**MODELO DE FORMACION GRADUAL**

**“NOTAS”**

**“Módulo 2. Herramientas de Desarrollo de Software”**

**“Parte 1. Microsoft Visual Studio y Visual Studio Code”**

**Datos:**

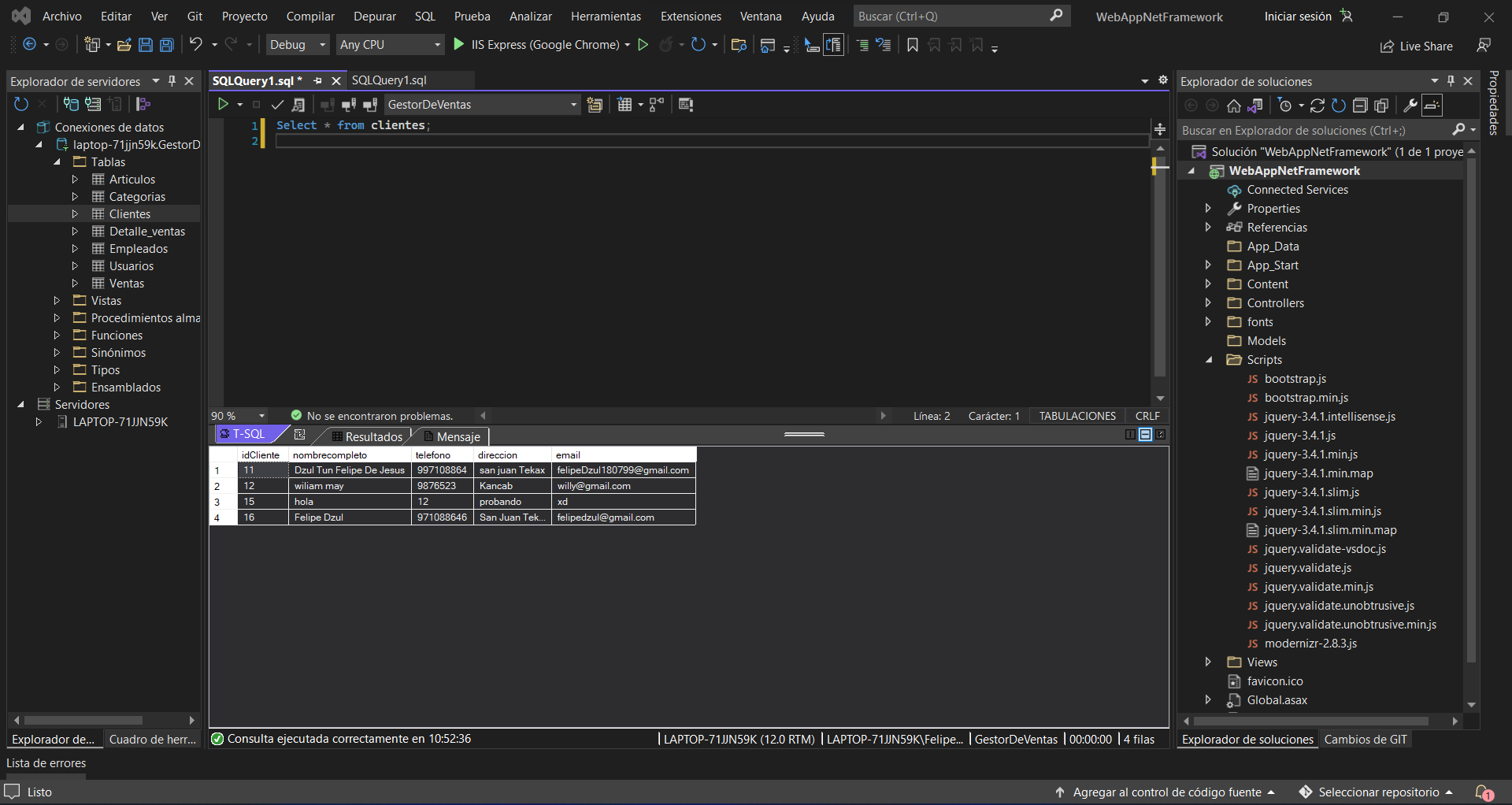
|  |  |  |  |
| --- | --- | --- | --- |
| **Plan de Estudios:** | **Ingeniería en Sistemas Computacionales** | **Grupo (s):** | **8AVS** |
| **Nombre del alumno(s):** | **Dzul Tun Felipe de Jesús** | **Matrícula:** | **181T0099** |
| **Coordinadora de Aprendizaje** | **Laura Marcela Flores Lima** | **Fecha de Edición:** | **29 / 03 / 22** |

**Unidad 1. Microsoft Visual Studio**

Nos permite crear sitios y aplicaciones web en cualquier entorno que soporte la plataforma .NET. Con esto se puede crear aplicaciones entre estaciones de trabajo, paginas web, dispositivos móviles, consolas, entre otros.

Lenguajes de programación: C++, C#, F#, Java, Visual Basic, Python, Ruby, .Net, PHP.

Entornos de Desarrollo Web: ASP.NET y ASP.NET MVC.



**VENTANAS**

* EXPLORADOR DE SOLUCIONES:

Es un área del entrono de desarrollo integrado que contiene su solución y ayuda a administrar los archivos de proyecto. Los archivos se muestran de maneja jerárquica (categorías, niveles), al estilo del explorador de Windows (Carpetas), carpetas y soluciones.

* Proporciona una vista organizada de los proyectos y sus archivos.
* Permite ver elementos y realizar tareas de administración de los elementos en una solución en un proyecto.
* EXPLORADOR DE SERVIDORES:

El explorador de servidores es una consola de administración. Esta ventana se utiliza para abrir conexiones de datos, iniciar sesión en los servidores, explorar bases de datos y ver servicios de sistema.

Para tener acceso al Explorador de servidores, elija **Explorador de Servidores** en el menú **Ver**.

Nodos Primarios 🡪 Conexiones de Datos (local Agregadas), Servidores (Disponibles), Microsoft Azure (Servidores Disponibles de Azure).

* LISTA DE ERRORES:

Mostrar **errores, advertencias y mensajes** que se generan cuando se edita y compila código.

Errores de sintaxis por IntelliSense.

Ver el archivo donde se produce el problema y desplazarse a la ubicación del error y filtrar entradas.

* VENTA DE PROPIEDADES:

Propiedades de un objeto seleccionado (Boton, listbox, etc).

* VENTANA DE INMEDIATO:

Utilizada para depurar y evaluar expresiones, ejecuta instrucciones, imprimir variables, etc.

Permite escribir expresiones del lenguaje de programación. La ventana Inmediato también admite IntelliSense.

Se ve solo cuando se está depurando.

* JERARQUIA DE LLAMADA:

Permite navegar por el código mostrando todas las llamadas a y desde un método, propiedad, o un constructor seleccionado. Permite entender como fluye el código (impacto en otros métodos).

* COMPILACION:

Para compilar aplicaciones y crear ensamblados y programas ejecutables con frecuencia durante un ciclo de desarrollo.

Métodos de compilación: Se puede compilar una aplicación mediante las opciones de compilación predeterminados en el IDE, en un símbolo del sistema o mediante Azure Pipelines.

* IDE (multiprocesador de proyecto C# y C++).
* Línea de comandos de MSBuild.
* Azure Pipelines.
* PUBLICACION:

Se utiliza el asistente para publicar con el fin de especificar la ubicación donde se publicará la aplicación. Puede publicar la aplicación en un sitio Web, en un servidor FTB o en una ruta de acceso de archivo. (Azure, Folder, FTP/FTPS Server, Web Server (IIS)).

* DEPURACION:

Permite observar el comportamiento del programa en tiempo de ejecución y detectar problemas.

Se puede interrumpir la ejecución del programa para examinar el código, evaluar, modificar variables, entre otras funciones.

**Puntos de Interrupción:** Detienen la ejecución del depurador para evaluar y examinar. (F9)

Condición de puntos de interrupción: condiciones le permite controlar cuándo y dónde se ejecuta un punto de interrupción.

* VENTANAS DE INSPECCION:

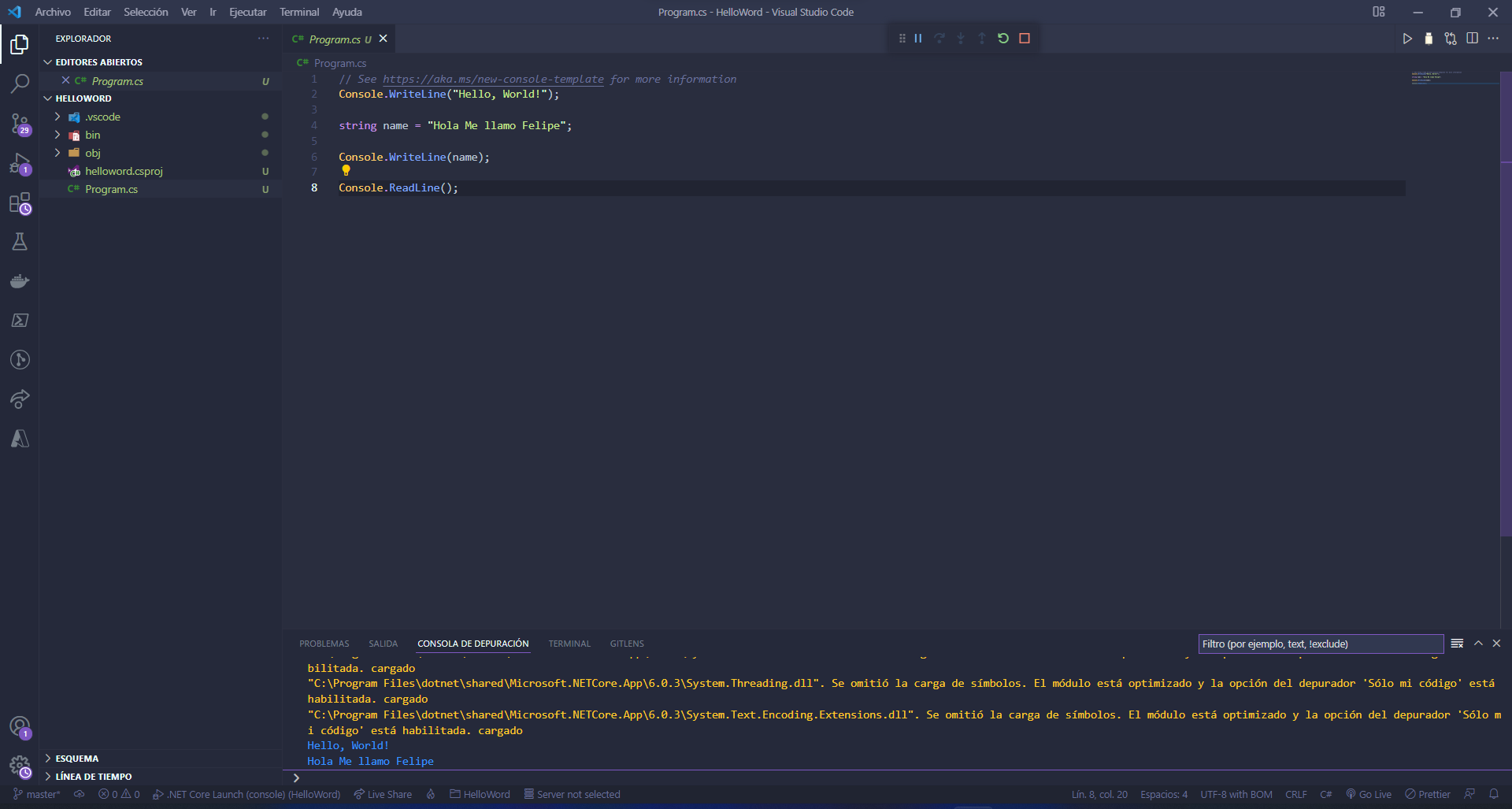
Para observar las variables y expresiones durante una sesión de depuración. La diferencia es que la ventana Inspección puede mostrar varias variables, mientras que la ventana Inspección rápida muestra las variables de una en una.

Se puede hacer cambios de valores de las variables.

**Unidad 2. Microsoft Visual Studio Code**

Visual Studio Code es un editor de código fuente ligero pero potente que se ejecuta en su escritorio y está disponible para Windows, MacOS y Linux.

Viene con soporte integrado para JavaScript, TypeScript y Node.js y tiene un completo ecosistema de extensiones para otros lenguajes (como C++, C #, Java, Python, PHP, Go) y tiempos de ejecución (como .NET y Unity).



* IntelliSense:

Visual Studio Code utiliza un resaltado de sintaxis y autocompletar con IntelliSense, que proporciona terminaciones inteligentes basadas en tipos de variables, definiciones de funciones y módulos importados.

* DEPURACION:

Depure el código directamente desde el editor.

Inicie o adjúntelas a las aplicaciones en ejecución y depure con puntos de interrupción, pilas de llamadas y una consola interactiva.

Nota: La depuración de código no está disponible para .NET Framework.

* COMANDOS DE GIT INTEGRADOS:

Trabajar con Git y otros proveedores de SCM nunca ha sido tan fácil. Revise los diffs, organice los archivos y realice confirmaciones directamente desde el editor. Empuje y extraiga desde cualquier servicio SCM alojado.

* EXTENSIBLE Y PERSONALIZABLE:

Instale extensiones para agregar nuevos idiomas, temas, depuradores y para conectarse a servicios adicionales.

Las extensiones se ejecutan en procesos separados, lo que garantiza que no ralentizarán su editor.

* MICROSOFT AZURE:

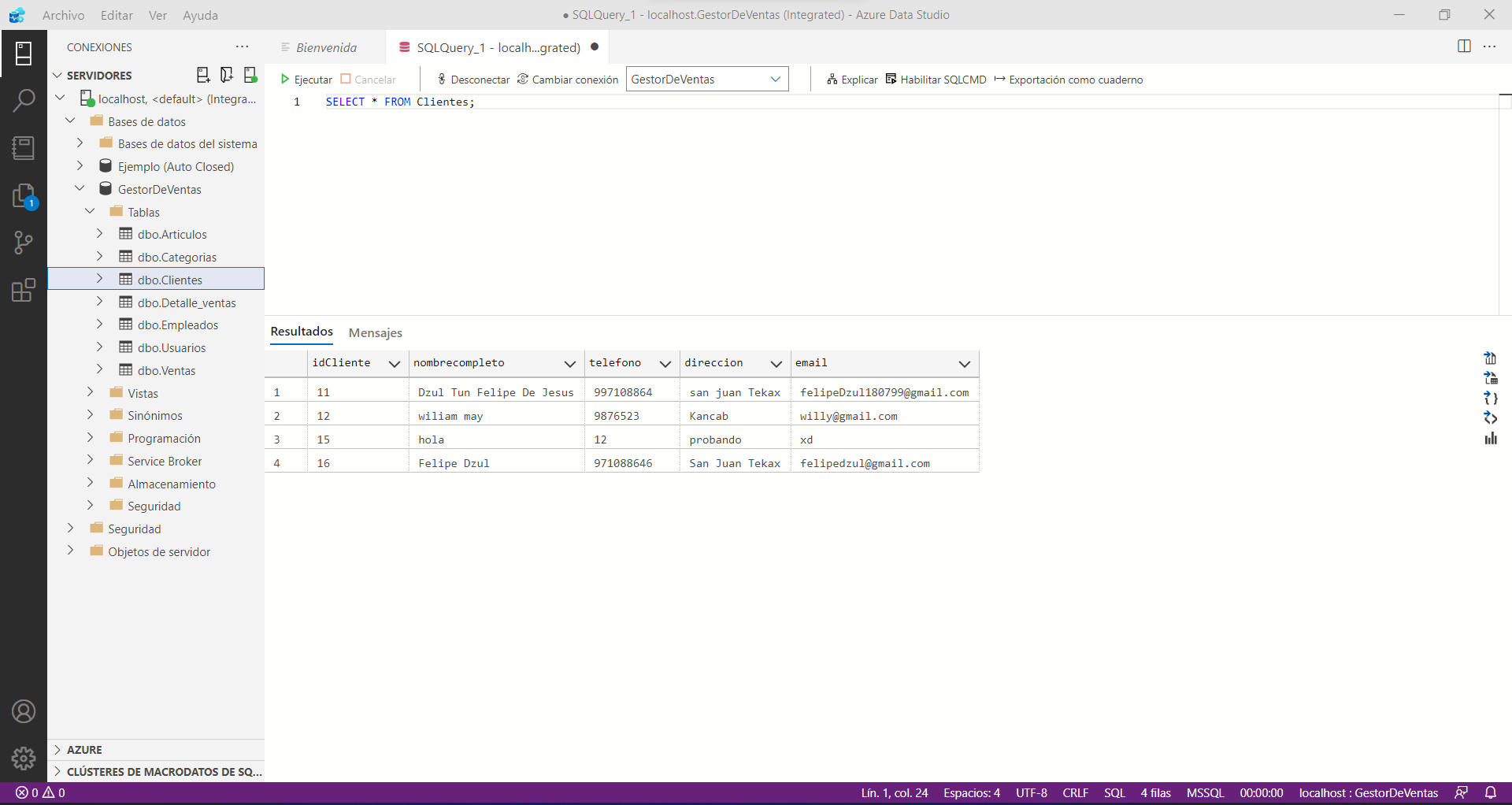
Con Microsoft Azure puede implementar y alojar sus sitios React, Angular, Vue, Node, Python (¡y más!), almacenar y consultar datos relacionales y basados en documentos, y escalar con computación sin servidor, todo con facilidad, todo desde VS Code.

**Unidad 3. Azure Data Studio**

Azure Data Studio es una herramienta de base de datos multiplataforma para profesionales de datos que usan plataformas de datos locales y en la nube en Windows, macOS y Linux.

Azure Data Studio ofrece una experiencia de editor moderna con IntelliSense, fragmentos de código, integración de control de código fuente y un terminal integrado.

Está diseñado teniendo en cuenta al usuario de la plataforma de datos, con el gráfico incorporado de conjuntos de resultados de consultas y paneles personalizables.



* IntelliSense:

Ofrece una experiencia de codificación SQL moderna y centrada en el teclado que facilita las tareas diarias con características integradas, como ventanas de varias pestañas, un editor SQL enriquecido, IntelliSense, finalización de palabras clave, fragmentos de código, navegación de código e integración de control de código fuente (Git).

* ETENSIBILIDAD Y CREACION DE EXTENSIONES:

Mejore la experiencia de Azure Data Studio ampliando la funcionalidad de la instalación base.

Azure Data Studio proporciona puntos de extensibilidad para las actividades de administración de datos y compatibilidad con la creación de extensiones.

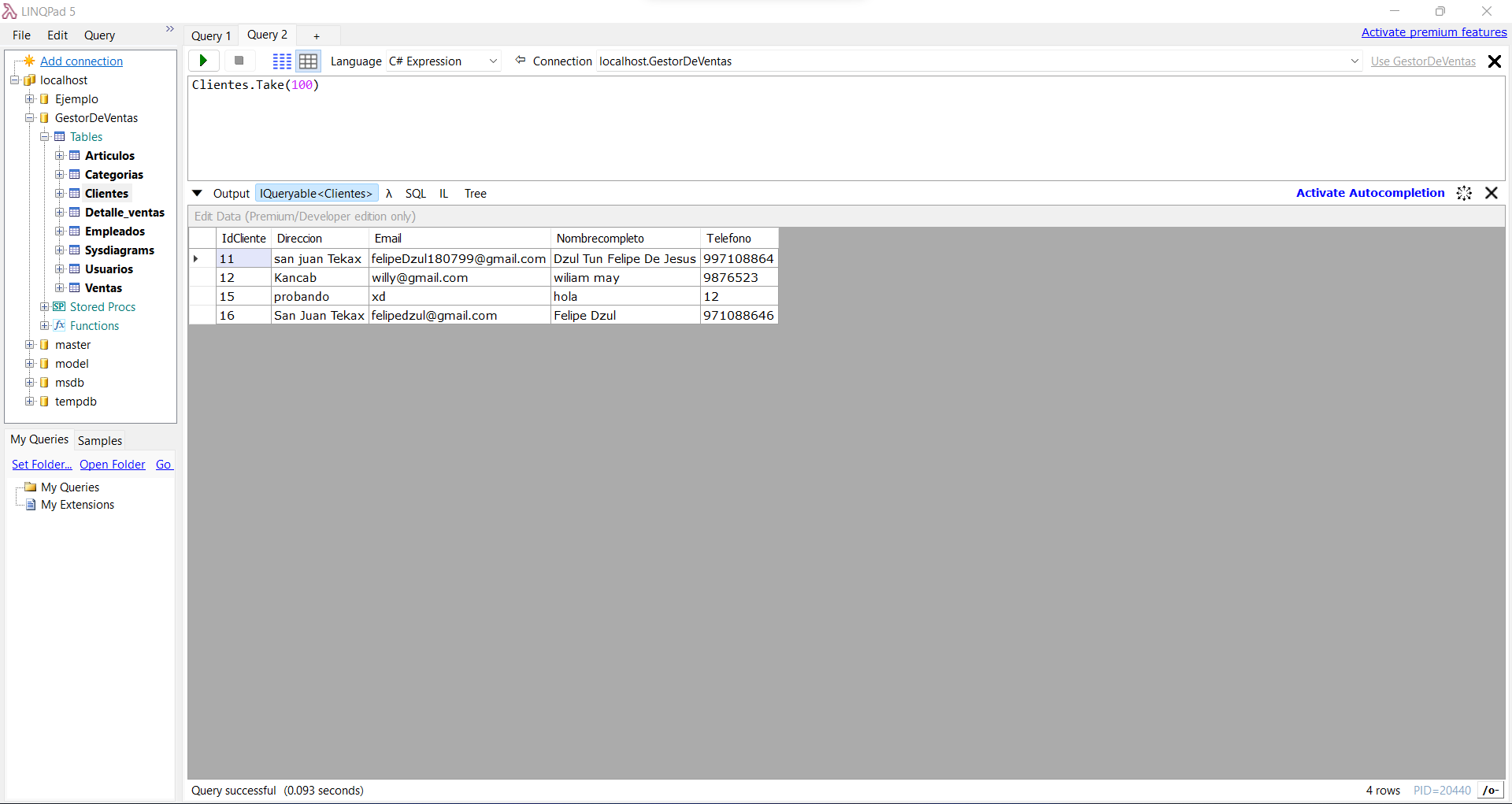
**Unidad 4. LINQPad**

LINQPad es una utilidad de software dirigida al desarrollo de .NET Framework y .NET Core.

Se utiliza para consultar de forma interactiva bases de datos SQL (entre otros orígenes de datos como OData o WCF Data Services) utilizando LINQ, así como para escribir código C# de forma interactiva sin necesidad de un IDE.

Esto amplía su uso a un “ambiente de pruebas” donde el código de C# se puede crear rápidamente prototipos fuera de Visual Studio.

También se puede utilizar para escribir código en los lenguajes VB.NET, SQL y F#.



* LINQ:

Con LINQPad, puede consultar de forma interactiva bases de datos en un lenguaje de consulta moderno: LINQ.

LINQPad incluye un motor optimizado para crear contextos de datos tipeados sobre la marcha, e incluye controladores y formateadores para:

* SQL Server, SQL Azure, SQL CE, Oracle, SQLite, PostgreSQL y MySQL.
* Azure Table Storage y Metadatos de Azure.
* Contextos personalizados de Entity Framework Core.
* OData y SharePoint entre otros.

**Unidad 5. Postman**

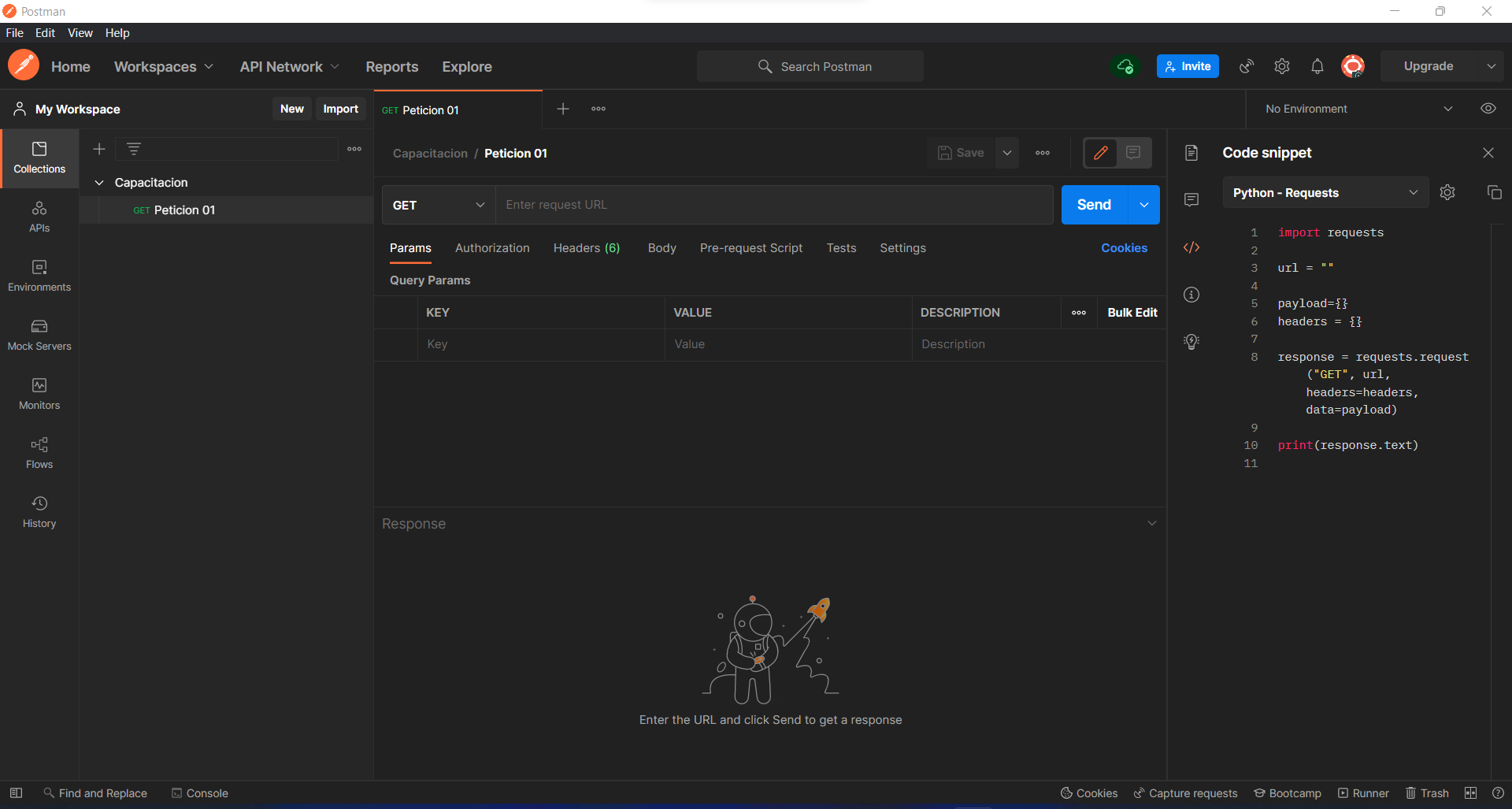
Es una aplicación que nos permite realizar pruebas de API y API REST.

Es un cliente HTTP que nos da la posibilidad de probar ‘Peticiones’ a través de una interfaz gráfica de usuario, por medio de la cual obtendremos diferentes tipos de respuesta que posteriormente deberán ser validados.

Postman es una plataforma API para crear y usar API. Postman simplifica cada paso del ciclo de vida de la API y optimiza la colaboración para que pueda crear mejores API, más rápido.

Postman surgió originariamente como una extensión para el navegador Google Chrome.

Hoy en día dispone de aplicaciones nativas para MAC y Windows y están trabajando en una aplicación nativa para Linux.



* COLECCIÓN DE APIs:

Las colecciones son carpetas donde se almacenan las peticiones y que permiten ser estructuradas por recursos, módulos o como el equipo lo desee.

* DOCUMENTACION DE LA API:

Haga de la documentación una parte central de su flujo de trabajo de API con las funciones de documentación automática de Postman.

Postman admite documentación habilitada para markdown y legible por máquina a través del formato Postman Collection, y también puede generar documentos a través de sus archivos OpenAPI.

Sus documentos incluirán automáticamente detalles sobre sus solicitudes, con código de muestra en varios lenguajes cliente.

* GENERACION DE CODIGO DE CLIENTE:

Una vez que tenga su llamada a la API justo donde la desee, puede usarla para generar fragmentos de código en su lenguaje o framework de elección.

Esos fragmentos se pueden copiar y pegar desde Postman a sus aplicaciones front-end y back-end.

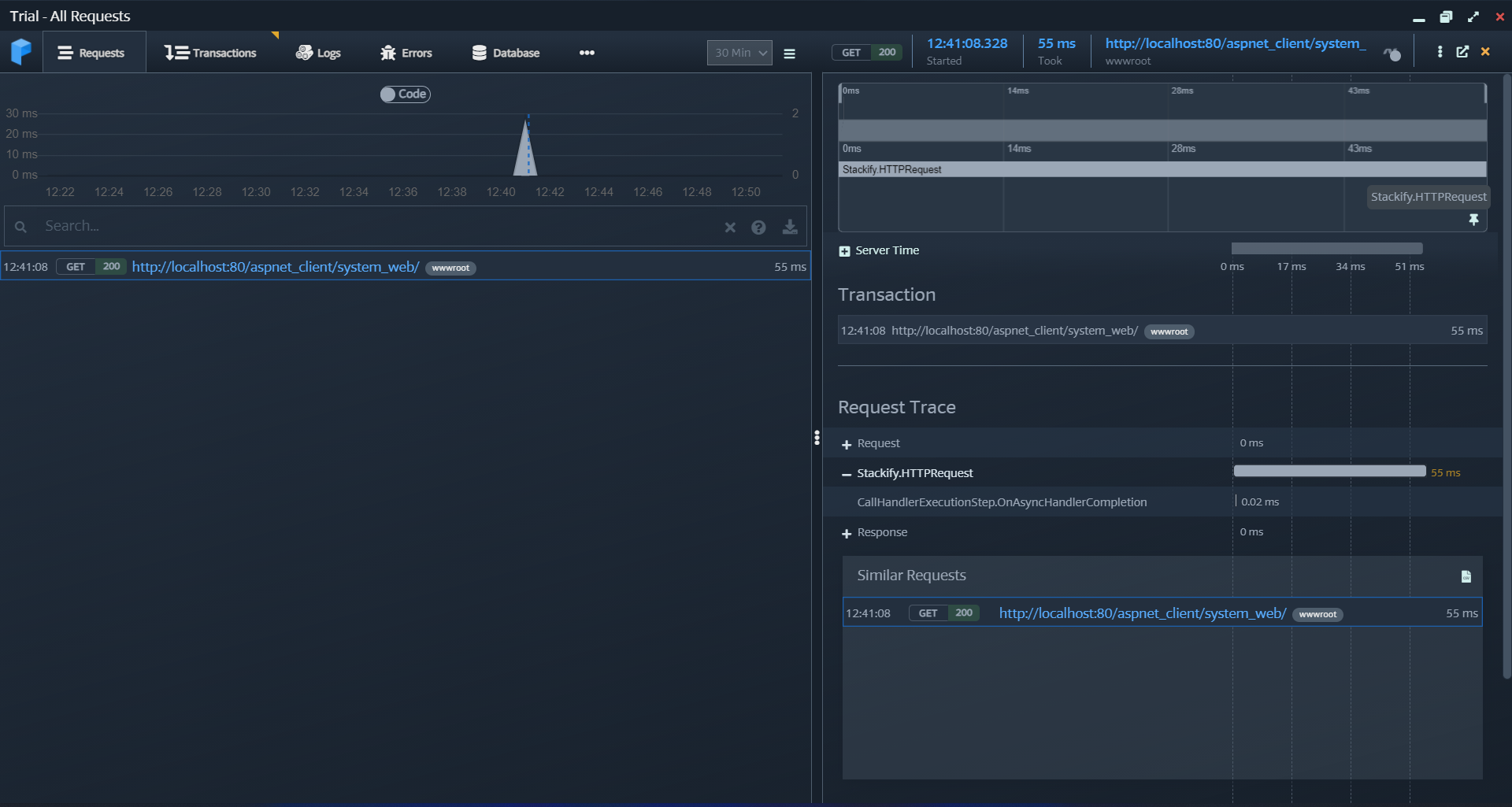
**Unidad 6. Prefix**

IMPORTANCIA DE UTILIZAR UNA HERRAMIENTA DE MONITOREO DE APLICACIONES.

* Mejorar la calidad del software que se está desarrollando.
* Descubrir problemas al principio de las etapas de desarrollo.
* Evitar o disminuir el número de errores en el entorno de producción durante el uso del software.
* Conocer el comportamiento de la aplicación en detalle cuando se está desarrollando.
* Durante las tareas de desarrollo o soporte de aplicaciones, los desarrolladores deben tener una vista detallada del comportamiento del código fuente cuando la aplicación se está ejecutando. Esto es fundamental para aumentar la calidad del producto y también para facilitar las tareas de soporte técnico.
* En un enfoque de desarrollo ágil, esto se puede utilizar para mejorar la calidad del software en cada iteración de desarrollo.

Prefix es una aplicación de generación de perfiles nativa, Prefix es local.

Toma el mismo perfil potente pero ligero que se ofrece en servidores de producción con **Retrace** se puede conectar a un servidor externo, lo pone a disposición mientras escribe y prueba su código en su estación de trabajo de desarrollo.



¿QUE PUEDO HACER CON PREFIX?

Prefix se ejecuta en segundo plano y escucha las solicitudes mientras ejecuta las aplicaciones.

Captura todas las solicitudes web y proporciona rastros detallados o instantáneas de lo que hicieron las solicitudes.

* Solucionar problemas de solicitudes web lentas.
* Buscar consultas SQL lentas.
* Buscar excepciones ocultas.
* Busque patrones de codificación incorrectos como consultas de base de datos N + 1.

ESCRIBIR UN MEJOR CODIGO:

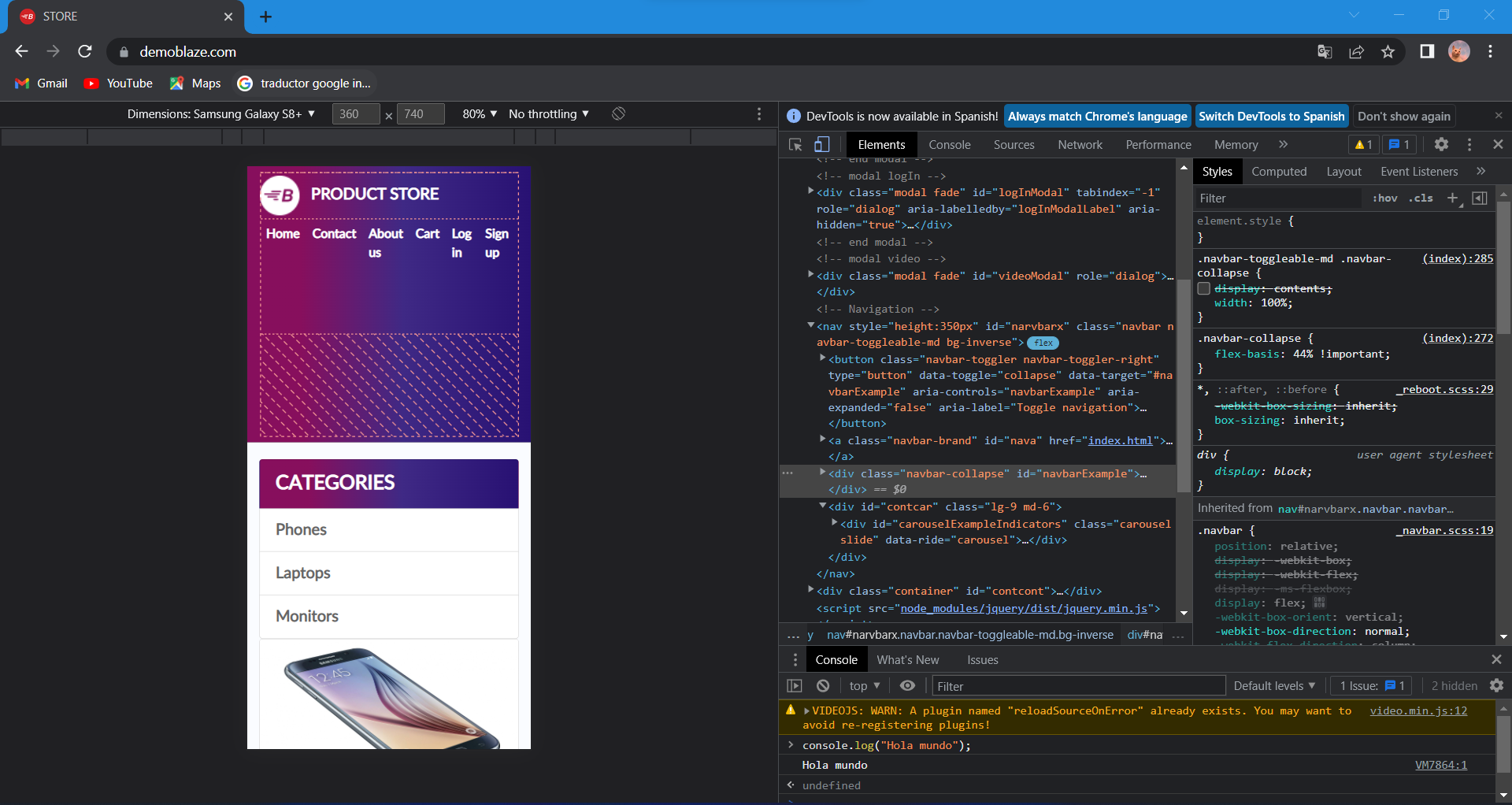
Con **Stackify**, los desarrolladores validan el comportamiento de su código en su propia máquina mientras lo escriben.

Esto permite a los equipos enviar más y arreglar menos porque adoptan las pruebas lo antes posible y con la mayor frecuencia posible.

**Unidad 7. Chrome Devtools**

Chrome Devtools es un conjunto de herramientas de desarrollo web integradas directamente en el navegador Google Chrome.

Devtools puede ayudarlo a editar páginas sobre la marcha y diagnosticar problemas rápidamente, lo que en última instancia lo ayuda a crear mejores sitios web, más rápido.



* SIMULADOR DE DISPOSITIVOS MOVILES:

Usa el modo de dispositivo para aproximar el aspecto y el rendimiento de tu página en un dispositivo móvil.

Modo de dispositivo es el nombre de la colección suelta de funciones de Chrome DevTools que te ayudan a simular dispositivos móviles.

Estas características incluyen:

* Simulación de una ventana gráfica móvil
* Limitación de la red
* Limitación de la CPU
* Simulación de geolocalización
* Orientación de la configuración
* PANEL DE ELEMENTOS:

El panel Elementos de la Herramienta de desarrollo le permite ver todo en un árbol DOM y permite la inspección de los elementos DOM.

A menudo podemos visualizar la pestaña Elementos cuando necesitamos identificar el fragmento de código HTML para un aspecto particular de la página.

Los elementos incluyen el estilo calculado, los estilos, las reglas CSS coincidentes, las métricas, las propiedades, los puntos de interrupción DOM y los detectores de eventos.

* CONSOLA:

Los desarrolladores web a menudo registran mensajes en la consola para asegurarse de que su JavaScript funciona como se esperaba.

Para registrar un mensaje, inserte una expresión similar en su JavaScript.

Cuando el navegador ejecuta su JavaScript y ve una expresión como esa, sabe que se supone que debe registrar el mensaje en la consola.

* PANEL DE RED:

Algunas de las funciones que podemos realizar son las siguientes:

* Buscar solicitudes de red por mensaje de texto.
* Buscar solicitudes de red por expresión regex.
* Filtrar (excluir) las solicitudes de red.
* Usar el filtro de propiedades para ver las solicitudes de red de un determinado dominio.
* Buscar solicitudes de red por tipo de recurso.