



UNIVERSIDAD NACIONAL DE INGENIERIA

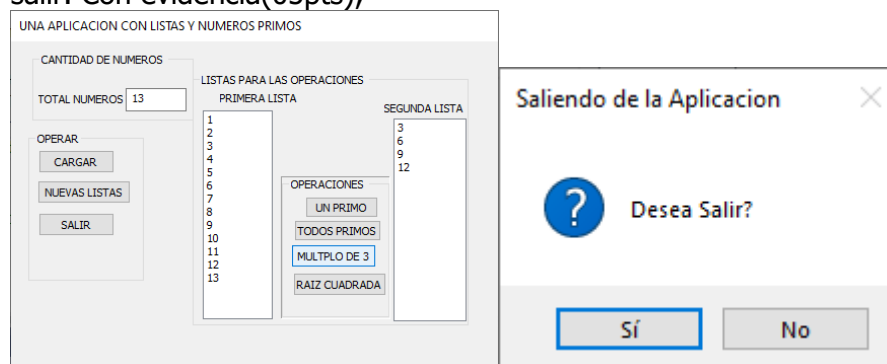
EXAMEN FINAL DE PROGRAMACION ORIENTADA A OBJETOS MB545

Tiempo: 110 min.

Fecha 20 de julio del 2021

- Se monitoreará al alumno, y deberá compartir su pantalla cuando se le requiera
- Las soluciones que sean parecidas o iguales a uno o más alumnos se les **calificará con A0** a todos ellos.
- Obligatoriamente deberá subir al aula virtual 2 archivos: Su solución en pdf y el archivo comprimido de la carpeta de su programa depurado (en rar o zip la cual debe pesar menos de 2MB)
- Por cada solución que desarrolle debe indicar su codificación, captura de pantalla de las interfaces y variables asociadas, y corrida de los programas con 2 juegos de valores distintos de entrada.
- **Tendrá** 30 minutos adicionales exclusivamente para que el alumno pueda subir las soluciones al aula virtual con anticipación, pasado el límite ya no se aceptará su solución.

1. Desarrolle una aplicación que tenga dos listas, una que carga los números consecutivos empezando desde 1 (unidad), y la otra lista contiene las operaciones que debe realizar de acuerdo a los botones de evento, cada botón debe realizar las operaciones indicadas, no olvidar la limpieza de las listas, como también la opción salir. Con evidencia(05pts),

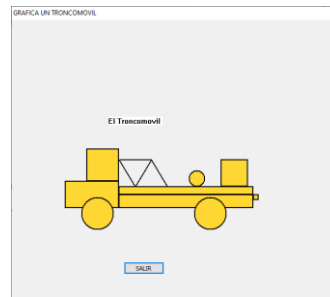
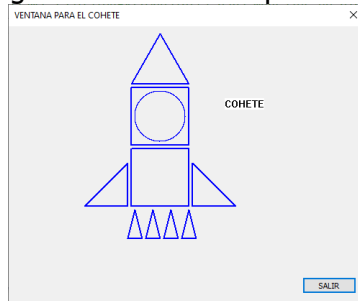


- 2 Se tiene la clase Serie que de manera aproximada, mediante Taylor, calcula el seno(x) de la forma $\text{sen}(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots$ donde x, Ud lo convierte a radian, solo para la función sen(x), realice dicha clase y luego, derive otra clase Trapecio que resuelve la integral por éste método de la función, aquí x es cualquier número, ya sea positivo o negativo. Solo en esta pregunta puede usar una aplicación de consola.

$$f(x) = \int_{0.5}^x (x^3 - x^2 + \sqrt{x+3}) dx$$

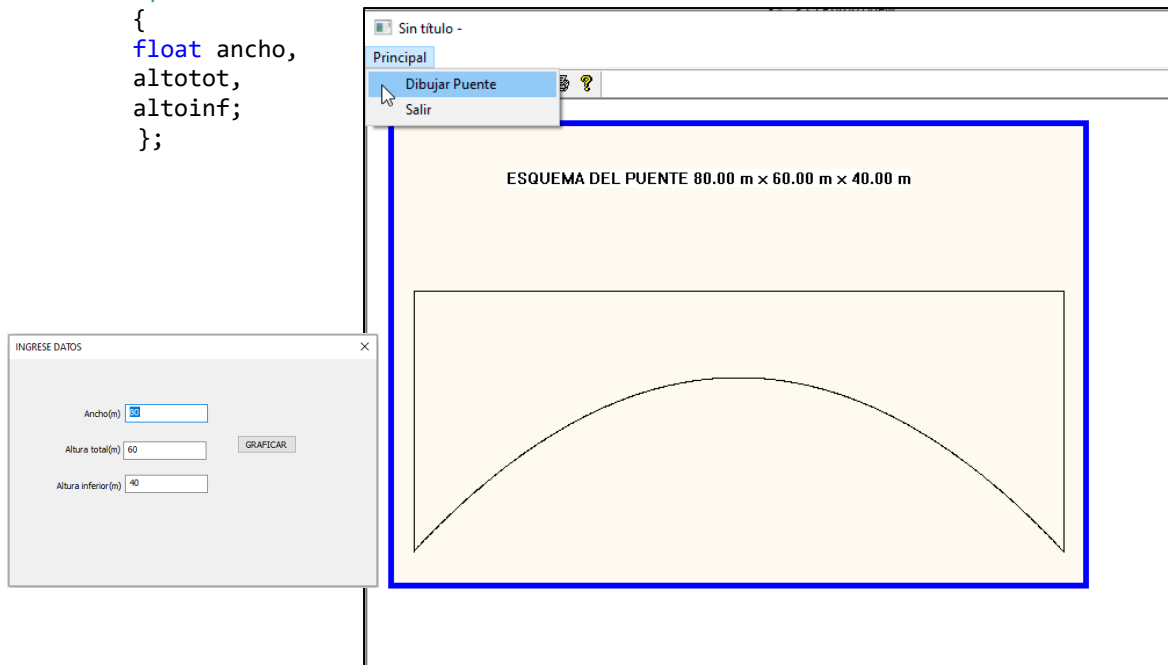
Con evidencia (5pts).

3. Diseñe un sistema con usuario y clave de acceso, con su respectivo mensaje de ingreso o de error en caso contrario, en cada caso con su respectivo icono y solo se acepta 03 intentos. Queda a su criterio dicha confección. Una vez ingresado deberá crear un menú en el cuadro de diálogo principal con 4 opciones que llamarán a cada una de las ventanas gráficas, a una opción de Ayuda(Acerca de...) que incluya los datos del alumno y una opción salir. La llamada a los gráficos dibujará las siguientes figuras tal como se aprecian a continuación.



4. Desarrolle un programa que esquematice un puente con soporte parabólico, la gráfica se debe dibujar desde el menú Dibujar puente, la cual llamará a un cuadro de diálogo modal, donde se cargarán sus medidas y luego graficará en el panel principal exactamente como la figura adjunta, mostrando en el título las 3 medidas generales del puente. Para cualquier medida que se ingrese, el gráfico debe encajarse en un rectángulo de 600x400, use la siguiente estructura de datos para definir las medidas del puente:

```
struct spuente
{
    float ancho,
    altotot,
    altoinf;
};
```



RUBRICA

CRITERIOS	5	3	2	1	0
CLARIDAD Y PRECISIÓN EN LA SOLUCION	La solución tiene evidencia del resultado con la corrida y lo guarda correctamente para su envío.	La solución no es completa, pero tiene corrida no hay evidencia.	No hay solución, la codificación no es completa, no tiene evidencia	No hay claridad en la codificación ni precisión	Respuesta con algunas líneas sin precisar
CONOCE Y RESUELVE LA PLATAFORMA VISUAL C++	Utiliza adecuadamente En modo consola la solución completa del problema planteado, usando las clases. E evidencia con la corrida	Resuelve en modo consola parcialmente la clase y la corrida no es completa, tiene evidencia	El resultado no demuestra el uso eficaz de la clase, no hay corrida	No hay solución del problema planteado, tiene líneas de código, pero no evidencia hacia una solución.	Sin solución ni líneas de código fuente
RESUELVE CON CONOCIMIENTO DE LA PLATAFORMA DE C++ CON WINDOWS	Usa una estrategia eficiente en la confección de formularios, ventanas y menú con opciones cumpliendo con lo requerido en la solución.	Usa una estrategia no adecuada acorde con la solución tiene errores subsanables, tiene corrida y evidencia sin llegar a la solución completa.	Usa una estrategia no entendible tiene errores de sintaxis. No concluye en la solución. Evidencia poca preparación	No Usa una estrategia eficiente, desconoce la solución tiene errores de sintaxis y no hay evidencia de solución	Sin solución ni código fuente evidencia no conocer el tema
CONOCE LA FORMA CORRECTA DE ENVIAR LA SOLUCION	Envía la solución en una carpeta a su drive del Gmail. Envía el link al profesor sin demora. Envía la solución en un archivo de Word o pdf al aula virtual de la solución del examen	Envía la solución al aula virtual. No envía el link a su profesor del curso.	Envía la solución solo parcialmente, no envía el total al aula ni al profesor	Envía pantallazos sin orden ni legibles	No envía nada desconoce

SOLUCION PROBLEMA N01

```
void CMFCLISTAPRIMOSDlg::OnBnClickedPasaprimo(){
    int n, i, ContDivisores = 0;
    CString strC1;
    //Obtener numero seleccionado de la primera caja de lista
    m_ListNumeros.GetText(m_ListNumeros.GetCurSel(),strC1);
    //Esta conmo cadena
    //Convertir a un valor numerico entero
    n = _wtoi(strC1);
    for (i = 1; i <= n; i++)
        if (n %i == 0)
            ContDivisores++;

    if (ContDivisores == 2)
        //Es primo entonces añadir nuenro a la segunda caja de Lista
        m_ListPrimos.AddString(strC1);

    // TODO: Agregue aquí su código de controlador de notificación de control
}
```

```
void CMFCLISTAPRIMOSDlg::OnBnClickedPasaprimos()
{
    int n, ce, i, j,cp=0;
    CString strC1;
    ce = m_ListNumeros.GetCount();//Contar elementos de la primera caja de
lista
    if (ce > 0)
    {
        for (i = 0; i < ce; i++)
        {
            m_ListNumeros.GetText(i, strC1); // Obtener el elemento
            n = _wtoi(strC1);
            int contDivisores = 0;
            for (j = 1; j <= n; j++)
            {
                if (n % j == 0)
                    contDivisores++;
            }
            if (contDivisores == 2) {
                // Es primo; entonces añadir numero a la segunda caja
de lista
                m_ListPrimos.AddString(strC1);
                //MessageBox(_T("Cantidad de primos "), _T("Numeros
Primos "), MB_ICONEXCLAMATION | MB_OK);
            }
        }
    }
}

// TODO: Agregue aquí su código de controlador de notificación de control
```

```
void CMFCLISTAPRIMOSDlg::OnBnClickedNuevalistas()
```

```

{
    UpdateData(true);
    m_ListNumeros.ResetContent();
    m_ListPrimos.ResetContent();
    m_CantidadNumeros = 0;
    UpdateData(false);
    // TODO: Agregue aquí su código de controlador de notificación de control
}

void CMFCLISTAPRIMOSDlg::OnBnClickedBoton3()
{
    int i, n, ce;
    CString strC;
    ce = m_ListNumeros.GetCount();
    if (ce > 0)
        for (i = 0; i < ce; i++) {
            m_ListNumeros.GetText(i, strC); //obtiene el elemento
            n = _wtoi(strC); // convierte a numerico
            if (n % 3 == 0) //ver si es par
                m_ListPrimos.AddString(strC); //;pasa a la otra lista
        }
    // TODO: Agregue aquí su código de controlador de notificación de control
}

void CMFCLISTAPRIMOSDlg::OnBnClickedCargar2()
{
    int i, n, ce, m;
    CString strC2;
    ce = m_ListNumeros.GetCount();
    if (ce > 0)
        for (i = 0; i < ce; i++) {
            m_ListNumeros.GetText(i, strC2); //obtiene el elemento
            m = sqrt(_wtoi(strC2));
            m_ListPrimos.AddString(_itow(m)); //;pasa a la otra lista
        }

    // TODO: Agregue aquí su código de controlador de notificación de control
}

```

SOLUCION PREGUNTA N02

```

#include <iostream>
#include<stdlib.h>
#include<time.h>
#include<math.h>
using namespace std;
class serie {
protected:
    int n;
public:
    serie();
    float valor(float);
    ~serie() {
        cout << "Liberando memoria..." << endl;
    }
};
serie::serie() {
    n = 10;
}
float serie::valor(float x) {
    float S = 0; float F = 1;

```

```

        for (int i = 1; i <= n; i++) {
            S += powf(x, 2 * i - 1) / F;
            F *= (2 * i - 1) * 2 * i;
        }
        return(S);
    }
}
class trapecio :public serie {
protected:
    float a;
    float b;
public:
    trapecio();
    void mostrarl();
    float calint();
};
trapecio::trapecio() {
    n = 100;
    a = 0.5;
    b = 1.5;
}
void trapecio::mostrarl() {
    cout << "Limite inferior = " << a << endl
         << "Limite superior = " << b << endl;
}
float trapecio::calint() {
    float S = 0;
    for (int i = 1; i <= n; i++) {
        S += ((b - a) / (2 * n)) * ((powf(a + (b - a) * i / n, 3) - powf(a + (b - a) * i / n, 2) + powf(a + (b - a) * i / n, 0.5) - 3) + (powf(a + (b - a) * (i - 1) / n, 3) - powf(a + (b - a) * (i - 1) / n, 2) + powf(a + (b - a) * (i - 1) / n, 0.5) - 3));
    }
    return(S);
}
}
void main() {
    serie taylor; float x; trapecio integral;
    cout << "Ingresar el valor de x en radianes: "; cin >> x;
    cout << "Aproximacion de senx mediante Taylor: " << taylor.valor(x) << endl;
    cout << "Integral mediante trapecio:" << endl;
    integral.mostrarl();
    cout << "Valor aproximado de la integral de x^3 - x^2 + raiz(x) - 3: " << integral.calint() << endl;
    system("pause");
}

```

SOLUCION 03

El cohete

```

mipincel.CreatePen(PS_SOLID, 2, RGB(0, 0, 255));
dc.SelectObject(mipincel);
////////////////////
dc.MoveTo(170, 170);
dc.LineTo(170, 250);
dc.MoveTo(170, 170);
dc.LineTo(250, 170);
dc.MoveTo(250, 170);
dc.LineTo(250, 250);
dc.MoveTo(250, 250);
dc.LineTo(170, 250);

```

```

//////////
dc.MoveTo(170, 165);
dc.LineTo(170, 85);
dc.MoveTo(170, 165);
dc.LineTo(250, 165);
dc.MoveTo(250, 165);
dc.LineTo(250, 85);
dc.MoveTo(250, 85);
dc.LineTo(170, 85);
//////////
dc.MoveTo(170, 80);
dc.LineTo(250, 80);
dc.MoveTo(170, 80);
dc.LineTo(210, 10.718);
dc.MoveTo(250, 80);
dc.LineTo(210, 10.718);
//////////
dc.MoveTo(165, 250);
dc.LineTo(165, 190);
dc.MoveTo(165, 250);
dc.LineTo(105, 250);
dc.MoveTo(165, 190);
dc.LineTo(105, 250);
//////////
dc.MoveTo(255, 250);
dc.LineTo(315, 250);
dc.MoveTo(255, 250);
dc.LineTo(255, 190);
dc.MoveTo(315, 250);
dc.LineTo(255, 190);
//////////
dc.MoveTo(175, 255);
dc.LineTo(165, 295);
dc.MoveTo(165, 295);
dc.LineTo(185, 295);
dc.MoveTo(175, 255);
dc.LineTo(185, 295);
dc.MoveTo(190, 295);
dc.LineTo(210, 295);
dc.MoveTo(190, 295);
dc.LineTo(200, 255);
dc.MoveTo(200, 255);
dc.LineTo(210, 295);
dc.MoveTo(215, 295);
dc.LineTo(235, 295);
dc.MoveTo(215, 295);
dc.LineTo(225, 255);
dc.MoveTo(235, 295);
dc.LineTo(225, 255);
dc.MoveTo(240, 295);
dc.LineTo(260, 295);
dc.MoveTo(240, 295);
dc.LineTo(250, 255);
dc.MoveTo(250, 255);
dc.LineTo(260, 295);
float x, y, u, r;
for (u = 0; u <= 0.1)
{
    r = 35;
    y = 125 - r * sin(u * 3.1415 / 180);
    x = 210 - r * cos(u * 3.1415 / 180);
    dc.SetPixel(x, y, RGB(0, 0, 255));
}

```

```

    }
    dc.TextOutW(200, 100, _T("COHETE"));
    //////////////////////////////////////
    //CARRO

    //
    // TODO: Agregue aquí su código de controlador de
    mensajes
    // No llama a CDialogEx::OnPaint() para mensajes de
    pintura

    CPen *n_pincel=new CPen; // Crea un objeto puntero del tipo
    CPen
    n_pincel->CreatePen(PS_SOLID,6,RGB(0,0,0)); //initialization

    dc.SelectObject(n_pincel);
    CBrush *n_brocha=new CBrush;
    n_brocha->CreateSolidBrush(RGB(252, 216, 119));
    dc.SelectObject(n_brocha);

    dc.Rectangle(100,300,200,350);

    dc.Rectangle(140,240,200,300);
    dc.Ellipse(130,330,190,390);
    dc.Rectangle(200,310,450,325);
    dc.Rectangle(200,325,450,350);
    dc.Ellipse(340,330,400,390);
    dc.Ellipse(330,280,360,310);

    dc.MoveTo(200,260);
    dc.LineTo(230,310);
    dc.MoveTo(260,260);
    .....
    dc.LineTo(260,260);

    dc.MoveTo(260,260);
    dc.LineTo(230,310);
    dc.MoveTo(230,310);
    dc.LineTo(290,310);
    dc.MoveTo(260,260);
    dc.LineTo(290,310);
    hPen = CreatePen(PS_DASHDOTDOT,1,RGB(0,255,255));
    SelectObject(hdc, hPen);
    .....
    dc.Rectangle(390,260,440,310);
    dc.Rectangle(450,325,460,335);}

```


SOLUCION PREGUNTA 04

```

void CEPprob1View::OnPrincipalDibujarpuente()
{
    Ccuadro1 v;
    v.DoModal();
    p.altoinf = v.m_altoinf;
    p.altotot = v.m_altotot;
    p.ancho = v.m_ancho;
    CClientDC g(this);
    CPen t;          CBrush f;
    float pi = atan(1.0) * 4,xy;
    t.CreatePen(PS_SOLID, 5, RGB(0, 0, 255)); g.SelectObject(t);
    f.CreateSolidBrush(RGB(255, 250, 240)); g.SelectObject(f);
    g.Rectangle(20, 20, 620, 420);
    int yy;
    for (int i = 0; i <= 1000; i++)
    {
        int xx = 40 + 560 * i / 1000;
        float x = -p.ancho / 2 + p.ancho * i / 1000;
        float y = x * x * (4 * p.altoinf / pow(p.ancho, 2))*150/ p.altoinf;
        yy = 240+y;
        g.SetPixel(xx, yy, RGB(0, 0, 0));
    }
    t.DeleteObject();
    t.CreatePen(PS_SOLID, 1, RGB(0, 0, 0)); g.SelectObject(t);
    xy = yy - (p.altotot / p.altoinf) * 150;
    g.MoveTo(40, yy); g.LineTo(40, xy);
    g.LineTo(600, xy); g.LineTo(600, yy);
    CString cad; cad.Format(_T("%.2f m x %.2f m x %.2f m"), p.ancho,
p.altotot, p.altoinf);
    g.TextOutW(120, 60, _T("ESQUEMA DEL PUENTE ")+cad);
}

```