# 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 2 horas y 40 minutos.
3. La prueba debe desarrollarla con Visual Studio 2019 y SQL Sever 2017
4. Deberá enviar ya sea como adjunto o como compartido todo el contenido desarrollado comprimido en un único archivo formato zip.
5. Esta prueba comprende los siguientes temas:

* Creación de una aplicación en una de las siguientes tecnologías (UWP, WPF, MVC, ASP.NET, Angular/HTML5)
* Dominio del lenguaje C#
* 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.
* Conocimiento básico en instrucciones SQL

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#**

**Nota: se calificará sobre ejecución de los programas, uso clases, entidades, variables, correcta interpretación de requerimientos, uso visible de un estándar y diseño de pantallas.**

**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(10PTS)



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. (30pts)
* En caso de éxito deberá mostrar el siguiente mensaje “Se ha agregado correctamente un nuevo producto” (5pts)
* En caso de error o fallo al registrar deberá indicar “No fue posible registrar el producto, intente nuevamente” (5pts)

**Ejercicio #2 – 50 puntos**

Explicar de forma detalla la siguiente instrucción SQL

BEGIN

    --

    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

            );

    --

    INSERT INTO @DETALLES

    (

                NumeroExpediente,   CANTIDAD

    )

    SELECT      A.NumeroExpediente, COUNT(\*)

    FROM        @MAESTROS           A

    GROUP BY    A.NumeroExpediente;

    --

    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

    --

    SELECT TOP 10   A.NumeroExpediente,     A.CANTIDAD

    FROM            @DETALLES               A

    ORDER BY        NumeroExpediente

END

Respuesta:

BEGIN

    --

    /\*EL @ INDICA QUE SE VA A DECLARAR UNA VARIABLE, EN ESTE CASO UNA TABLA LLAMADA MAESTROS\*/

    DECLARE @MAESTROS TABLE

            (

                NumeroExpediente    CHAR(14),

                IdentificadorMoneda         SMALLINT,

                Moneda              VARCHAR(50),

                IdentificadorDelito         INT,

                Delito              VARCHAR(255),

                IdentificadorProvincia      SMALLINT,

                Provincia           VARCHAR(10)

            );

    /\*IGUAL QUE EL CASO ANTERIOR SE DECLARA UNA TABLA LLAMADA DETALLES\*/

    DECLARE @DETALLES TABLE

            (

                NumeroExpediente    CHAR(14),

                CANTIDAD            SMALLINT

            );

    --

    /\*

    SE INSERTA EN LAS COLUMNAS "NumeroExpediente" Y "CANTIDAD" DE LA TABLA DETALLES LOS VALORES RESULTANTES DE UNA CONSULTA, EN ESTE CASO LA CONSULTA PROVIENE DE LOS DATOS EXTRAIDOS DE LA TABLA MAESTRO, EN DONDE AL CAMPO "NumeroExpediente" DE LA TABLA DETALLES SE LE INSERTAN LOS DATOS "NumeroExpediente" DE LA TABLA MAESTRO.

NOTA: LA LETRA “A” HACE ALUSION A UNA TABLA TEMPORAL EN EL CAMPO DE CANTIDAD SE INSERTARA EL RESULTADO DE LA OPERACION RESERVADA "COUNT(\*)"

    QUE EN ESTE CASO EL RESULTADO SIEMPRE VA SER = 1

    \*/

    INSERT INTO @DETALLES

    (

                NumeroExpediente,   CANTIDAD

    )

    SELECT      A.NumeroExpediente, COUNT(\*)

    FROM        @MAESTROS           A

    GROUP BY    A.NumeroExpediente;

    --

    /\*

    SE INSERTAN EN LAS COLUMNAS

    "NumeroExpediente, IdentificadorMoneda,Moneda,IdentificadorDelito,Delito,IdentificadorProvincia"

    DE LA TABLA MAESTROS LOS DATOS RESULTANTES DE UNA CONSULTA REALIZADA A LAS TABLAS LLAMADADAS "Expediente","Moneda" Y "Delito"

    (SE DESCONOCE ORIGEN DE LAS TABLAS, PARA ESTE CASO SE TOMARÁ EN CONSIDERACION QUE LA TABLAS EXISTEN)

    \*/

    INSERT INTO @MAESTROS

    (

                NumeroExpediente,       IdentificadorMoneda,        Moneda,             IdentificadorDelito,

                Delito,                 IdentificadorProvincia

    )

    /\*SE REFERENCIAN LOS DATOS QUE SE QUIEREN EXTRAER DE LAS TABLAS\*/

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

                C.TC\_Descripcion,       A.IdentificadorProvincia

    FROM        Expediente  A

    /\*

    SE COMPARAN LOS DATOS DE LAS TABLAS "Expediente" y "Moneda"

    SACANDO TODOS LOS SOLO LOS DATOS QUE CONSIDAN CON LA IGUALDAD

    REFERENTE "B.IdentificadorMoneda = A.IdentificadorMoneda"

    (EL RESULTADO CONSIDERA TODAS LAS COLUMNAS RESULTANTES FORMANDO UNA NUEVA TABLA, ESTO POR EFECTOS DEL INNER JOIN)

    \*/

    INNER JOIN  Moneda          B

    ON          B.IdentificadorMoneda = A.IdentificadorMoneda

      /\*

    SE VUELVE A REALIZAR UNA COMPARACION DE DATOS

    PERO AHORA ENTRE LA SELECCION OBTENIDA DEL "INNER JOIN" Y LA TABLA "Delito"

    DONDE LOS DATOS CONSIDEAN CON LA IGUALDAD " C.IdentificadorDelito = A.IdentificadorDelito;"

    SE VUELVE A FORMAR UNA NUEVA TABLA LLENANDO DE FORMA COMPLETA LAS FILAS QUE CUMPLAN CON LA IGUALDAD

    Y DE FORMA INCOMPLETA LAS FILAS QUE NO CONSIDAN CON LA IGUALDAD, PERO SOLO CONSERVANDO LOS DATOS DE LA PRIMERA TABLA

    OBTENIDA EN EL INNER JOIN. (SE CONSERVAN LOS DATOS A LA IZQUIERDA)

        \*/

    LEFT JOIN   Delito          C

    ON          C.IdentificadorDelito           = A.IdentificadorDelito;

    --

    /\*

    EL "UPDATE" SE UTILIZA PARA UNA ACTUALIZACION DE LA TABLA

    PERO EN ESTE CASO NO ESPOSIBLE REALIZAR LA ACTULIZACION

    YA QUE SE TIENE UN ERROR DE SINTAXIS O OMISION DE INSTRCCINES

    \*/

    UPDATE      B

    SET         Provincia = A.TC\_Descripcion

    FROM        Provincia   A

    LEFT JOIN   @MAESTROS           B

    ON          B.IdentificadorProvincia    = A.IdentificadorProvincia

    --

    /\*

    SELECCIONA LOS DATOS "A.NumeroExpediente,A.CANTIDAD"

    DE LAS PRIMERAS 10 FILAS

    DE LA TABLA "DETALLES"

    Y LAS ORDENA SEGUN LA COLUMNA "NumeroExpediente"

    \*/

    SELECT TOP 10   A.NumeroExpediente,     A.CANTIDAD

    FROM            @DETALLES               A

    ORDER BY        NumeroExpediente

END