

Introducción

Sobre el autor.

Soy Ing. en Sistemas con Master en Comercio Electrónico, fundador de Juegos Mentales Online, una plataforma que genera ingresos compartiendo retos y desafíos para los perfiles sociales de ciertas empresas, con clientes como Malta Morena, Perfumuda y otros.

CEO de MersyRD, una solución en línea (SAAS) de expedientes médicos para clínicas, hospitales y doctores independientes que ha sido implementada, además de usada por un gran numero de médicos, en el Hospital General Dr. Vinicio Calventi y de forma modular en el Darío Contreras.

Creador de **QuestionaryApp**, una aplicación de cuestionarios privados para profesores y agencias de contrataciones. **Notes & Expenses Tool**, Una app para administrar notas, notificaciones, agenda, gastos e ingresos y **única aplicación del mundo** capaz de convertir una nota (extensa) en imágenes, ideal para influencers y personas que quieren convertir sus textos extensos en fotos para el IG. **Torneo Predicciones**, una app móvil desarrollada en Xamarin con el fin de desafiar a tus amigos intentando acertar predicciones y ganar puntos en el proceso (Plagio a la clara (pero mejorada) del proyecto Rusia App del maestro Zulu). Entre otras aplicaciones que pueden auditar en GitHub.

Analista programador en Dirección General de Impuestos Internos, profesor en Universidad Dominicana Organización y Método, así como colaborador docente en la UASD.

Además de tener 10 años de experiencia en el desarrollo a la medida de soluciones informáticas, páginas web y consultoría para diversas instituciones y pequeñas empresas en PraySoft Dominicana y Atagua Soft.

Así como también colaboración en varios proyectos de comunidad.

¿Qué Vamos a Ver?

En este curso vamos a trabajar con ASP .Net Core, usando el marco de trabajo MVC, escogemos Mvc NetCore y no Mvc Legacy, porque además de tener varios años de desfase el segundo, los conocimientos que adquieras en netCore, te sirven con ligeros cambios en Legacy, tomando en cuenta que también en Core vas a tener un desarrollo con la posibilidad de correr en Linux de esa forma ahorrar costos en servidores :D.

La única diferencia radical que tendríamos entre. netCore y Legacy, serían las inyecciones de dependencias y las configuraciones, cuando lleguemos a esos tópicos, me comprometo a enseñarles cómo se hacen en ambos, ya que cuando ustedes salgan a la calle se van a topar con muchas aplicaciones hechas en MVC Legacy que no van a ser migradas en mucho tiempo y necesitamos saber cómo meterles mano si nos topamos con algunas.

La orientación de este curso es enfocada a convertirnos en Fullstack Developers, a pesar de que nos enfocaremos en construir una solución orientada al Backend, pero hay que ser realistas, todos sabemos que en el mundo real las empresas pequeñas con las cuales usted puede conseguir una «picota», no andan contratando un desarrollador para el FrontEnd y otro para el BackEnd, normalmente quieren uno que «meta mano» en todo, **si bien es cierto que es bueno especializarse**, la idea de este curso es que usted salga de acá a meter mano y a ganar dinero, por lo tanto, aunque usted no se debería volver un master en diseño, al menos debería conocer lo básico como:

- ¿Qué es CSS?
- ¿Cuáles reglas básicas de usabilidad debe seguir una aplicación?
- ¿Cuáles Frameworks existen para facilitar tu trabajo y hacerlo ver como algo profesional?

Y eso no es solo una mentalidad tercermundista de Latino América, eso pasa en todos lados del mundo, solo las grandes empresas se pueden dar el lujo de contratar gente especializada, aunque normalmente esas solo sub contratan el trabajo. Hay que tener en cuenta que no todas las empresas son como el sueño dorado de Microsoft, Alphabet (Google), Apple o Amazon.

Pero, aun cuando usted logre llegar a una empresa grandes ligas, en un lugar que contrate gente especializada, ¿imagine que usted no sepa nada de lo que es Css? Una cosa es que usted no tenga la vena aquella que se le atribuye a quienes tienen el arte para el diseño, pero si usted no sabe lo básico de estilo, no se va a entender nunca con el diseñador, o si no sabe lo básico de arquitectura no se va a llevar bien con el encargado de infraestructura, por eso la orientación del curso será a meter mano con todo lo que involucra la creación de un proyecto completo desde cero.

Ojo y nota aclaratoria, así como disclaimer, **este no es un curso de programación básica**, ni de C Sharp, a pesar de que la forma en que se explica es muy básica, partiendo técnicamente desde cero y de hecho es la forma que me ha funcionado con mis estudiantes de clases particulares, es bueno que conozca al menos de manera esencial lo que es C# y conocimientos simples de programación orientada a objetos, en caso de ser necesario se impartirá de manera gratuita un curso desde cero que te ayude a seguir el ritmo de las clases.

Algunos de los tópicos que vamos a tocar son:

- Aprendiendo a levantar requerimientos.
- Trabajar ordenado y con control de versiones.
- Construyendo interfaces de usuario.
- Contrayendo APIs.
- Programación Orientada a Objetos de verdad.
- Arquitectura Limpia.
- Manipulación de datos con Entity Framework Core.
- Manipulación de objetos con Linq.
- Servicios de Azure.
- Aplicaciones multi Usuario Seguras.
- Aplicaciones multi Owner (saas).
- Entity Filters.
- Patrón Repositorio y Unidad de Trabajo de Forma Correcta.

Todo esto y muchos conceptos más, los iremos viendo y aprendiendo en el contexto de una aplicación del mundo real, partiendo desde un documento de requerimientos, extrayendo los casos de uso y el análisis de cómo implementar estos.

Requisitos

- Conocimientos básicos de C#.
- Conocimientos básicos de POO.
- Conocimientos de modelado y estructura de datos.
- Visual Studio Community >= 2017.
- Net Core >= 2.2 (yo usaré el 3.1, pero son compatibles)
- Sql Server Managment Studio >= 14.5

Nosotros no vamos a profundizar en aquellas herramientas que no tengan una implicación directa con la creación de nuestro programa, pero debemos conocer la mayoría de ellas al menos por arribita, corresponderá a la tutoría personalizada la parte de profundización sobre algún tema x que no tenga que ver con el desarrollo de nuestra aplicación o investigarlo por su cuenta, sin embargo siéntase libre de preguntar que yo no muerdo y soy muy abierto a los comentarios y debates, si hay algo de la clase que no les gusta lo podemos cambiar o algo que se necesite reforzar lo podemos debatir.

Nuestra aplicación, a pesar de ser de consumo, asumo que ustedes conocen la diferencia entre aplicaciones de consumo y empresariales, la vamos a enfocar a cómo funciona el mercado en la calle a nivel empresarial, el cual es bastante distinto a como nos enseñaron en la universidad o como dictan la mayoría de los cursos que hay en las diversas plataformas.

Nuestro proyecto final, será **un clon de MersyRD, es decir, una solución de manejo de expedientes médicos para Clínicas, Hospitales y Centros de Atención Primaria, el mismo tendrá el módulo base de Expedientes digitales (recetas, historia general, afecciones y CIE10), un Punto de Venta (que solo saliendo por ahí a vender este único modulo ya recuperan la inversión que hicieron en este curso), Gestión de Citas y Control de Turnos, siendo un sistema multi usuario, pero a la vez multi empresa**, aunque los conocimientos que vamos a adquirir aquí, nos pueden servir para hacer cualquier tipo de aplicación del mundo real, incluso una red social en .net completa que le haga la competencia a Facebook, por cierto, no lo intente, Google no pudo incluso con todos los millones de los que disponía.

En este curso te llevaremos de la mano sin importar si eres un desarrollador junior, novato o alguien que apenas está incursionando en el mundo de la

programación, total, yo soy #JrForEver y quien mejor que un junior para acompañarte en el camino y comprender por lo que estas pasando, iremos desde lo más básico hasta lo más avanzado para que mejores tus habilidades de desarrollo.

Iniciaremos conociendo la plantilla básica de netCore, como está estructurada y cómo funciona el esqueleto, para posteriormente ir transformándola en nuestra aplicación.

La aplicación a desarrollar

Mersy Core: Es un sistema de expedientes clínicos para que los médicos de una institución o de su consulta privada, **lleven el registro clínico de sus pacientes**, los médicos o su personal, pueden **registrar sus horarios de disponibilidad**, los pacientes pueden **agendar citas** indicando que van a asistir a una consulta, esta **se agenda en su calendario**, debe enviarse **notificaciones al paciente o el doctor en caso de cambios** de parte y parte, en la cita, el turno se maneja mediante un sistema de control de turnos y **la consulta puede ser facturada** como una **orden** o **factura**, donde **los servicios pueden ser multi-precios dependiendo de la ARS**, facturas que pueden ser **cobradas, condonadas, o financiadas**, financiamiento que puede ser **saldado o abonado** hasta ser saldado.

¿Qué aprenderemos?

Para desarrollar nuestra aplicación veremos algunos elementos del desarrollo web moderno, tales como:

- Bootstrap.
- Estilos.
- Usabilidad.
- Javascript.
- Ef code first.
- RESTful APIs.
- Seguridad.
- Plain Old CLR Object.
- Object-oriented design.

Y lo que vayamos necesitando en el camino según lo requiera la trama.

Nosotros vamos a iniciar con tópicos básicos, quiero que esa parte quede bien clara, por tanto, el código inicial no estará optimizado y será la vergüenza de los dioses del software, pero a medida que vayamos avanzando iremos evolucionando el mismo a uno más limpio, mantenible y elegante, que hable por sí mismo con solo verlo sin necesidad de tener que añadir comentarios adicionales, práctica que se conoce como código suficientemente descriptivo o auto descriptivo.

Te garantizo que con esta guía vas a poder llevar tus habilidades en el desarrollo .NET al siguiente nivel.

Estructura del Curso.

La estructura del curso será impartida o se divide en seis partes:

- **Fundamentos para tirar código chapucero en .Net Core.**
- **Tópicos Avanzados.**
- **Patrones y Arquitectura.**
- **El porqué de las cosas.**
- **Vainita rara que debemos aprender a hacer.**
- **Terminado el Proyecto.**

Fundamentos para tirar código en .Net Core.

En la primera parte abarcaremos los tópicos tales como, ¿por dónde carajos arranco?

Por eso en esta parte iniciaremos con el documento de requerimientos, te enseñare a extraer los principales casos de uso, que tienen un mayor impacto en el desarrollo de nuestra aplicación y luego la construiremos paso a paso.

El enfoque de esta parte es enseñarte la mentalidad del ingeniero de software, pensar como lo haría un ingeniero de software, como abordar un problema, romperlo en problemas más pequeños y resolverlos paso a paso. De igual forma tocaremos los cimientos del desarrollo de software Backend y Frontend, usando y aprendiendo en el camino.

Entity Framework Code First, Bootstrap, validaciones de datos, seguridad, aspectos artísticos de interfaz de usuario para crear interfaces amigables, etc.

Tópicos Avanzados.

En la segunda parte, implementaremos servicios.

En esta parte haremos bastante énfasis en el diseño orientado a objetos, evitar la invalidez de un objeto, construir APIs y valernos de Bootstrap para construir aplicaciones modernas.

Patrones y Arquitectura.

En esta parte veremos un poco de arquitectura y patrones de diseño implementando las mejores prácticas.

El porqué de las cosas.

Ya con los conocimientos de como haber construido un par de módulos básicos de una aplicación funcional lo suficientemente eficiente como para salir a hacer dinero o colgarla en tu GitHub para buscar empleo en el área y con la emoción de que tienes los conocimientos necesarios para meter mano, vamos a interrumpir el desarrollo de nuestra app, para explicarte desde cero como funciona la arquitectura MVC, como funciona por dentro NetCore y como funciona el modelo cliente-servidor.

La razón de poner esto hasta acá y no al inicio, es que a mi juicio y según lo que he podido apreciar en mis años como docente, es que estos módulos son la principal barrera que impiden a algunos potenciales desarrolladores arrancar. Si bien es bueno saber cómo funciona la estructura del microondas y como este calienta las moléculas de agua en el interior de la comida para calentarla, entiendo que primero debemos aprender cómo funciona el microondas, aprender a calentar la comida y una vez tenemos el hambre saciada me siento a aprender como funcionan las ondas de radio.

Vainita rara que debemos aprender a hacer.

Retomando nuestro proyecto, vamos a aprender a hacer ciertas cosas que, aunque no nos servirán de nada en nuestro proyecto como tal, son tópicos que se espera que un desarrollador de alto nivel tenga conocimiento, este módulo es una especie de snippet list de aquellos elementos que no hayamos visto durante el curso.

Terminado el Proyecto.

En esta última parte culminaremos todos los módulos restantes, ya de una forma más directa sin explicar gran cosa más que la razón por la cual tomamos tal o cual decisión, porque implementamos tal código de x forma y no de h, aquí explicaremos en que momentos es bueno hacer una mala practica en aras de un código más legible o entendible para otros programadores, cuando optimizar y cuando enfocarnos.

¡¡¡¡Vamo' a Meté Mano!!!!

Te recomiendo que aunque creas que sabes demasiado de lo básico, no te saltes directo a la siguiente parte, porque créeme que hay algunos truquitos que quizá no conozcas y que te serán de mucha ayuda en tu desarrollo profesional, en la primera parte es donde explicamos todos los conceptos, por lo que en la segunda y tercera parte ya nos vamos a enfocar directamente en trabajar sin explicar mucho a no ser que sea algo nuevo que no hayamos implementado antes.

Otra cosa de vital importancia para el aprendizaje, no importa que tan básico consideres que sean los ejercicios, te recomiendo encarecidamente que intentes resolverlos por tu cuenta, intenta realizarlos y después consulta la respuesta, también toma en cuenta que no existe una sola forma de hacer las cosas, puede que algunas de las respuestas que yo de a algún ejercicio no este 100% optimizada por el nivel en el que nos encontremos por lo que quizá el tuyo esté mucho mejor, en programación, a pesar de que todo es cero y uno, un resultado se puede conseguir por diversas formas, unas más óptimas que otras, pero mientras estamos aprendiendo lo importante es hacerlo.

Con cariño, Ing. Starling Germosen.