Tema: Algoritmos de Triangulación.
Objetivo General:

- **Conocer e implementar un ejemplo haciendo uso de los algoritmos correspondientes.**

Objetivos Específicos del Laboratorio:

1. **Implementar los algoritmos propuestos por el profesor.**
2. **Analizar la teoría de Fisk para resolver los problemas de la Galería de Arte.**
3. **Motivar el uso de Lenguaje C como herramienta de implementación de Algoritmos.**

Integrantes:

1. _____
2. _____
3. _____
4. _____
5. _____

Trabajo Previo:

- ✓ **Analizar los algoritmos propuestos por el docente.**
- ✓ **Utilizando las funciones de atributos de primitivos gráficos coloque una "nube de puntos" y realice la búsqueda y colocación de los guardas.**

Desarrollo del contenido del Laboratorio:

1. **Entrar al software y ponerlo a punto para trabajar.**
2. **Desarrollar los ejercicios propuesto para el laboratorio.**
3. **Presentar al docente los avances del laboratorio.**

Docente: Gabriel Rafael Lacayo Saballos
E-mail: gabriellacayo@gmail.com

Viernes 13 de mayo de 2011
Grupos: 2M2-CO y 2M3-CO

Algoritmos de Ayuda

```
/*************************************************************************
* DATOS DEFINIDOS POR EL USARIO PARA EL MANJO DE LOS PUNTOS DEL POLIGONO
*************************************************************************/
typedef int tPointd[2];
typedef  tPointi tPolygoni[100];

/*************************************************************************
* ESTA FUNCION CALCULA EL AREA DE UN TRIANGULO, PASANDO COMO ARGUMENTO TRES*
* PUNTOS A, B Y C    *
*************************************************************************/
int Area2(tPointi a, tPointi b,tPointi c)
{
        return (( a[0]*b[1]) - (a[1]*b[0])+(a[1]*c[0])-(a[0]*c[1])+(b[0]*c[1])-(c[0]*b[1]));
}

/*************************************************************************
* ESTA FUNCION VERIFICA SI EL PUNTO DEL POLIGONO ESTA A LA IZQUIERDA DE UN *
* SEGMENTO, RETORNO TRUE SI AREA2>0
*************************************************************************/
bool Left(tPointi a,tPointi b,tPointi c)
{
        return Area2(a,b,c)>0;
}

/*************************************************************************
* ESTA FUNCION VERIFICA SI UN PUNTO ESTA A LA IZQUIERDA O SI ESTA SOBRE UN *
* SEGMENTO, RETORNA TRUE SI AREA2>=0
*************************************************************************/
bool LeftOn(tPointi a, tPointi b,tPointi c)
{
        return Area2(a,b,c)>=0;
}

/*************************************************************************
*ESTA FUNCION VERIFICA SI UN PUNTO ESTA SOBRE UN SEGMENTO    *
*************************************************************************/
bool Collinear(tPointi a, tPointi b,tPointi c)
{
        return Area2(a,b,c)==0;
}

/*************************************************************************
*ESTA FUNCION ANALIZA LA INTERSECCION PROPIA ENTRE DOS PUNTOS          *
*************************************************************************/
bool IntersectProp(tPointi a,tPointi b, tPointi c,tPointi d)
{
        if(Collinear(a,b,c)||Collinear(a,b,d) || Collinear(c,d,a)||Collinear(c,d,b))
                return FALSE;
        return Xor(Left(a,b,c),Left(a,b,d))&& Xor(Left(c,d,a),Left(c,d,b));
}

/*************************************************************************
*ESTA FUNCION VERIFICA SI UN PUNTO ESTA ENTRE UN SEGMENTO              *
*************************************************************************/
bool Between(tPointi a,tPointi b, tPointi c)
{
        if (!Collinear(a,b,c))
                return FALSE;
        if(a[X] != b[X])
                return(( a[X] <= c[X]) && (c[X] <= b[X])) || (( a[X] >= c[X]) && (c[X] >= b[X]));
        else
                return(( a[Y] <= c[Y]) && (c[Y] <= b[Y])) || (( a[Y] >= c[Y]) && (c[Y] >= b[Y]));
}
```

```
/*****************************************************************************
*ESTA FUNCION ANALIZA LA INTERSECCION ENTRE DOS PUNTOS                      *
*****************************************************************************/
bool Intersect(tPointi a,tPointi b, tPointi c,tPointi d)
{
        if(IntersectProp(a,b,c,d))
                return TRUE;
        else
                if(Between(a,b,c)||Between(a,b,d)|| Between(c,d,a) ||Between(c,d,b))
                        return TRUE;
        else
                return FALSE;
}

/*****************************************************************************
*ESTA FUNCION IDENTIFICA LAS INTERSECCIONES DE LAS DIAGONALES DE UN POLIGONO   *
*****************************************************************************/
bool Diagonalie(int i,int j, int n, tPolygoni P)
{
        int k,k1;
        for(k=0;k<n;k++)
                {
                        k1=(k+1)%n;
                        if( !( (k==i) || (k1==i) || (k==j) || (k1==j)))
                                if( Intersect(P[i],P[j],P[k],P[k1]))
                                        return FALSE;
                }
        return TRUE;
}

/*****************************************************************************
*ESTA FUNCION DETERMINA LAS DIAGONALES INTERNAS DE LAS EXTERNAS
*****************************************************************************/
bool InCone(int i,int j,int n,tPolygoni P)
{
        int i1,in1;

        i1=(i+1) %n;
        in1=(i+n-1)%n;

        if(LeftOn(P[in1],P[i],P[i1]))
                return Left(P[i],P[j],P[in1])&& Left (P[j],P[i],P[i1]);
        else
                return !(LeftOn( P[i],P[j],P[i1])&& LeftOn(P[j],P[i],P[in1]));
}

/*****************************************************************************
*ESTA FUNCION DETERMINA LA VALIDES DE LAS DIAGONALES   *
*****************************************************************************/
bool Diagonal(int i,int j,int n,tPolygoni P)
{
        return (InCone(i,j,n,P) && Diagonalie(i,j,n,P));
}

/*****************************************************************************
ESTA FUNCION RETORNA EL VALOR DE LA CONJUNCION DE LA NEGACION DE DOS   *
FUNCIONES
*****************************************************************************/
bool Xor(bool x,bool y)
{
        return !x ^ !y;
}
/*ESTATUS DE LAS COORDENADAS DEL PUNTERO DEL MOUSE*/
         textcolor(3);gotoxy(2,7); cprintf ("x=%d ",mposx(8));
         gotoxy(2,8); cprintf ("y=%d ",mposy(8));
```