# Prueba técnica

## Sub proceso de sistemas jurisdiccionales.

### Indicaciones Generales

1. La prueba tiene un valor de 100 puntos
2. El tiempo de duración de la prueba será de 3 horas.
3. La prueba debe desarrollarla con Visual Studio 2019
4. Deberá enviar ya sea como adjunto o como compartido todo el contenido desarrollado comprimido en un único archivo formato zip.
5. Tome en consideración para la confección de la solución de la prueba, la implementación de:

* Creación de una aplicación en una de las siguientes tecnologías (UWP, WPF, MVC)
* Dominio del lenguaje C#
* Manejo de lenguaje XAML
* Uso de programación orientada a objetos
* Creación de objetos (Entidades)
* Ingreso de Datos, Procesamiento de Datos y Salida de Datos o Resultados.
* Declaración y uso de variables de diferentes tipos (int, string,float, char, decimal, string, objetos entre otros)
* Uso de Declaración de Variables (locales, globales, públicas y statics)
* Uso de gráficos en 2D, diseño de pantalla y controles
* Uso de propiedades en los controles gráficos
* Uso de Listas LINQ
* Operaciones agregar y consultar sobre listas genéricas en memoria
* Uso de operadores lógicos
* Uso de clases
* Uso de eventos
* Estándar al definir objetos, propiedades, variables, parámetros, métodos, funciones.
* Referencia entre objetos
* Mensajes de usuarios
* Controles de usuario.
* Diseño gráfico de la interface de usuario
* Estudio y comprensión de un requerimiento negocio enfocado en metodologías ágiles de desarrollo
* Traducción de historias de usuario.
* Guardar en base de datos SQL y manejo básico en instrucciones SQL

1. Esta prueba comprende los siguientes temas los cuales serán los puntos a evaluar en la calificación de la primera parte:

|  |  |
| --- | --- |
| **Observaciones** | **Puntaje** |
| Guarda correctamente la información | 10 |
| Hace visible estándar en código fuente | 4 |
| Usa correctamente las variables | 2 |
| Demuestra buenas prácticas | 5 |
| Código es limpio y comprensible | 6 |
| Muestra correctamente los mensajes según lo indicado en la HU cuando es exitoso | 2 |
| Muestra correctamente los mensajes según lo indicado en la HU cuando es error | 2 |
| Demuestra conocimiento en diseño mediante uso de lenguaje(XAML, html, componentes angular, MVC, entre otros) | 2 |
| Demuestra comprensión de requerimiento por medio de HU | 6 |
| Demuestra conocimiento en manejo de listas en memoria | 2 |
| Demuestra conocimiento en manejo de objetos, uso correcto de los nombres de objetos de negocio | 2 |
| Uso correcto de instancias de objetos | 2 |
| Definición correcta de variables | 2 |
| Diseño completo según lo indicado en la HU | 3 |
| Total | 50 |

1. La prueba está basada en dos ejercicios para su desarrollo:

|  |  |
| --- | --- |
| **I EJERCICIO** | **50 puntos** |
| **II EJERCICIO** | **50 puntos** |
| **TOTAL** | **100 puntos** |

# INTRUCCIONES: Los algoritmos deben ser programados en el lenguaje de programación C#

## Ejercicio #1 – 50 puntos

El señor José como parte de la ampliación de su microempresa de frutas y verduras desea agilizar su negocio en cuanto al inventario que lleva de sus productos, este negocio requiere estar moviendo las frutas y verduras de un lugar a otro, para inventariar y llevar control de los productos malos, buenos y propensos a entrar en estado de perdida. Don José desea tener forma de agilizar la gestión de inventarios por lo que decidió contratarlo usted como desarrollador para que realice la solución considerando como base las historias de usuario redactadas por el analista que previamente contrató don Jose.

Don Jose cuenta con un amigo asesor experto en TI el cual lo aconsejó de que su solución puede implementarla poco a poco de forma ágil sin esperar que esté toda la solución completa ,por lo que sugirió iniciar con el catálogo de productos manejados en la verdulería y posterior a estos continuar con los demás módulos, la idea principal es que don Jose pueda recibir en el primer día el módulo con el objetivo de que pueda ir registrando los productos que maneja en la verdulería, se estima que el registros de todos estos productos pueda llevarle a don José dos semanas, lo que daría tiempo a usted como desarrollador de hacer los demás módulos mientras donde José registra los productos, las siguientes partes de la solución van a ser analizadas y priorizadas por don Jose cuando reciba este primer producto.

Seguidamente se describe el requerimiento para este primer entregable el cual se detalla como requerimiento 1 a continuación.

### REQ01

Cómo un administrador de la verdulería se requiere poder realizar el registro de los productos manejados para poder tener un inventario sobre la cantidad de productos en existencia.

### Criterios de aceptación.

El sistema desplegará una pantalla similar a la siguiente imagen



El usuario deberá ingresar la siguiente información para cada registro de tipo de producto

* ID: Debe ser único y debe ser calculado, el cliente no debe digitarlo
* Nombre: Permite un máximo de 100 caracteres
* Tipo producto: Solo pueden ser una de las siguientes opciones, Frutas, Verduras, Vegetales, tubérculos
* Tipo de precio: Puede ser una de las siguientes opciones, por Unidad, por Kilo, por empaquete o bandeja como el caso de las fresas.
* Precio: Debe permitir ingreso de números con decimales.

El sistema debe presentar la opción de guardar

* Al hacer click en Guardar el sistema deberá de registrar el producto en el repositorio correspondiente.
* En caso de éxito deberá mostrar el siguiente mensaje “Se ha agregado correctamente un nuevo producto”
* En caso de error o fallo al registrar deberá indicar “No fue posible registrar el producto, intente nuevamente”

## Ejercicio #2 – 50 puntos

Explicar de forma detallada el resultado por instrucciones del siguiente script.

BEGIN

    --Instrucción 1

    DECLARE @MAESTROS TABLE

            (

                NumeroExpediente    CHAR(14),

                IdentificadorMoneda         SMALLINT,

                Moneda              VARCHAR(50),

                IdentificadorDelito         INT,

                Delito              VARCHAR(255),

                IdentificadorProvincia      SMALLINT,

                Provincia           VARCHAR(10)

            );

    DECLARE @DETALLES TABLE

            (

                NumeroExpediente    CHAR(14),

                CANTIDAD            SMALLINT

            );

--Instrucción 2

    INSERT INTO @DETALLES

    (

                NumeroExpediente,   CANTIDAD

    )

    SELECT      A.NumeroExpediente, COUNT(\*)

    FROM        @MAESTROS           A

    GROUP BY    A.NumeroExpediente;

--Instrucción 3

    INSERT INTO @MAESTROS

    (

                NumeroExpediente,       IdentificadorMoneda,        Moneda,             IdentificadorDelito,

                Delito,                 IdentificadorProvincia

    )

    SELECT      A.TC\_NumeroExpediente,  A.IdentificadorMoneda,  B.TC\_Descripcion,   A.IdentificadorDelito,

                C.TC\_Descripcion,       A.IdentificadorProvincia

    FROM        Expediente  A

    INNER JOIN  Moneda          B

    ON          B.IdentificadorMoneda           = A.IdentificadorMoneda

    LEFT JOIN   Delito          C

    ON          C.IdentificadorDelito           = A.IdentificadorDelito;

    --

    UPDATE      B

    SET         Provincia           = A.TC\_Descripcion

    FROM        Provincia   A

    LEFT JOIN   @MAESTROS           B

    ON          B.IdentificadorProvincia    = A.IdentificadorProvincia

--Instrucción 4

    SELECT TOP 10   A.NumeroExpediente,     A.CANTIDAD

    FROM            @DETALLES               A

    ORDER BY        NumeroExpediente

END

Respuesta:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Instrucción 1 | Instrucción 2 | Instrucción 3 | Instrucción 4 |
| Explicación de instrucción | Declara dos tablas temporales llamadas @MAESTROS Y @DETALLLES. @MAESTROS tiene columnas para el número de expediente, identificador de moneda, nombre de la moneda, identificador de delito, nombre del delito, identificador de provincia y nombre de la provincia. @DETALLES tiene columnas para el número de expediente y la cantidad. | Inserta datos en la tabla @DETALLES. La instrucción SELECT cuenta el número de filas para cada número de expediente en la tabla @MAESTROS y lo inserta junto con el número de expediente correspondiente en la tabla @DETALLES. | Inserta datos en la tabla @MAESTROS. La instrucción SELECT une las tablas Expediente, Moneda y Delito y selecciona varias columnas de cada tabla para insertar en la tabla @MAESTROS. | Selecciona los 10 primeros números de expediente y cantidad de la tabla @DETALLES, ordenados por número de expediente |
| Explicación de resultado | Como resultado se crean dos tamblas temporales | La instrucción INSERT INTO @DETALLES selecciona el número de expediente y cuenta el número de filas para cada número de expediente en la tabla @MAESTROS utilizando la función COUNT(\*), agrupado por número de expediente. Los resultados se insertan en la tabla @DETALLES. En resumen, la instrucción insertará en la tabla @DETALLES un registro por cada número de expediente en la tabla @MAESTROS, que contendrá el número de expediente y la cantidad de filas correspondiente | Hace un insert en la tabla maestros y realiza un select uniendo las tablas expediente moneda y delito | Realiza un select de los primeros 10 expedientes y cantidad de la tabla@DETALLES d los cuales ordena por numero de expediente |