal

Reporte Final de Prácticas Profesionales

Alumno:

Efrén David Lugo Cueva

TUTORA:

DRA. MERY HELEN PESANTES ESPINOZA

Hermosillo, Son. 28 abril, 2020

INDICE GENERAL

INDICE GENERAL.....	1
1. INTRODUCCIÓN.....	4
1.1 EXPLICACIÓN DEL PROYECTO.....	4
1.2 OBJETIVOS DEL PROYECTO.....	4
1.3 METODOLOGÍA.....	4
2. DESCRIPCIÓN DEL CONTEXTO.....	5
2.1 ENTORNO DONDE SE LOCALIZA LA UNIDAD RECEPTORA.....	5
2.2 EQUIPAMIENTO E INTALACIONES DONDE SE DESARROLLARON LAS ACTIVIDADES DE PRÁCTICAS PROFESIONALES.....	6
2.3 DESCRIPCIÓN DE LA NORMATIVIDAD O REGLAS DE OPERACIÓN DEL SISTEMA O UNIDAD RECEPTORA.....	7
3. FUNDAMENTO TEÓRICO DE LAS HERRAMIENTAS Y CONOCIMIENTOS APLICADOS.....	7
3.1 FRAMEWORK ANGULAR.....	7
3.2 BASES DE DATOS.....	8
3.2 SQL SERVER 2017 MANAGEMENT.....	8
3.4 ADO.NET ENTITY FRAMEWORK.....	9
3.5 DESARROLLO DE SISTEMAS BASADOS EN WEB.....	9
3.6 FRAMEWORK BOOTSTRAP.....	10
3.7 VISUAL STUDIO CODE.....	10
4. DESCRIPCIÓN DETALLADA DE LAS ACTIVIDADES.....	10
4.1 RETROALIMENTACIÓN DEL SISTEMA.....	11
4.2 INSTALACIÓN DE LAS HERRAMIENTAS NECESARIAS.....	11
4.3 CREACIÓN DE DIAGRAMA DE LA BASE DE DATOS.....	11
4.3.1 Recopilar datos.....	11
4.3.2 Definir claves primarias.....	11
4.3.3 Crear relaciones entre tablas.....	11
4.3.3.1 Uno a muchos.....	11
4.3.3.2 Muchos a muchos.....	12
4.3.3.3 Uno a uno.....	12
4.3.4 Independencia estructural.....	13
4.3.5 Simplicidad conceptual.....	13
4.3.6 Facilidad de diseño, implementación, mantenimiento y uso.....	13
4.4 CREACIÓN DE FORMULARIO REACTIVO PARA EL REGISTRO.....	13
4.4.1 Paso 1: crear el modelo.....	13

4.4.2 Paso 2: Ahora en nuestro componente creamos un FormGroup con FormControl para cada uno de los campos:	14
4.4.3 Paso 3: Ahora creamos el formulario en el HTML del template y conectamos los campos como hemos hecho antes:.....	14
4.4.4 Paso 4: El resultado es:.....	15
4.5 CREANDO EL PROYECTO WEBAPI	16
4.5.1 Ejecutando el Proyecto WebAPI.....	17
4.6 CREACIÓN DE TABLA PARA MOSTRAR LOS REGISTROS REALIZADOS	20
4.6.1 Instalación.....	20
4.6.2 Ejemplo de configuración mínima.....	20
5. ANÁLISIS DE LA EXPERIENCIA ADQUIRIDA	23
5.1 ANÁLISIS GENERAL DEL SISTEMA, SU DISEÑO, DESARROLLO Y ORGANIZACIÓN.....	23
5.2 ANÁLISIS DE LOS OBJETIVOS DEL PROYECTO.....	23
5.3 ANÁLISIS DE LAS ACTIVIDADES REALIZADAS	23
6. CONCLUSIONES Y RECOMENDACIONES	24
7. REFERENCIAS BIBLIOGRÁFICAS Y VIRTUALES	25
9. CARTA DE FINALIZACIÓN	27

Ilustraciones:

Ilustración 2.1 Mapa de la ubicación de la Fiscalía General de Justicia Del Estado	5
Ilustración 2.2 Edificio donde se encuentra la unidad Receptora.....	6
Ilustración 2.3 Oficina donde realice mi estancia profesional.	6
Ilustración 3 componentes de una BD.	8
Ilustración 4 Ejemplo de tablas con relación uno a muchos.....	12
Ilustración 4.1 Ejemplo de tablas con relación Muchos a muchos	12
Ilustración 4.2 Ejemplo de tablas con relación uno a uno	12
Ilustración 4.3 Formulario.	15
Ilustración 4.4 Creando proyecto.....	16
Ilustración 4.5 Seleccionando plantilla.....	16
Ilustración 4.6 Proyecto generado.	17
Ilustración 4.7 Estructura de la WebApi.....	19
Ilustración 4.8 Tabla dinámica.....	21
Ilustración 4.9 Tabla dinámica con datos.....	22

1. INTRODUCCIÓN

Los alumnos que cubrirán un grado de licenciatura en la Universidad de Sonora tienen un requisito que cubrir que forma parte del plan estudios, el cual requiere que poner en práctica los conocimientos teóricos que se han ido adquiriendo durante la carrera. Tal actividad es conocida como las prácticas profesionales, las cuales pueden ser realizadas en alguna unidad del sector público o del sector privado.

En la carrera de Ingeniería en Sistemas de Información, que se encuentra dentro del Departamento de Ingeniería Industrial, este requisito se manifiesta como una estancia profesional. La estancia profesional debe de durar al menos 340 horas, lo que en el plan de estudios se traduciría a 20 créditos.

Por lo tanto, el presente trabajo es un reporte de las actividades realizadas durante las practicas realizadas en la Fiscalía General de Justicia del Estado.

1.1 EXPLICACIÓN DEL PROYECTO

La **Fiscalía General de Justicia Del Estado De Sonora** tenía la necesidad de crear un sistema de registro de sentencias condenatorias, absolutorias y mixtas, ya que, el llenado de las sentencias se estaba realizando por medio de un Excel, por lo que optaron realizar el sistema denominado "**BitPro**".

La Dirección de Sistemas es la que se encargaría de realizar el proyecto.

El sistema considera lo siguiente:

Sentencia condenatorias: Cuando se comprueban los elementos del cuerpo del delito y la responsabilidad del sentenciado imponiéndole, como consecuencia, una pena o medida de seguridad, sin perder de vista que conforme al artículo 21 C., la imposición de las penas es propia y exclusiva de la autoridad judicial. [1]

Sentencia absolutoria: Emitida por el Tribunal de Enjuiciamiento en forma escrita, estimando que el imputado no es culpable del delito más allá de toda duda razonable. Tiene como efecto ordenar la cancelación de las medidas cautelares que se hayan impuesto en todo registro público y policial, y debe cumplirse inmediatamente. [2]

1.2 OBJETIVOS DEL PROYECTO

El objetivo del proyecto que se me asignó por la Dirección de Sistemas de Información de La **Fiscalía General de Justicia Del Estado De Sonora** es brindar apoyo a las agencias que manejan el llenado de las sentencias, y facilitar el trabajo de estas personas, desarrollando el sistema "**BitPro**", el cual facilitaría el trabajo de las agencias.

El sistema tenía que permitir el registro de sentencias donde también podían consultar las que ya se registraron y modificarlas.

1.3 METODOLOGÍA

El desarrollo del sistema "**BitPro**" se realizó siguiendo una lista de tareas, las cuales se tenían que cumplir una por una, la lista de tareas se actualizaba dependiendo de las necesidades del cliente, y cada semana se tenía que informar al Jefe del departamento los avances, para que el pudiera verificar que si se realizó cumpliendo

los objetivos y metas que el sistema necesitaba. Se realizó en varias etapas: la primera etapa era la realización del esquema de la base de datos. La segunda etapa fue el diseño de interfaces que se quería para el sistema y la tercera etapa era la parte del Backend.

2. DESCRIPCIÓN DEL CONTEXTO

La **Fiscalía General de Justicia Del Estado De Sonora** se localiza en Hermosillo, Sonora [3].

Las funciones de la Fiscalía son:

- **Perseguir los delitos** del orden común cometidos en el territorio del Estado.
- **Velar por la legalidad** como uno de los principales rectores de la convivencia social, promover la pronta, completa e imparcial procuración e impartición de justicia.
- **Proteger los intereses** de personas menores de edad y con capacidades diferentes, así como los individuales y sociales, en los términos que determinen las leyes.
- **Cuidar la correcta aplicación** de las medidas de política criminal, en la esfera de su competencia.

2.1 ENTORNO DONDE SE LOCALIZA LA UNIDAD RECEPTORA

Las oficinas se localizan Blvd. Rosales y Paseo del Canal, Colonia Centro, Edificio Sur C.P. 83000, Hermosillo.

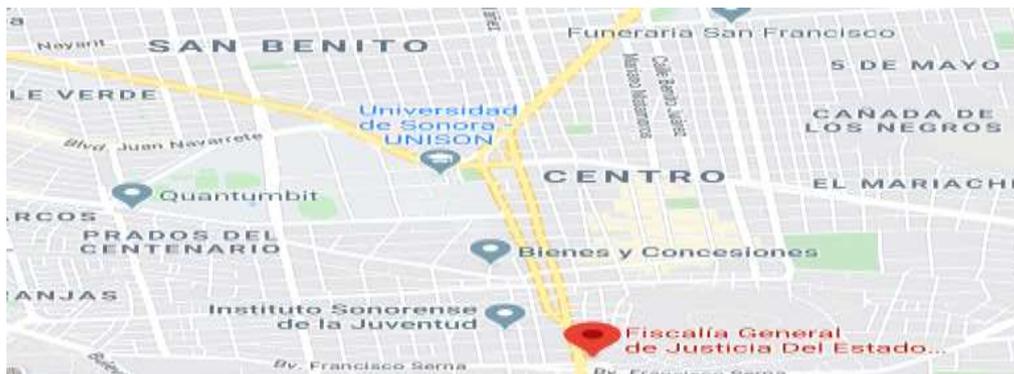


Ilustración 2.1 Mapa de la ubicación de la Fiscalía General de Justicia Del Estado



Ilustración 2.2 Edificio donde se encuentra la unidad Receptora.

2.2 EQUIPAMIENTO E INTALACIONES DONDE SE DESARROLLARON LAS ACTIVIDADES DE PRÁCTICAS PROFESIONALES

Las practicas profesionales las realicé en la de Dirección de Sistemas de Información.

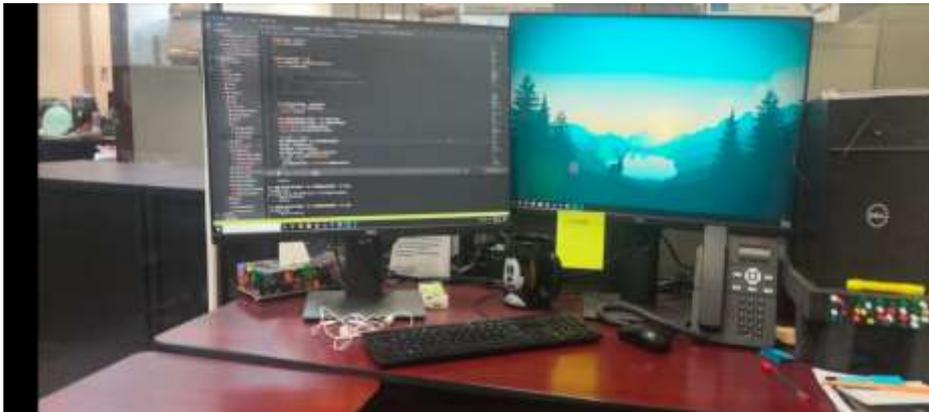


Ilustración 2.3 Oficina donde realice mi estancia profesional.

Las instalaciones contaban con 2 escritorios, 2 monitores, teléfono personal, servicio de internet, baño disponible y aire acondicionado

Para realizar mi estancia profesional me otorgaron una computadora Dell, con Windows 10.

2.3 DESCRIPCIÓN DE LA NORMATIVIDAD O REGLAS DE OPERACIÓN DEL SISTEMA O UNIDAD RECEPTORA.

Se tienen varias reglas que tienes que cumplir en la organización:

- Llegar a la hora indicada
- Respetar a tus compañeros
- Cuidar el equipo de computo
- Mantener limpio tu lugar de trabajo

Requisitos del sistema "BitPro"

- Amigable con el usuario
- Fácil de usar
- Campos validados
- Responsivo

3. FUNDAMENTO TEÓRICO DE LAS HERRAMIENTAS Y CONOCIMIENTOS APLICADOS.

Durante el desarrollo de este proyecto, se tuvo la oportunidad de aplicar los conocimientos que fueron aprendidos durante la carrera de Ingeniería en Sistemas de Información, los cuales fueron necesarios para poder desarrollar este proyecto de manera correcta y satisfactoria, además de que se adquirió experiencia utilizándolos como profesional en un ambiente laboral real. A continuación, se describen los conocimientos y herramientas que fueron utilizadas durante el desarrollo de este sistema.

3.1 FRAMEWORK ANGULAR

Angular es un framework opensource desarrollado por Google para facilitar la creación y programación de aplicaciones web de una sola página, las webs SPA (Single Page Application).

Angular separa completamente el frontend y el backend en la aplicación, evita escribir código repetitivo y mantiene todo más ordenado gracias a su patrón MVC (Modelo-Vista-Controlador) asegurando los desarrollos con rapidez, a la vez que posibilita modificaciones y actualizaciones.

En una web SPA aunque la velocidad de carga puede resultar un poco lenta la primera vez que se abre, navegar después es totalmente instantáneo, ya que se ha cargado toda la página.

Solamente es una ruta la que se tiene que enviar al servidor, y Angular lo que hace 'por debajo' es cambiar la vista al navegar para que dé la apariencia de una web normal, pero de forma más dinámica.

Entre otras ventajas, este framework es modular y escalable adaptándose a nuestras necesidades y al estar basado en el estándar de componentes web,

y con un conjunto de interfaz de programación de aplicaciones (API) permite crear nuevas etiquetas HTML personalizadas que pueden reutilizarse.

El lenguaje principal de programación de Angular es Typescript, y así toda la sintaxis y el modo de hacer las cosas en el código es el mismo, lo que añade coherencia y consistencia a la información, permitiendo por ejemplo, la incorporación de nuevos programadores, en caso de ser necesarios, ya que pueden continuar su trabajo sin excesiva dificultad. [4]

3.2 BASES DE DATOS

Una **base de datos** es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. En este sentido; una biblioteca puede considerarse una base de datos compuesta en su mayoría por documentos y textos impresos en papel e indexados para su consulta. Actualmente, y debido al desarrollo tecnológico de campos como la informática y la electrónica, la mayoría de las bases de datos están en formato digital, siendo este un componente electrónico, por tanto se ha desarrollado y se ofrece un amplio rango de soluciones al problema del almacenamiento de datos.

Hay programas denominados sistemas gestores de bases de datos, abreviado SGBD (del inglés Database Management System o DBMS), que permiten almacenar y posteriormente acceder a los datos de forma rápida y estructurada. Las propiedades de estos DBMS, así como su utilización y administración, se estudian dentro del ámbito de la informática. [5]

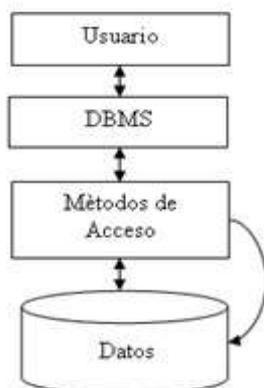


Ilustración 3 componentes de una BD.

3.2 SQL SERVER 2017 MANAGEMENT

SQL Server Management Studio (SSMS) es la interfaz de usuario cliente preferida y oficial con la cual se puede manejar, configurar, desplegar, actualizar y administrar una instancia SQL Server. La herramienta es bastante amigable para el usuario y también es exhaustiva. Es incluida en cada versión de SQL Server y es regularmente actualizada y mejorada. [6]

3.4 ADO.NET ENTITY FRAMEWORK

Entity Framework es un conjunto de tecnologías en ADO.NET que admite el desarrollo de aplicaciones de software orientadas a datos. Los arquitectos y desarrolladores de aplicaciones orientadas a datos han luchado con la necesidad de lograr dos objetivos muy diferentes. Deben modelar las entidades, las relaciones y la lógica de los problemas comerciales que están resolviendo, y también deben trabajar con los motores de datos utilizados para almacenar y recuperar los datos. Los datos pueden abarcar múltiples sistemas de almacenamiento, cada uno con sus propios protocolos; Incluso las aplicaciones que funcionan con un solo sistema de almacenamiento deben equilibrar los requisitos del sistema de almacenamiento con los requisitos de escribir código de aplicación eficiente y fácil de mantener.

Entity Framework permite a los desarrolladores trabajar con datos en forma de objetos y propiedades específicos del dominio, como clientes y direcciones de clientes, sin tener que preocuparse por las tablas y columnas subyacentes de la base de datos donde se almacenan estos datos. Con Entity Framework, los desarrolladores pueden trabajar a un nivel más alto de abstracción cuando manejan datos, y pueden crear y mantener aplicaciones orientadas a datos con menos código que en las aplicaciones tradicionales. [7]

3.5 DESARROLLO DE SISTEMAS BASADOS EN WEB

Los desarrolladores web diseñan y crean sitios web. Son responsables del aspecto del sitio. También son responsables de los aspectos técnicos del sitio, como su rendimiento y capacidad, que son medidas de la velocidad de un sitio web y de cuánto tráfico puede manejar el sitio. Además, los desarrolladores web pueden crear contenido para el sitio.

Al crear un sitio web, los desarrolladores deben hacer realidad la visión de sus clientes. Construyen tipos particulares de sitios web, como comercio electrónico, noticias o sitios de juegos, para satisfacer las necesidades de los clientes. Los diferentes tipos de sitios web requieren diferentes aplicaciones. Por ejemplo, un sitio de juegos debería ser capaz de manejar gráficos avanzados, mientras que un sitio de comercio electrónico necesitaría una aplicación de procesamiento de pagos. El desarrollador decide qué aplicaciones y diseños se adaptarán mejor al sitio.

Algunos desarrolladores manejan todos los aspectos de la construcción de un sitio web, y otros se especializan en un cierto aspecto del mismo. Los siguientes son ejemplos de tipos de desarrolladores web especializados:

Los desarrolladores web de back-end son responsables de la construcción técnica general del sitio web. Crean el marco básico del sitio y aseguran que funcione como se espera. Los desarrolladores web de back-end también establecen procedimientos para permitir que otros agreguen nuevas páginas al sitio web y se reúnan con la gerencia para discutir cambios importantes en el sitio.

Los desarrolladores web front-end son responsables de cómo se ve un sitio web. Crean el diseño del sitio e integran gráficos, aplicaciones (como una herramienta de pago minorista) y otro contenido. También escriben programas de diseño web en una variedad de lenguajes de computadora, como HTML o JavaScript. [8]

3.6 FRAMEWORK BOOTSTRAP

Bootstrap es un framework CSS y Javascript diseñado para la creación de interfaces limpias y con un diseño responsive. Además, ofrece un amplio abanico de herramientas y funciones, de manera que los usuarios pueden crear prácticamente cualquier tipo de sitio web haciendo uso de estos. [9]

3.7 VISUAL STUDIO CODE

Es un editor de código fuente desarrollado por Microsoft para Windows, Linux y macOS. Incluye soporte para depuración, control de Git integrado, resaltado de sintaxis, finalización de código inteligente, fragmentos de código y refactorización de código. También es personalizable, de modo que los usuarios pueden cambiar el tema del editor, los métodos abreviados de teclado y las preferencias. Es gratuito y de código abierto.

Características:

El código combina la interfaz de usuario optimizada de un editor moderno con asistencia y navegación de código enriquecido y una experiencia de depuración integrada, sin la necesidad de un IDE completo. Visual Studio Code, cuenta con herramientas de Debug hasta opciones para actualización en tiempo real de nuestro código en la vista del navegador y compilación en vivo de los lenguajes que lo requieran (por ejemplo, en el caso de SASS a CSS). Además de las extensiones, tendremos la posibilidad de optar por otros themes o bien configurarlo a nuestro gusto. Para modificar el esquema de colores y los iconos. [10]

4. DESCRIPCIÓN DETALLADA DE LAS ACTIVIDADES

Las actividades se llevaron a cabo en un periodo aproximado del 15 de octubre al 15 de diciembre del 2019.

Base de datos: Se recopilaron los datos necesarios para almacenar, crear tablas, definir claves primarias, definir atributos y crear relaciones entre tablas. Se crearon las tablas necesarias para realizar el sistema.

Diseño de interfaz: Se creó una interfaz amigable para el usuario, que fuera responsive, para que se pudiera adaptar a las diferentes pantallas y que cumpla con los estándares de validación.

Backend: Crear las conexiones con la base de datos, crear controladores, desarrollo de funciones e implementar la seguridad necesaria para el sitio web.

4.1 RETROALIMENTACIÓN DEL SISTEMA

Me explicaron como tenía que funcionar el sistema, lo que el sistema debía contener y con que herramientas trabajaría para lograr el correcto funcionamiento del sistema "BitPro".

4.2 INSTALACIÓN DE LAS HERRAMIENTAS NECESARIAS

Se hizo la instalación de los programas necesarios para poder desarrollar el sistema, las herramientas mencionadas en la sección 3. Se hicieron pruebas y se otorgaron los permisos necesarios para poder utilizar las bases de datos.

4.3 CREACIÓN DE DIAGRAMA DE LA BASE DE DATOS

4.3.1 Recopilar datos

Se deben recopilar los datos necesarios para almacenarlos en la base de datos. Estos datos se dividen en diferentes tablas.

Se debe elegir un tipo de datos apropiado para cada columna. Por ejemplo: números enteros, números de punto flotante, texto, fecha, etc.

4.3.2 Definir claves primarias

Para cada tabla se debe elegir una columna (o pocas columnas) como clave primaria, que identificará de forma única cada fila de la tabla. La clave primaria también se utiliza para hacer referencia a otras tablas.

4.3.3 Crear relaciones entre tablas

Una base de datos que consista de tablas independientes y no relacionadas tiene poco propósito. Las tablas que se tuvieron que crear para desarrollar el sistema son: sentencia, resolución, personas, caso y audiencia.

El aspecto más crucial en el diseño de una base de datos relacional es identificar las relaciones entre las tablas. Los tipos de relación son:

4.3.3.1 Uno a muchos

En una base de datos «Listado de clases», un maestro puede enseñar en cero o más clases, mientras que una clase es impartida por un solo maestro. Este tipo de relación se conoce como uno a muchos.

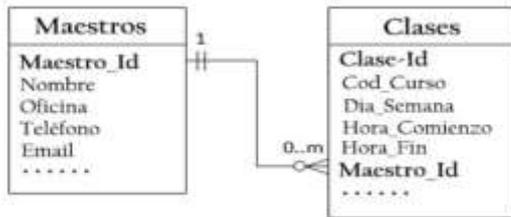


Ilustración 4 Ejemplo de tablas con relación uno a muchos.

4.3.3.2 Muchos a muchos

En una base de datos «Venta de productos», el pedido de un cliente puede contener varios productos, y un producto puede aparecer en varios pedidos. Este tipo de relación se conoce como muchos a muchos.

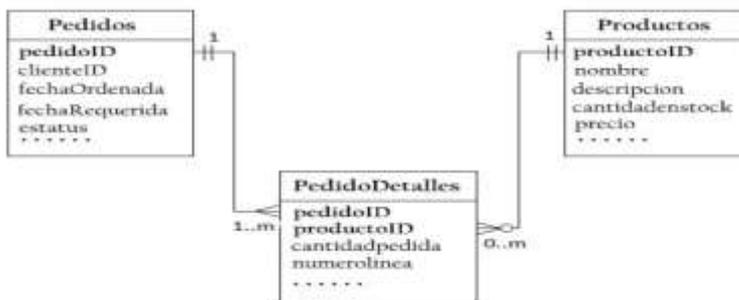


Ilustración 4.1 Ejemplo de tablas con relación Muchos a muchos

4.3.3.3 Uno a uno

En la base de datos «Venta de productos», un producto puede tener información opcional, como descripción adicional y su imagen. Mantenerla dentro de la tabla Productos generaría muchos espacios vacíos.

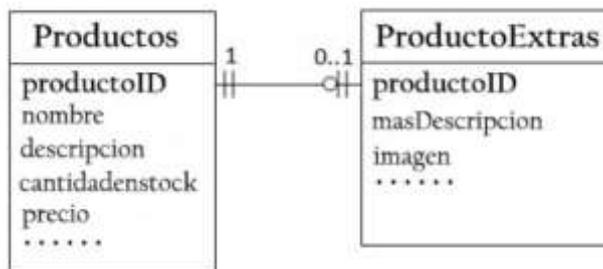


Ilustración 4.2 Ejemplo de tablas con relación uno a uno.

Ventajas

4.3.4 Independencia estructural

En el modelo relacional de base de datos, los cambios en la estructura de la base de datos no afectan el acceso a los datos.

Cuando es posible realizar cambios en la estructura de la base de datos sin afectar la capacidad del DBMS para acceder a los datos, se puede decir que se ha logrado la independencia estructural.

4.3.5 Simplicidad conceptual

El modelo relacional de base de datos es aún más simple a nivel conceptual que el modelo jerárquico o el de red de base de datos.

Dado que el modelo relacional de base de datos libera al diseñador de los detalles del almacenamiento físico de los datos, los diseñadores pueden concentrarse en la vista lógica de la base de datos.

4.3.6 Facilidad de diseño, implementación, mantenimiento y uso

El modelo relacional de base de datos logra tanto la independencia de los datos como la independencia de la estructura, lo que hace que el diseño, el mantenimiento, la administración y el uso de la base de datos sean mucho más fáciles que los otros modelos. [11]

4.4 CREACIÓN DE FORMULARIO REACTIVO PARA EL REGISTRO

Sé creo un formulario del registro de las sentencias, se utilizó el framework angular utilizando la herramienta de FormGroup para crear formularios reactivos, se siguió un ejemplo básico para crear este formulario. [12]

4.4.1 Paso 1: crear el modelo

```
export interface User {  
  name: string;  
  password: string;  
  passwordRepeat: string,  
}  
}
```

4.4.2 Paso 2: Ahora en nuestro componente creamos un FormGroup con FormControl para cada uno de los campos:

```
import { Component, OnInit } from '@angular/core';
import { FormControl, FormGroup } from '@angular/forms';

@Component({...})
export class SignupFormComponent implements OnInit {
  user: FormGroup;
  ngOnInit() {
    this.user = new FormGroup({
      name: new FormControl(""),
      password: new FormControl(""),
      passwordRepeat: new FormControl("")
    });
  }
}
```

4.4.3 Paso 3: Ahora creamos el formulario en el HTML del template y conectamos los campos como hemos hecho antes:

```
<form novalidate [formGroup]="user">

<label>

<span>Full name</span>

<input

type="text"

placeholder="Your full name"

formControlName="name">

</label>

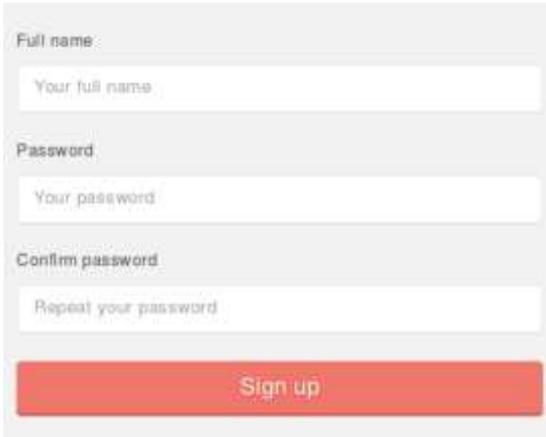
<label>

<span>Email address</span>

<input
```

```
type="password"
placeholder="Your password"
formControlName="password">
</label>
<label>
<span>Confirm address</span>
<input
type="password"
placeholder="Confirm your password"
formControlName="passwordRepeat">
</label>
<button type="submit">Sign up</button>
</form>
```

4.4.4 Paso 4: El resultado es:



The image shows a registration form with three input fields and a submit button. The first field is labeled 'Full name' and contains the placeholder text 'Your full name'. The second field is labeled 'Password' and contains the placeholder text 'Your password'. The third field is labeled 'Confirm password' and contains the placeholder text 'Repeat your password'. Below the fields is a red button with the text 'Sign up'.

Ilustración 4.3 Formulario.

4.5 CREANDO EL PROYECTO WEBAPI

Nos dieron un tutorial que debíamos de seguir para crear nuestra WebApi. [13]

Abrimos Visual Studio 2017 y vamos a crear una aplicación Web ASP.NET, en nuestro caso para .NET Framework

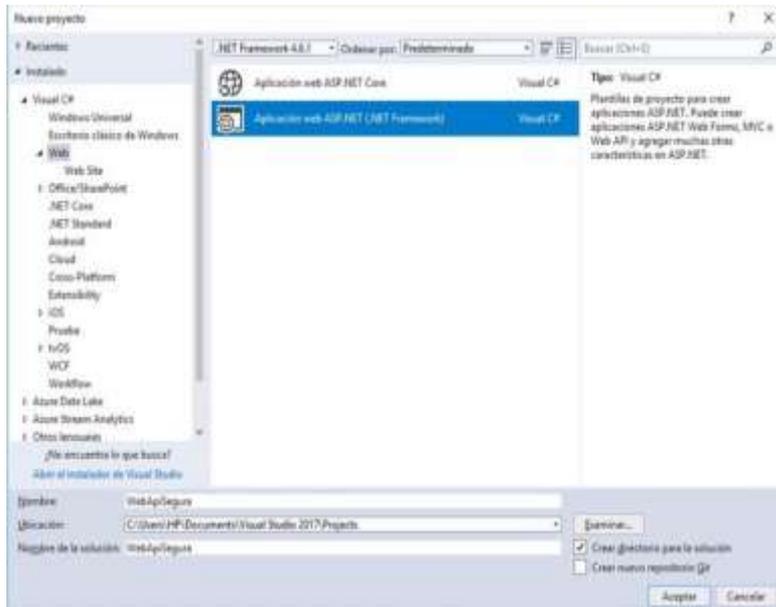


Ilustración 4.4 Creando proyecto.

Le damos el nombre de WebApiSegura y la ubicación por defecto de Visual Studio.

Ahora aparecerá la pantalla donde elegiremos la plantilla de ASP.NET que vamos a utilizar:

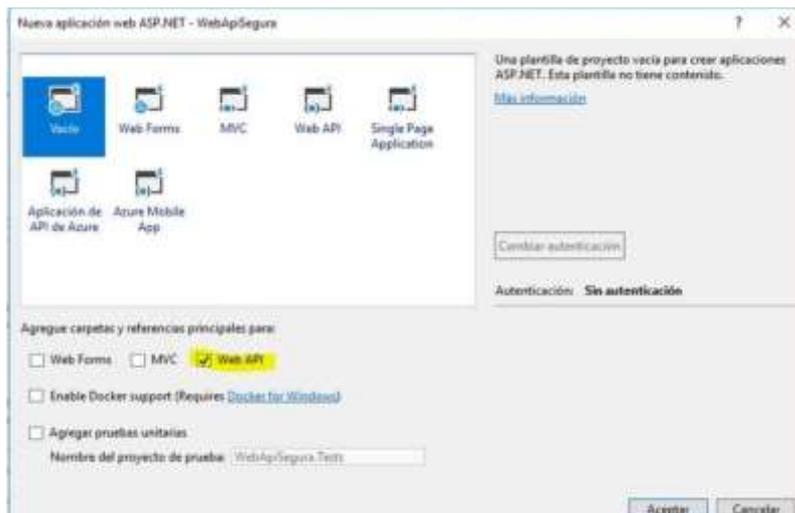


Ilustración 4.5 Seleccionando plantilla

Aquí, seleccionamos proyecto Vacío, check en WebAPI y pulsamos botón Aceptar para crear la solución, en este momento empezará a crearse nuestro proyecto.

Una vez generado el proyecto, lo veremos en el Explorador de soluciones de Visual Studio:



Ilustración 4.6 Proyecto generado.

4.5.1 Ejecutando el Proyecto WebAPI

Ejecutamos el proyecto, para ver que todo esta correcto y veremos la siguiente pantalla:

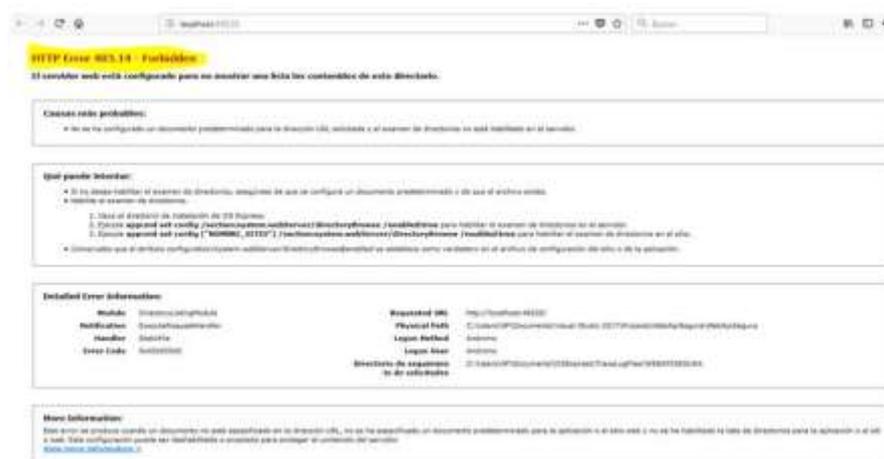


Ilustración 4.7 Proyecto ejecutado

Configurando el puerto para WebAPI

Por defecto, nuestra WebAPI se configura en un puerto al crear la solución, pero desde Visual Studio podemos cambiarlo, pulsando botón derecho, opción propiedades sobre WebApiSegura:



Ilustración 4.8 Configuración de la API

En nuestro ejemplo, vamos a trabajar en el puerto 49220 y ahora sabemos dónde cambiar la configuración de IIS Express para levantar nuestra WebAPI y tenerla funcionando.

Creando la estructura WebAPI

Vamos a crear la estructura del API

En nuestro explorador de soluciones realizar los siguientes pasos:

1. Sobre la carpeta controllers: botón derecho / agregar / controlador...
2. Se abrirá una nueva ventana de dialogo para elegir controlador.
3. Seleccionamos "Controlador WebApi 2 - en blanco".
4. Nos pedirá el nombre y lo llamaremos "LoginController".
5. Crear un segundo Controlador como los pasos 1,2,3.
6. Nos pedirá el nombre y lo llamaremos "CustomersController".
7. Dentro de carpeta Controller crear una nueva clase "TokenGenerator".
8. Dentro de carpeta Controller crear una nueva clase "TokenValidationHandler".
9. Dentro de carpeta Models crear una nueva clase "LoginRequest".

En este punto, ya hemos creado las clases para nuestro proyecto, evidentemente vacías, como vemos en la imagen:

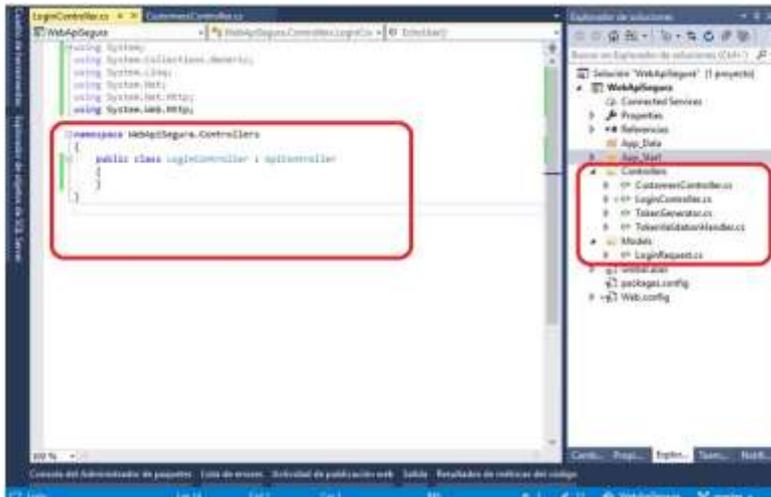


Ilustración 4.9 Estructura de la WebApi

Recordar: los controladores WebAPI heredan de "ApiController" y los controladores MVC de "Controller" en .NET Framework.

Implementación del código

Ahora, toca poner código a nuestras clases. Hay muchas formas de realizar esta implementación y esto es lo que más confusión genera entre los desarrolladores, en nuestro caso, trabajaremos con estas premisas:

LoginRequest: clase donde recibiremos las credenciales del usuario.

IHttpActionResult: para las respuestas HTTP StatusCode al cliente siguiendo filosofía RESTful.

[AttributeRoutes]: para decorar las rutas de los controladores del API en cada acción.

[Authorize]: Decorador para autorizar peticiones válidas al API (necesitará un JWT válido).

[AllowAnonymous]: Decorador para permitir peticiones anónimas al API (no necesitará un JWT)

web.config: Definimos los settings necesarios para nuestro Token JWT.

Comprobando que todo funciona

Una vez tenemos el código listo, vamos a verificar que realmente nuestra API funciona y hace lo que debe.

4.6 CREACIÓN DE TABLA PARA MOSTRAR LOS REGISTROS REALIZADOS

El encargado compartió un tutorial para poder realizar la creación de la tabla dinámica. [14]

4.6.1 Instalación

La biblioteca está disponible como paquete npm, por lo que todo lo que necesita hacer es ejecutar el siguiente comando:

```
npm install --save ng2-smart-table
```

4.6.2 Ejemplo de configuración mínima

Lo primero que debe hacer es importar las directivas ng2-smart-table en su componente.

```
import { Ng2SmartTableModule } from 'ng2-smart-table';
```

Luego regístrelo agregando a la lista de directivas de su módulo:

```
// ...
```

```
@NgModule({
```

```
  imports: [
```

```
    // ...
```

```
    Ng2SmartTableModule,
```

```
    // ...
```

```
  ],
```

```
  declarations: [ ... ]
```

```
})
```

Ahora, necesitamos configurar la tabla y agregarla a la plantilla. La única configuración requerida para que el componente comience a funcionar es una configuración de columnas.

Registremos la propiedad de configuración dentro del componente donde queremos tener la tabla y configuremos algunas columnas (documentación de configuración):

```
settings = {
```

```

columns: {
  id: {
    title: 'ID'
  },
  name: {
    title: 'Full Name'
  },
  username: {
    title: 'User Name'
  },
  email: {
    title: 'Email'
  }
}
};

```

Finalmente pongamos el componente ng2-smart-table dentro de la plantilla:

```

@Component({
  template: `
<ng2-smart-table [settings]="settings"></ng2-smart-table>
`
})

```

Actions	ID	Full Name	User Name	Email
Add New	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
No data found				

Ilustración 4.10 Tabla dinámica

Agregar datos a la tabla:

```

data = [
  {
    id: 1,
    name: "Leanne Graham",

```

```

username: "Bret",
email: "Sincere@april.biz"
},
{
id: 2,
name: "Ervin Howell",
username: "Antonette",
email: "Shanna@melissa.tv"
},
{
id: 11,
name: "Nicholas DuBuque",
username: "Nicholas.Stanton",
email: "Rey.Padberg@rosamond.biz"
}
];

```

Y pasar los datos a la tabla:

```

@Component({
template: `
<ng2-smart-table [settings]="settings" [source]="data"></ng2-smart-table>

```

Ahora tiene algunos datos en la tabla:

Actions	ID	Full Name	User Name	Email
Add New	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Edit Delete	1	Leanne Graham	Bret	Sincere@april.biz
Edit Delete	2	Ervin Howell	Antonette	Shanna@melissa.tv
Edit Delete	3	Clementine Bauch	Siemens	Neha@jweller.com
Edit Delete	4	Patricia Lebsack	Katherine	Juliana.DConner@koyf.net
Edit Delete	5	Cheryl Dalton	Kenneth	Lucio_Hettinger@annie.ca
Edit Delete	6	Ms. Dennis Schulze	Leopoldo_Corwey	Kateryn_Dach@james.org
Edit Delete	7	Kurtis Wiegand	Eliyn Skiles	Telly.Hooper@jolly.org
Edit Delete	8	Nicholas Runolfsdottir V	Marione_Herzog	Sherwood@rosamond.me
Edit Delete	9	Genieve Rasmussen	Delphine	Charm_McConnaughy@stan.eu
Edit Delete	10	Dorothea Ditzel	Morish Stantoni	Rey.Padberg@kiana.biz

Ilustración 4.11 Tabla dinámica con datos

5. ANÁLISIS DE LA EXPERIENCIA ADQUIRIDA

Durante mi estancia en las practicas profesionales, adquirí mucha experiencia en el área de desarrollo, muchas herramientas nunca las había utilizado y en algunos casos no sabía que existían. Me ayudaron creciendo profesionalmente y como desarrollador, utilicé lenguajes de programación nuevos, aprendí como funciona Angular junto con .net framework, el manejo de las bases de datos, a trabajar bajo presión y en equipo con otros desarrolladores, crear componentes, modelos, tablas inteligentes, método post, put y get, crear servicios y también el uso y creación de procedimientos almacenados.

Además de eso, pude observar como otros programadores con mucha experiencia trabajan y realizan sistemas en equipo.

5.1 ANÁLISIS GENERAL DEL SISTEMA, SU DISEÑO, DESARROLLO Y ORGANIZACIÓN.

El sistema realizado durante mi estancia fue de mucha ayuda para las agencias que lo utilizaron, los encargados de llenar las sentencias ahora pueden trabajar de una manera más eficiente, con mayor control y sin miedo a perder los datos. Al realizar el sistema adquirí muchos conocimientos que me ayudarán para mi futuro como profesional.

5.2 ANÁLISIS DE LOS OBJETIVOS DEL PROYECTO

Pude completar con éxito todas las actividades que me pidieron que realizara día a día, pude conocer como funciona la empresa para completar sus proyectos.

Se cumplieron con los objetivos mencionados en la sección 1.2, se podía registrar, modificar y ver sentencias registradas.

5.3 ANÁLISIS DE LAS ACTIVIDADES REALIZADAS

Las actividades que realice durante el transcurso de mi estancia profesional me ayudaron a obtener experiencia en el desarrollo de sistemas, manejo de bases de datos, en la parte del front-end y del back-end y a trabajar en equipo con otros desarrolladores. La mayoría de las actividades que realicé no tenía conocimiento alguno, pero fui adquiriendo el conocimiento diariamente, aparte recibía mucha ayuda de mí compañero de trabajo, el cual me ayudaba cuando lo necesitaba.

6. CONCLUSIONES Y RECOMENDACIONES

En conclusión, fue muy buena la experiencia que obtuve al realizar mis practicas profesionales en este lugar, aprendí muchas cosas de programación que no sabia que existían, aprendí el framework angular, aprendí el lenguaje typescript, el manejo de bases de datos y crear WebApis, cada día aprendía algo nuevo, todos los días la dificultad incrementaba lo cual me obligaba a buscar información en internet y a ser autodidacta. Pude reforzar los conocimientos adquiridos en la carrera, y aprendí nuevas tecnologías que en el mundo laboral se utilizan.

7. REFERENCIAS BIBLIOGRÁFICAS Y VIRTUALES

- [1] Anónimo. Diccionario Jurídico. Sentencia condenatoria.
Sitio web: <http://diccionariojuridico.mx/definicion/sentencia-condenatoria/>
- [2] Anónimo. Sentencia absolutoria | Justicia Penal. Imco.org.mx
Sitio web: <https://imco.org.mx/justiciapenal/blog/definicion/sentencia-absolutoria/>
- [3] Fiscalía Sonora. Fiscalia.sonora.gob.mx
Sitio web: <http://fiscalia.sonora.gob.mx/quienes-somos>
- [4] Anónimo. Desarrolladores de calidad. ¿Qué es Angular y para qué sirve?
Sitio web: <https://www.qualitydevs.com/2019/09/16/que-es-angular-y-para-que-sirve/>
- [5] Anónimo. Base de datos. Es.wikipedia.org
Sitio web: https://es.wikipedia.org/wiki/Base_de_datos
- [6] Muhammad Imran. SQL Server Management Studio – Una guía de instalación paso a paso.
Sitio web: <https://www.sqlshack.com/es/sql-server-management-studio-una-guia-de-instalacion-paso-a-paso/>
- [7] Anónimo. Docs.microsoft.com
Sitio web: <https://docs.microsoft.com/en-us/dotnet/framework/data/adonet/ef/overview>
- [8] Anónimo. Web Developers
Sitio web: <https://www.bls.gov/ooh/computer-and-information-technology/web-developers.htm/#tab-2>
- [9] Anónimo. Bootstrap - ¿Qué es y Cómo funciona? | Blog
Sitio web: <https://axarnet.es/blog/bootstrap>
- [10] Anónimo. Visual Studio Code - EcuRed
Sitio web: http://www.ecured.cu/Visual_Studio_Code
- [11] Anónimo. Modelo relacional de base de datos: elementos, cómo hacerlo, ejemplo – Lifeder. Salvavidas
Sitio web: <https://www.lifeder.com/modelo-relacional-base-datos/>
- [12] Anónimo. Angular - FormControl y FormGroup para crear formularios reactivos. Pociones de codificación
Sitio web: <https://codingpotions.com/angular-formularios>
- [13] Anónimo. Construyendo una Web API REST segura con JSON Web Token en .NET (Parte II)

Sitio web: <https://enmilocalfunciona.io/construyendo-una-web-api-rest-segura-con-json-web-token-en-net-parte-ii/>

[14] Anónimo. ng2-smart-table: biblioteca de tablas de datos angulares con funciones de clasificación, filtrado, paginación y agregar / editar / eliminar.

Sitio web: <https://akveo.github.io/ng2-smart-table/#/demo>

8. CARTA DE FINALIZACIÓN



Fiscalía General de Justicia del Estado de Sonora.

Hermosillo Sonora a 17 de Marzo de 2020

A QUIEN CORRESPONDA

Por medio de la presente hago de su constar que **Efrén David Lugo Cueva** con el número de expediente **215200590** de la carrera de Ingeniería en Sistemas de Información en la Universidad de Sonora llevó a cabo sus prácticas profesionales en la **Fiscalía General de Justicia del Estado de Sonora**, ubicada en Calle Pedro Moreno #49 esq. Manuel Z. Cubillas. Col. Las Palmas, C.P. 83260. En el periodo de 01 de diciembre al 01 de Febrero del presente año en el proyecto "BitPro" en el que estuvo durante una duración de **340 horas**.

Sin mas por el momento quedo a sus órdenes.

Atentamente

Karina Alejandra Sierra Gámez

Director de Innovación Tecnológica

Fiscalía General de Justicia del Estado de Sonora.



UNIVERSIDAD DE SONORA

COORDINACIÓN DIVISIONAL DE: Ingeniería Industrial

PRÁCTICAS PROFESIONALES

DEPARTAMENTO: Ingeniería Industrial

UNIDAD REGIONAL CENTRO CAMPUS HERMOSILLO

FPP-4

REPORTE FINAL DE ACTIVIDADES

Periodo: Del 01 / Diciembre / 2019 al 01 / Febrero / 2020

Cantidad de 340 Horas de un total de 340 Avance: 100 %

Nombre del practicante: Efrén David Lugo Cueva

Expediente: 215800590 Programa Educativo (Licenciatura): Ind. en sistemas

Nombre del Proyecto: BitPro

Datos de la Unidad Receptora (Razón Social): Fiscalía General de Justicia del Estado de Sonora

Responsable de la Unidad Receptora (Nombre/Puesto): Karina Alejandra Sierra Gómez

Contacto: Teléfono/UR: _____ Ext. _____ Celular: 667120 8146

DESCRIPCIÓN GENERAL DE ACTIVIDADES

En mi estancia en las Prácticas Profesionales desarrolle la parte de un sistema de registro. El sistema consiste en que varias personas pueden registrar sentencias y evitar el apeleo, lo cual se logra también necesitaban ver las sentencias y modificarlas en caso de ser necesario.

RETROALIMENTACIÓN

En caso de requerirse, anexar reportes, formatos, diagramas que apoyen las actividades realizadas.

Observaciones Generales:

Nombre y firma del alumno	Nombre y firma del tutor de prácticas profesionales Unison.	Nombre y firma del responsable del programa de la unidad receptora Sello de la UR

Original entregar en físico a Tutor de Prácticas Profesionales y Copia alumno.

Enviar en PDF al Coordinador Responsable de Prácticas Profesionales de la carrera.