

Anexo

CÓDIGOS

A.1. INTRODUCCIÓN

En las páginas siguientes se enseña los códigos utilizados para crear la aplicación del algoritmo matricial para CATIA.

Se han separado los códigos en función del bloque al que pertenecen; en primer lugar se ven los códigos del bloque de MATLAB y posteriormente se verán los correspondientes al bloque de BASIC.

A.2. CÓDIGOS MATLAB

A.2.1. Archivo rot90.m

```
function [A90]=rot90(A)
[a,b]=size(A);
for j=1:b, %creamos los términos de la matriz A90 girando los que teníamos de la
matriz A
    for i=1:a,
```

```
    A90(b+1-j,i)=A(i,j);  
end  
end
```

A.2.2. Archivo rot180.m

```
function [A180]=rot180(A)  
[a,b]=size(A);  
for j=1:b, %creamos los terminos de la matriz A180 girando los q teniamos de la matriz  
A  
    for i=1:a,  
        A180(a+1-i,b+1-j)=A(i,j);  
    end  
end
```

A.2.3. Archivo rot270.m

```
function [A270]=rot270(A)  
[a,b]=size(A);  
for j=1:b, %creamos los terminos de la matriz A270 girando los q teniamos de la matriz  
A  
    for i=1:a,  
        A270(j,a+1-i)=A(i,j);  
    end  
end
```

A.2.4. Archivo esq.m

```
function [INFO]=esq(Vi,Vj,c,ROTACIONES)  
g=1;  
for e=1:length(c)  
    ViP=Vi(e,:);  
    VjP=Vj(e,:);  
    for f=1:ROTACIONES(e),
```

```

ViQ=ViP(1+(f-1)*c(e):f*c(e));
VjQ=VjP(1+(f-1)*c(e):f*c(e));
S=sparse(ViQ,VjQ,ones(1,length(ViQ)));
[m,n]=size(S);
F1=S(1,:);
[I1,J1,s1]=find(F1);
F2=S(:,1);
[I2,J2,s2]=find(F2);
densidaddehuecos=((n-length(I1))+(m-length(I2))+(S(1,1)-1))/(n+m-1);
G(g,1:5)=[e,f,m,n,densidaddehuecos];
g=g+1;
end
end
a=min(G(:,5));
f=1;
for e=1:length(G(:,1)),
    if G(e,5)>a,
        else
            INFO(f,:)=G(e,:);
            f=f+1;
        end
    end
end
end

```

A.2.5. Archivo separacion.m

```

function [A]=separacion(A1,margen)
[a,b]=size(A1);
A=zeros(a+2*margen,b+2*margen);
for j=1:b,
    for i=1:a,
        if A1(i,j)>0,
            A(i:i+margen*2,j:j+margen*2)=ones(1+margen*2,1+margen*2);
        else
        end
    end
end

```

```
end  
end
```

A.2.6. Archivo orientacionznegativa.m

```
function [SA]=orientacionznegativa(A)  
[a,b]=size(A);  
for i=1:a,  
    SA(a+1-i,1:b)=A(i,1:b);  
End
```

A.2.7. Archivo prox2.m

```
function [INFOR]=prox2(Vi,Vj,D,c,m,n,ROTACIONES,cantidad)  
K=[m,n,0,0,0];  
K1=K;  
limite=0;  
[p,q]=size(D);  
[Di,Dj]=find(D);  
DD=D;  
for i=1:p,  
    j=1;  
    s=1;  
    while s>0,  
        LB=find(1-DD(i,:)>0);  
        if length(LB)<1,  
            LB=q+1;  
            LA=n+1;  
            s=0;  
        else  
            DDD=DD(i,LB(1):q);  
            DD(i,:)=[zeros(1,LB(1)-1),DDD];  
            LA=find(DD(i,:)>0);  
            if length(LA)<1,
```

```

    LA=n+1;
    s=0;
else
    DDD=DD(i,LA(1):q);
    DD(i,:)=[ones(1,LA(1)-1),DDD];
end
end
H1(2*i,j)=LA(1)-LB(1);
H1(2*i-1,j)=LB(1);
c1(i)=j;
j=j+1;
end
end
for e=1:length(c),
    if cantidad(e)>0,
        ViP=Vi(e,:);
        VjP=Vj(e,:);
        for f=1:ROTACIONES(e),
            ViQ=ViP((f-1)*c(e)+1:f*c(e));
            VjQ=VjP((f-1)*c(e)+1:f*c(e));
            Q=sparse(ViQ,VjQ,ones(1,length(ViQ)));
            if max(ViP(1:c(e)))*max(VjP(1:c(e)))>limite,
                H2=0;
                for i=1:max(ViQ),
                    j=1;
                    s=1;
                    while s>0,
                        LB=find(Q(i,:)>0);
                        if length(LB)<1,
                            s=0;
                        else
                            QQ=Q(i,LB(1):max(VjQ));
                            Q(i,:)=[ones(1,LB(1)-1),QQ];
                            LA=find(1-Q(i,:)>0);

```

```

    if length(LA)<1,
        LA=max(VjQ)+1;
        s=0;
    else
        QQ=Q(i,LA(1):max(VjQ));
        Q(i,:)=zeros(1,LA(1)-1),QQ];
    end
    H2(2*i,j)=LA(1)-LB(1);
    H2(2*i-1,j)=LB(1);
    c2(i)=j;
end
j=j+1;
end
end
g=min(p+1,m-max(ViQ)+2);
a=1;
while a<g,
    s2=0;
    VB=0;
    for i=1:2:2*min(max(ViQ),p-a+1),
        VV=find(H1(2*(a)-1+i,:)>=H2(i+1,1));
        if length(VV)<1,
            VB((i+1)/2)=q;
        else
            VB((i+1)/2)=H1(2*(a-1)+i,VV(1))-H2(i,1);
        end
    end
end
valorb=max(VB)+1;
B=ones(max(ViQ),1)*max(valorb,1);
while s2<1,
    i=a;
    B1=B;
    if min(p-a+1,max(ViQ))*min(q-B(1)+1,max(VjQ))>limite,
        b=B;

```

```

while i<min(max(ViQ)+a,p+1),
    s1=0;
    while s1<1,
        j=1;
        b1=b;
        while j<c2(i-a+1)+1,
            L1=find(H1(2*i-1,:)>H2(2*(i-a+1)-1,j)+b(i-a+1)-1);
            if length(L1)<1,
                else
                    if L1(1)==1,
                        b(i-a+1)=H1(2*i-1,L1(1))-H2(2*(i-a+1)-1,j)+1;
                    else
                        if H1(2*i,L1(1)-1)+H1(2*i-1,L1(1)-1)<H2(2*(i-a+1)-
1,j)+b(i-a+1)-1+H2(2*(i-a+1),j),
                            L2=find([zeros(1,L1(1)-
1),H1(2*i,L1(1):c1(i))]>=H2(2*(i-a+1),j));
                            b(i-a+1)=H1(2*i-1,L2(1))-H2(2*(i-a+1)-1,j)+1;
                        else
                            end
                        end
                    end
                end
            end
            j=j+1;
        end
        if b(i-a+1)==b1(i-a+1);
            s1=1;
        else
            end
        end
        B(i-a+1)=b(i-a+1);
        i=i+1;
    end
else
    end
if B==B1;

```

```

        s2=1;
    else
        bm=max(B);
        B=bm*ones(max(ViQ));
    end
end
end
if min(p-a+1,max(ViQ))*min(q-B(1)+1,max(VjQ))>limite,
    K=[a,B(1),min(p-a+1,max(ViQ))*min(q-B(1)+1,max(VjQ)),f,e];
    limite=K(3);
else
end
end
a=a+1;
end
else
end
end
else
end
end
end

if K(3)<1
    e=1;
    while e<length(c)+1,
        ViP=Vi(e,:);
        VjP=Vj(e,:);
        f=1;
        while f<ROTACIONES(e)+1,
            ViQ=ViP((f-1)*c(e)+1:f*c(e));
            VjQ=VjP((f-1)*c(e)+1:f*c(e));
            if max(VjQ)+q<n+1,
                if max(ViQ)<m+1,
                    K=[1,q+1,0,f,e];
                    e=length(c);
                    f=ROTACIONES(e);
                end
            end
        end
    end
end

```



```
        else
        end
    else
    end
    f=f+1;
end
e=e+1;
end
else
end

if K(3)<1
    e=1;
    while e<length(c)+1,
        ViP=Vi(e,:);
        VjP=Vj(e,:);
        f=1;
        while f<ROTACIONES(e)+1,
            ViQ=ViP((f-1)*c(e)+1:f*c(e));
            VjQ=VjP((f-1)*c(e)+1:f*c(e));
            if max(ViQ)+p<m+1,
                if max(VjQ)<n+1,
                    K=[p+1,1,0,f,e];
                    e=length(c);
                    f=ROTACIONES(e);
                else
                end
            else
            end
            f=f+1;
        end
        e=e+1;
    end
end
else
```

end

INFOR=K;

A.2.8. Archivo prueba.m

```
function [H]=prueba(m,n,numeroformas,margen)
```

```
k=1;
```

```
while k<numeroformas+1
```

```
    [A1,cant,orientacionz]=matriz(k);
```

```
    A=separacion(A1,margen);
```

```
    S=sparse(A);
```

```
    [Vi0,Vj0,s]=find(S);
```

```
    A90=rot90(A);
```

```
    if max(Vi0)==max(Vj0),
```

```
        if A90==A,
```

```
            if orientacionz>0,
```

```
                SA=orientacionznegativa(A);
```

```
                if SA==A,
```

```
                    Vi1=Vi0; Vi2=0; Vi3=0; Vi4=0; Vi5=0; Vi6=0; Vi7=0; Vi8=0;
```

```
                    Vj1=Vj0; Vj2=0; Vj3=0; Vj4=0; Vj5=0; Vj6=0; Vj7=0; Vj8=0;
```

```
                    ROTACIONES(k)=1;
```

```
                    Correspondencia(k,1:8)=[1,0,0,0,0,0,0,0];
```

```
                else
```

```
                    SS=sparse(SA);
```

```
                    [ViS0,VjS0,s]=find(SS);
```

```
                    Vi1=Vi0; Vi2=ViS0; Vi3=0; Vi4=0; Vi5=0; Vi6=0; Vi7=0; Vi8=0;
```

```
                    Vj1=Vi0; Vj2=VjS0; Vj3=0; Vj4=0; Vj5=0; Vj6=0; Vj7=0; Vj8=0;
```

```
                    ROTACIONES(k)=2;
```

```
                    Correspondencia(k,1:8)=[1,5,0,0,0,0,0,0];
```

```
                end
```

```
            else
```

```
                Vi1=Vi0; Vi2=0; Vi3=0; Vi4=0; Vi5=0; Vi6=0; Vi7=0; Vi8=0;
```

```
                Vj1=Vj0; Vj2=0; Vj3=0; Vj4=0; Vj5=0; Vj6=0; Vj7=0; Vj8=0;
```

```

ROTACIONES(k)=1;
Correspondencia(k,1:8)=[1,0,0,0,0,0,0,0];

end
else
S90=sparse(A90);
[Vi90,Vj90,s]=find(S90);
A180=rot180(A);
if A180==A,
    if orientacionz>0,
        SA=orientacionznegativa(A);
        if SA==A,
            Vi1=Vi0; Vi2=Vi90; Vi3=0; Vi4=0; Vi5=0; Vi6=0; Vi7=0; Vi8=0;
            Vj1=Vj0; Vj2=Vj90; Vj3=0; Vj4=0; Vj5=0; Vj6=0; Vj7=0; Vj8=0;
            ROTACIONES(k)=2;
            Correspondencia(k,1:8)=[1,2,0,0,0,0,0,0];
        else
            if SA==A90,
                Vi1=Vi0; Vi2=Vi90; Vi3=0; Vi4=0; Vi5=0; Vi6=0; Vi7=0; Vi8=0;
                Vj1=Vi0; Vj2=Vj90; Vj3=0; Vj4=0; Vj5=0; Vj6=0; Vj7=0; Vj8=0;
                ROTACIONES(k)=2;
                Correspondencia(k,1:8)=[1,2,0,0,0,0,0,0];
            else
                SS=sparse(SA);
                [ViS0,VjS0,s]=find(SS);
                SA90=rot90(SA);
                SS90=sparse(SA90);
                [ViS90,VjS90,s]=find(SS90);
                Vi1=Vi0; Vi2=Vi90; Vi3=ViS0; Vi4=ViS90; Vi5=0; Vi6=0; Vi7=0;
Vi8=0;
                Vj1=Vi0; Vj2=Vj90; Vj3=VjS0; Vj4=VjS90; Vj5=0; Vj6=0; Vj7=0;
Vj8=0;
                ROTACIONES(k)=4;
                Correspondencia(k,1:8)=[1,2,5,6,0,0,0,0];
            end
        end
    end
end

```

```

        end
    end
else
    Vi1=Vi0; Vi2=Vi90; Vi3=0; Vi4=0; Vi5=0; Vi6=0; Vi7=0; Vi8=0;
    Vj1=Vj0; Vj2=Vj90; Vj3=0; Vj4=0; Vj5=0; Vj6=0; Vj7=0; Vj8=0;
    ROTACIONES(k)=2;
    Correspondencia(k,1:8)=[1,2,0,0,0,0,0,0];

end

else
    S180=sparse(A180);
    [Vi180,Vj180,s]=find(S180);
    A270=rot270(A);
    S270=sparse(A270);
    [Vi270,Vj270,s]=find(S270);
    if orientacionz>0,
        SA=orientacionznegativa(A);
        if SA==A,
            Vi1=Vi0; Vi2=Vi90; Vi3=Vi180; Vi4=Vi270; Vi5=0; Vi6=0; Vi7=0;
Vi8=0;
            Vj1=Vj0; Vj2=Vj90; Vj3=Vj180; Vj4=Vj270; Vj5=0; Vj6=0; Vj7=0;
Vj8=0;
            ROTACIONES(k)=4;
            Correspondencia(k,1:8)=[1,2,3,4,0,0,0,0];
        else
            if SA==A90,
                Vi1=Vi0; Vi2=Vi90; Vi3=Vi180; Vi4=Vi270; Vi5=0; Vi6=0; Vi7=0;
Vi8=0;
                Vj1=Vj0; Vj2=Vj90; Vj3=Vj180; Vj4=Vj270; Vj5=0; Vj6=0; Vj7=0;
Vj8=0;
                ROTACIONES(k)=4;
                Correspondencia(k,1:8)=[1,2,3,4,0,0,0,0];
            else

```

```

    if SA==A180,
        Vi1=Vi0; Vi2=Vi90; Vi3=Vi180; Vi4=Vi270; Vi5=0; Vi6=0;
Vi7=0; Vi8=0;
        Vj1=Vj0; Vj2=Vj90; Vj3=Vj180; Vj4=Vj270; Vj5=0; Vj6=0;
Vi7=0; Vj8=0;
        ROTACIONES(k)=4;
        Correspondencia(k,1:8)=[1,2,3,4,0,0,0,0];
    else
        if SA==A270,
            Vi1=Vi0; Vi2=Vi90; Vi3=Vi180; Vi4=Vi270; Vi5=0; Vi6=0;
Vi7=0; Vi8=0;
            Vj1=Vj0; Vj2=Vj90; Vj3=Vj180; Vj4=Vj270; Vj5=0; Vj6=0;
Vi7=0; Vj8=0;
            ROTACIONES(k)=4;
            Correspondencia(k,1:8)=[1,2,3,4,0,0,0,0];
        else
            SS=sparse(SA);
            [ViS0,VjS0,s]=find(SS);
            SA90=rot90(SA);
            SS90=sparse(SA90);
            [ViS90,VjS90,s]=find(SS90);
            SA180=rot180(SA);
            SS180=sparse(SA180);
            [ViS180,VjS180,s]=find(SS180);
            SA270=rot270(SA);
            SS270=sparse(SA270);
            [ViS270,VjS270,s]=find(SS270);
            Vi1=Vi0; Vi2=Vi90; Vi3=Vi180; Vi4=Vi270; Vi5=ViS0;
Vi6=ViS90; Vi7=ViS180; Vi8=ViS270;
            Vj1=Vi0; Vj2=Vj90; Vj3=Vj180; Vj4=Vj270; Vj5=VjS0;
Vi6=VjS90; Vj7=VjS180; Vj8=VjS270;
            ROTACIONES(k)=8;
            Correspondencia(k,1:8)=[1,2,3,4,5,6,7,8];
        end

```

```

        end
    end
end
else
    Vi1=Vi0; Vi2=Vi90; Vi3=Vi180; Vi4=Vi270; Vi5=0; Vi6=0; Vi7=0;
Vi8=0;
    Vj1=Vj0; Vj2=Vj90; Vj3=Vj180; Vj4=Vj270; Vj5=0; Vj6=0; Vj7=0;
Vj8=0;
    ROTACIONES(k)=4;
    Correspondencia(k,1:8)=[1,2,3,4,0,0,0,0];
end

end

end
else
    S90=sparse(A90);
    [Vi90,Vj90,s]=find(S90);
    A180=rot180(A);
    if A180==A,
        if orientacionz>0,
            SA=orientacionznegativa(A);
            if SA==A,
                Vi1=Vi0; Vi2=Vi90; Vi3=0; Vi4=0; Vi5=0; Vi6=0; Vi7=0; Vi8=0;
                Vj1=Vj0; Vj2=Vj90; Vj3=0; Vj4=0; Vj5=0; Vj6=0; Vj7=0; Vj8=0;
                ROTACIONES(k)=2;
                Correspondencia(k,1:8)=[1,2,0,0,0,0,0,0];
            else
                SS=sparse(SA);
                [ViS0,VjS0,s]=find(SS);
                SA90=rot90(SA);
                SS90=sparse(SA90);
                [ViS90,VjS90,s]=find(SS90);
                Vi1=Vi0; Vi2=Vi90; Vi3=ViS0; Vi4=ViS90; Vi5=0; Vi6=0; Vi7=0;
Vi8=0;

```

```

    Vj1=Vi0; Vj2=Vj90; Vj3=VjS0; Vj4=VjS90; Vj5=0; Vj6=0; Vj7=0;
Vj8=0;
    ROTACIONES(k)=4;
    Correspondencia(k,1:8)=[1,2,5,6,0,0,0,0];
end
else
    Vi1=Vi0; Vi2=Vi90; Vi3=0; Vi4=0; Vi5=0; Vi6=0; Vi7=0; Vi8=0;
    Vj1=Vj0; Vj2=Vj90; Vj3=0; Vj4=0; Vj5=0; Vj6=0; Vj7=0; Vj8=0;
    ROTACIONES(k)=2;
    Correspondencia(k,1:8)=[1,2,0,0,0,0,0,0];

end

%     ROTACIONES(k)=2;
%     Vi180=0;Vi270=0;
%     Vj180=0;Vj270=0;
else
    S180=sparse(A180);
    [Vi180,Vj180,s]=find(S180);
    A270=rot270(A);
    S270=sparse(A270);
    [Vi270,Vj270,s]=find(S270);
    if orientacionz>0,
        SA=orientacionznegativa(A);
        if SA==A,
            Vi1=Vi0; Vi2=Vi90; Vi3=Vi180; Vi4=Vi270; Vi5=0; Vi6=0; Vi7=0;
Vi8=0;
            Vj1=Vj0; Vj2=Vj90; Vj3=Vj180; Vj4=Vj270; Vj5=0; Vj6=0; Vj7=0;
Vj8=0;
            ROTACIONES(k)=4;
            Correspondencia(k,1:8)=[1,2,3,4,0,0,0,0];
        else
            if SA==A180,

```

```

    Vi1=Vi0; Vi2=Vi90; Vi3=Vi180; Vi4=Vi270; Vi5=0; Vi6=0; Vi7=0;
Vi8=0;
    Vj1=Vj0; Vj2=Vj90; Vj3=Vj180; Vj4=Vj270; Vj5=0; Vj6=0; Vj7=0;
Vj8=0;

    ROTACIONES(k)=4;
    Correspondencia(k,1:8)=[1,2,3,4,0,0,0,0];
else
    SS=sparse(SA);
    [ViS0,VjS0,s]=find(SS);
    SA90=rot90(SA);
    SS90=sparse(SA90);
    [ViS90,VjS90,s]=find(SS90);
    SA180=rot180(SA);
    SS180=sparse(SA180);
    [ViS180,VjS180,s]=find(SS180);
    SA270=rot270(SA);
    SS270=sparse(SA270);
    [ViS270,VjS270,s]=find(SS270);
    Vi1=Vi0; Vi2=Vi90; Vi3=Vi180; Vi4=Vi270; Vi5=ViS0; Vi6=ViS90;
Vi7=ViS180; Vi8=ViS270;
    Vj1=Vj0; Vj2=Vj90; Vj3=Vj180; Vj4=Vj270; Vj5=VjS0; Vj6=VjS90;
Vj7=VjS180; Vj8=VjS270;
    ROTACIONES(k)=8;
    Correspondencia(k,1:8)=[1,2,3,4,5,6,7,8];

end

end

else
    Vi1=Vi0; Vi2=Vi90; Vi3=Vi180; Vi4=Vi270; Vi5=0; Vi6=0; Vi7=0; Vi8=0;
    Vj1=Vj0; Vj2=Vj90; Vj3=Vj180; Vj4=Vj270; Vj5=0; Vj6=0; Vj7=0; Vj8=0;
    ROTACIONES(k)=4;
    Correspondencia(k,1:8)=[1,2,3,4,0,0,0,0];
end

```



```

    end
end
Vi(k,1:(length(Vi1)+length(Vi2)+length(Vi3)+length(Vi4)+length(Vi5)+length(Vi6)+length(Vi7)+length(Vi8)))=[Vi1',Vi2',Vi3',Vi4',Vi5',Vi6',Vi7',Vi8'];
Vj(k,1:(length(Vj1)+length(Vj2)+length(Vj3)+length(Vj4)+length(Vj5)+length(Vj6)+length(Vj7)+length(Vj8)))=[Vj1',Vj2',Vj3',Vj4',Vj5',Vj6',Vj7',Vj8'];

c(k)=length(Vi0);
cantidad(k)=cant;
k=k+1;
end
INFO=esq(Vi,Vj,c,ROTACIONES)
tiempodeejecucion=cputime
if INFO(1,1)>0,
    H(1,1:4)=[1,1,Correspondencia(INFO(1,1),INFO(1,2)),INFO(1,1)];
    cantidad(INFO(1,1))=cantidad(INFO(1,1))-1;
    if cantidad(INFO(1,1))<1,
        Vi(INFO(1,1),:)=zeros(1,length(Vi(INFO(1,1),:)));
    else
    end
else
    H(1,1:4)=[m,n,0,0];
end

Di=Vi(H(1,4),(INFO(1,2)-1)*c(H(1,4))+1:INFO(1,2)*c(H(1,4)));
Dj=Vj(H(1,4),(INFO(1,2)-1)*c(H(1,4))+1:INFO(1,2)*c(H(1,4)));
Ds=ones(1,length(Di));
j=2;
tiempoporpieza=cputime;
while max(cantidad)>0,
    D=sparse(Di,Dj,Ds);

    full(D);
    [INFOR]=prox2(Vi,Vj,D,c,m,n,ROTACIONES,cantidad);

```

INFOR

if INFOR(5)>0,

H(j,1:4)=[INFOR(1),INFOR(2),Correspondencia(INFOR(5),INFOR(4)),INFOR(5)];

Die=Vi(H(j,4),(INFOR(4)-1)*c(H(j,4))+1:INFOR(4)*c(H(j,4)));

Dje=Vj(H(j,4),(INFOR(4)-1)*c(H(j,4))+1:INFOR(4)*c(H(j,4)));

Di=[Di,Die+INFOR(1)-1];

Dj=[Dj,Dje+INFOR(2)-1];

Ds=ones(1,length(Di));

cantidad(INFOR(5))=cantidad(INFOR(5))-1;

if cantidad(INFOR(5))==0,

Vi(INFOR(5),:)=zeros(1,length(Vi(INFOR(5),:)));

else

end

else

H(j,1:4)=[INFOR(1),INFOR(2),0,0];

cantidad=zeros(length(cantidad))

end

j=j+1;

tiempoporpieza=cputime-tiempoporpieza

tiempoporpieza=cputime;

end

tiempodeejecucion=cputime-tiempodeejecucion

[Archivo,error]=fopen('C:\ Archivos de módulo\resultado.resul','a');

fprintf(Archivo,'%6.0f\r\n %6.0d\r\n %6.0f\r\n %6.0f\r\n',H');

fclose(Archivo);

A.3. CÓDIGOS BASIC

A.3.1. Formulario frmAgregarFormas

Dim Archivo() As String

Public NumeroArchivos As Double

Public Pieza As String

Option Explicit

Private Sub cmdOrientacionAutomatica_Click()

ReDim Preserve Archivo(1 To NumeroArchivos + 1)

Archivo(NumeroArchivos + 1) = txtPathForma.Text

Pieza = txtPathForma.Text

NumeroArchivos = NumeroArchivos + 1

hide

frmOrientacionAutomatica.Show

End Sub

Private Sub cmdOrientacionManual_Click()

ReDim Preserve Archivo(1 To NumeroArchivos + 1)

Archivo(NumeroArchivos + 1) = txtPathForma.Text

Pieza = txtPathForma.Text

NumeroArchivos = NumeroArchivos + 1

hide

frmOrientacionManual.Show

End Sub

Private Sub cmdVolver4_Click()

hide

```
frmAgregarformas.Show
```

```
End Sub
```

A.3.2. Formulario frmDimensionesHoja

```
Dim margen As String
```

```
Private Sub cmdSiguiete2_Click()
```

```
If Val(txtX.Text) * Val(txtY.Text) = 0 Then
```

```
hide
```

```
frmIntroduzcaDimensiones.Show
```

```
Else
```

```
Open "C:\Archivos de módulo\disposicion.m" For Output As #1
```

```
Print #1, "function H=disposicion", Chr(13) + Chr(10), "m=", txtX.Text, ";n=",  
txtY.Text, ";numerodeformas=", frmSeleccionarForma1.NumeroArchivos, ";margen=",  
margen, ";", Chr(13) + Chr(10), "H=prueba(m,n,numerodeformas,margen);"
```

```
Close #1
```

```
Open "C:\Archivos de módulo\" & frmInicio.CarpetaDisposicion & "\" &  
"dimensiones.hoja" For Output As #2
```

```
Print #2, txtX.Text, Chr(13) + Chr(10), txtY.Text, Chr(13) + Chr(10), margen
```

```
Close #2
```

```
End If
```

```
End Sub
```

```
Private Sub cmdVolver2_Click()
```

```
hide
```

```
frmAgregarformas.Show
```

```
End Sub
```

```
Private Sub txtMargen_Change()
```

margen = txtMargen.Text

End Sub

A.3.3. Formulario frmInicio

Dim alfa1 As Double

Dim valora1 As String

Dim valora As Double

Dim valorb1 As String

Dim valorb As Double

Dim numeroforma1 As String

Dim numeroforma As Double

Dim numerogiro1 As String

Dim numerogiro As Double

Dim xmaxima1 As String

Dim xmaxima As Double

Dim ymaxima1 As String

Dim ymaxima As Double

Dim xminima1 As String

Dim xminima As Double

Dim yminima1 As String

Dim yminima As Double

Dim hojax1 As String

Dim hojax As Double

Dim hojay1 As String

Dim hojay As Double

Dim valori1 As String

Dim valori As Double

Dim valorj1 As String

Dim valorj As Double

Dim Escala As Double

Dim margen1 As String

```
Dim margen As Double
Public CarpetaDisposicion As String

Private Sub cmdCargardisposicion_Click()

If CarpetaDisposicion = "" Then
hide
frmIntroduzcaCarpeta.Show

Else
Open "C:\Archivos de módulo\" & frmInicio.CarpetaDisposicion & "\" &
"dimensiones.hoja" For Input As #5
Open "C:\Archivos de módulo\" & frmInicio.CarpetaDisposicion & "\" &
"resultado.resul" For Input As #1
Line Input #5, hojax1
Line Input #5, hojay1
Line Input #5, margen1
hojax = Val(hojax1)
hojay = Val(hojay1)
margen = margen1
Close #5
'Añadido
Dim documents1 As Documents
Set documents1 = CATIA.Documents

Dim drawingDocument1 As DrawingDocument
Set drawingDocument1 = documents1.Add("Drawing")

drawingDocument1.Standard = catISO

Dim drawingSheets1 As DrawingSheets
Set drawingSheets1 = drawingDocument1.Sheets

Dim drawingSheet1 As DrawingSheet
```

```
Set drawingSheet1 = drawingSheets1.Item("Sheet.1")
```

```
drawingSheet1.PaperSize = catPaperA4
```

```
drawingSheet1.[Scale] = 1#
```

```
If hojax / 297 < hojay / 210 Then
```

```
    Escala = hojay / 210
```

```
Else
```

```
    Escala = hojax / 297
```

```
End If
```

```
drawingSheet1.Orientation = catPaperLandscape
```

```
drawingSheet1.Activate
```

```
Dim specsAndGeomWindow1 As SpecsAndGeomWindow
```

```
Set specsAndGeomWindow1 = CATIA.ActiveWindow
```

```
Dim specsViewer1 As SpecsViewer
```

```
Set specsViewer1 = specsAndGeomWindow1.ActiveViewer
```

```
specsViewer1.Reframe
```

```
Dim drawingViews1 As DrawingViews
```

```
Set drawingViews1 = drawingSheet1.Views
```

```
Dim drawingView1 As DrawingView
```

```
Set drawingView1 = drawingViews1.Add("AutomaticNaming")
```

```
drawingView1.X = 0#
```

```
drawingView1.Y = 0#
```

```
drawingView1.Activate
```

drawingSheet1.Activate

Dim factory2D1 As Factory2D

Set factory2D1 = drawingView1.Factory2D

Dim line2D1() As Line2D

Dim line2D2() As Line2D

Dim line2D3() As Line2D

Dim line2D4() As Line2D

Dim lineT As Line2D

Dim lineB As Line2D

Dim lineL As Line2D

Dim lineR As Line2D

Set lineT = factory2D1.CreateLine(0, hojay / Escala, hojax / Escala, hojay / Escala)

Set lineB = factory2D1.CreateLine(0, 0, hojax / Escala, 0)

Set lineL = factory2D1.CreateLine(0, 0, 0, hojay / Escala)

Set lineR = factory2D1.CreateLine(hojax / Escala, 0, hojax / Escala, hojay / Escala)

'frmCargarDisposicion.imaDisposicion.Scale (0, 0)-(hojax, -hojay)

'alfa1 = 0

linea = 1

Do While Not EOF(1)

'While alfa1 < 1

Line Input #1, valora1

Line Input #1, valorb1

Line Input #1, numerogiro1

Line Input #1, numeroforma1

valora = Val(valora1)

valorb = Val(valorb1)

numerogiro = Val(numerogiro1)

numeroforma = Val(numeroforma1)

If numeroforma = 0 Then

'alfa1 = 1

Else

Open "C:\Archivos de módulo\" & frmInicio.CarpetaDisposicion & "\" & numeroforma
& ".size" For Input As #2

Line Input #2, xminima1

Line Input #2, xmaxima1

Line Input #2, yminima1

Line Input #2, ymaxima1

xminima = Val(xminima1)

yminima = Val(yminima1)

xmaxima = Val(xmaxima1)

ymaxima = Val(ymaxima1)

Close #2

Open "C:\Archivos de módulo\" & frmInicio.CarpetaDisposicion & "\" & numeroforma
& ".eli" For Input As #3

Open "C:\Archivos de módulo\" & frmInicio.CarpetaDisposicion & "\" & numeroforma
& ".elj" For Input As #4

If numerogiro = 1 Then

Do While Not EOF(3)

Line Input #3, valori1

Line Input #4, valorj1

valori = Val(valori1) - yminima + 1

valorj = Val(valorj1) - xminima + 1

ReDim line2D1(1 To linea) As Line2D

ReDim line2D2(1 To linea) As Line2D

ReDim line2D3(1 To linea) As Line2D

ReDim line2D4(1 To linea) As Line2D

Set line2D1(linea) = factory2D1.CreateLine((valorj + valorb - 2 + margen) / Escala, (-2
+ valori + valora + margen) / Escala, (valorj + valorb - 2 + margen) / Escala, (-1 +
valori + valora + margen) / Escala)

Set line2D2(linea) = factory2D1.CreateLine((valorj + valorb - 2 + margen) / Escala, (-1
+ valori + valora + margen) / Escala, (valorj + valorb - 1 + margen) / Escala, (-1 +
valori + valora + margen) / Escala)

```
Set line2D3(linea) = factory2D1.CreateLine((valorj + valorb - 1 + margen) / Escala, (-1 + valori + valora + margen) / Escala, (valorj + valorb - 1 + margen) / Escala, (-2 + valori + valora + margen) / Escala)
```

```
Set line2D4(linea) = factory2D1.CreateLine((valorj + valorb - 1 + margen) / Escala, (-2 + valori + valora + margen) / Escala, (valorj + valorb - 2 + margen) / Escala, (-2 + valori + valora + margen) / Escala)
```

```
linea = linea + 1
```

```
'frmCargarDisposicion.imaDisposicion.Line (valorj + valorb - 2, 2 - valori - valora)-(valorj + valorb - 1, 1 - valori - valora), RGB(255 / numeroforma, 255 * Sin(numeroforma), 255 * Cos(numeroforma)), BF
```

```
Loop
```

```
Close #3
```

```
Close #4
```

```
ElseIf numerogiro = 2 Then
```

```
Do While Not EOF(3)
```

```
Line Input #3, valori1
```

```
Line Input #4, valorj1
```

```
valori = Val(valori1) - yminima + 1
```

```
valorj = Val(valorj1) - xminima + 1
```

```
ReDim line2D1(1 To linea) As Line2D
```

```
ReDim line2D2(1 To linea) As Line2D
```

```
ReDim line2D3(1 To linea) As Line2D
```

```
ReDim line2D4(1 To linea) As Line2D
```

```
Set line2D1(linea) = factory2D1.CreateLine((valori + valorb - 2 + margen) / Escala, (-xminima + xmaxima + valora - valorj + margen) / Escala, (valori + valorb - 2 + margen) / Escala, (-xminima + xmaxima + valora - valorj + 1 + margen) / Escala)
```

```
Set line2D2(linea) = factory2D1.CreateLine((valori + valorb - 2 + margen) / Escala, (-xminima + xmaxima + valora - valorj + 1 + margen) / Escala, (valori + valorb - 1 + margen) / Escala, (-xminima + xmaxima + valora - valorj + 1 + margen) / Escala)
```

```
Set line2D3(linea) = factory2D1.CreateLine((valori + valorb - 1 + margen) / Escala, (-xminima + xmaxima + valora - valorj + 1 + margen) / Escala, (valori + valorb - 1 + margen) / Escala, (-xminima + xmaxima + valora - valorj + margen) / Escala)
```

```
Set line2D4(linea) = factory2D1.CreateLine((valori + valorb - 1 + margen) / Escala, (-
xminima + xmaxima + valora - valorj + margen) / Escala, (valori + valorb - 2 + margen)
/ Escala, (-xminima + xmaxima + valora - valorj + margen) / Escala)
```

```
linea = linea + 1
```

```
'frmCargarDisposicion.imaDisposicion.Line (valori + valorb - 2, xminima - xmaxima -
valora + valorj)-(valori + valorb - 1, xminima - xmaxima - valora + valorj - 1),
RGB(255 / numeroforma, 255 * Sin(numeroforma), 255 * Cos(numeroforma)), BF
```

```
Loop
```

```
Close #3
```

```
Close #4
```

```
ElseIf numerogiro = 3 Then
```

```
Do While Not EOF(3)
```

```
Line Input #3, valori1
```

```
Line Input #4, valorj1
```

```
valori = Val(valori1) - yminima + 1
```

```
valorj = Val(valorj1) - xminima + 1
```

```
ReDim line2D1(1 To linea) As Line2D
```

```
ReDim line2D2(1 To linea) As Line2D
```

```
ReDim line2D3(1 To linea) As Line2D
```

```
ReDim line2D4(1 To linea) As Line2D
```

```
Set line2D1(linea) = factory2D1.CreateLine((xmaxima - xminima - valorj + valorb +
margen) / Escala, (-yminima - valori + ymaxima + valora + margen) / Escala, (xmaxima
- xminima - valorj + valorb + margen) / Escala, (-yminima - valori + ymaxima + valora
+ 1 + margen) / Escala)
```

```
Set line2D2(linea) = factory2D1.CreateLine((xmaxima - xminima - valorj + valorb +
margen) / Escala, (-yminima - valori + ymaxima + valora + 1 + margen) / Escala,
(xmaxima - xminima - valorj + valorb + 1 + margen) / Escala, (-yminima - valori +
ymaxima + valora + 1 + margen) / Escala)
```

```
Set line2D3(linea) = factory2D1.CreateLine((xmaxima - xminima - valorj + valorb + 1
+ margen) / Escala, (-yminima - valori + ymaxima + valora + 1 + margen) / Escala,
(xmaxima - xminima - valorj + valorb + 1 + margen) / Escala, (-yminima - valori +
ymaxima + valora + margen) / Escala)
```

```
Set line2D4(linea) = factory2D1.CreateLine((xmaxima - xminima - valorj + valorb + 1
+ margen) / Escala, (-yminima - valori + ymaxima + valora + margen) / Escala,
(xmaxima - xminima - valorj + valorb + margen) / Escala, (-yminima - valori +
ymaxima + valora + margen) / Escala)
```

```
linea = linea + 1
```

```
'frmCargarDisposicion.imaDisposicion.Line (xmaxima - xminima - valorj + valorb,
yminima + valori - ymaxima - valora)-(xmaxima - xminima - valorj + valorb + 1,
yminima + valori - ymaxima - valora - 1), RGB(255 / numeroforma, 255 *
Sin(numeroforma), 255 * Cos(numeroforma)), BF
```

```
Loop
```

```
Close #3
```

```
Close #4
```

```
ElseIf numerogiro = 4 Then
```

```
Do While Not EOF(3)
```

```
Line Input #3, valori1
```

```
Line Input #4, valorj1
```

```
valori = Val(valori1) - yminima + 1
```

```
valorj = Val(valorj1) - xminima + 1
```

```
ReDim line2D1(1 To linea) As Line2D
```

```
ReDim line2D2(1 To linea) As Line2D
```

```
ReDim line2D3(1 To linea) As Line2D
```

```
ReDim line2D4(1 To linea) As Line2D
```

```
Set line2D1(linea) = factory2D1.CreateLine((ymaxima - yminima - valori + valorb +
margen) / Escala, (-2 + valora + valorj + margen) / Escala, (ymaxima - yminima - valori
+ valorb + margen) / Escala, (-1 + valora + valorj + margen) / Escala)
```

```
Set line2D2(linea) = factory2D1.CreateLine((ymaxima - yminima - valori + valorb +
margen) / Escala, (-1 + valora + valorj + margen) / Escala, (ymaxima - yminima - valori
+ valorb + 1 + margen) / Escala, (-1 + valora + valorj + margen) / Escala)
```

```
Set line2D3(linea) = factory2D1.CreateLine((ymaxima - yminima - valori + valorb + 1
+ margen) / Escala, (-1 + valora + valorj + margen) / Escala, (ymaxima - yminima -
valori + valorb + 1 + margen) / Escala, (-2 + valora + valorj + margen) / Escala)
```

```
Set line2D4(linea) = factory2D1.CreateLine((ymaxima - yminima - valori + valorb + 1
+ margen) / Escala, (-2 + valora + valorj + margen) / Escala, (ymaxima - yminima -
valori + valorb + margen) / Escala, (-2 + valora + valorj + margen) / Escala)
```

```
linea = linea + 1
```

```
'frmCargarDisposicion.imaDisposicion.Line (ymaxima - yminima - valori + valorb, 2 -
valora - valorj)-(ymaxima - yminima - valori + valorb + 1, 1 - valora - valorj), RGB(255
/ numeroforma, 255 * Sin(numeroforma), 255 * Cos(numeroforma)), BF
```

```
Loop
```

```
Close #3
```

```
Close #4
```

```
ElseIf numerogiro = 5 Then
```

```
Do While Not EOF(3)
```

```
Line Input #3, valori1
```

```
Line Input #4, valorj1
```

```
valori = Val(valori1) - yminima + 1
```

```
valorj = Val(valorj1) - xminima + 1
```

```
valori = ymaxima - yminima - valori + 2
```

```
ReDim line2D1(1 To linea) As Line2D
```

```
ReDim line2D2(1 To linea) As Line2D
```

```
ReDim line2D3(1 To linea) As Line2D
```

```
ReDim line2D4(1 To linea) As Line2D
```

```
Set line2D1(linea) = factory2D1.CreateLine((valorj + valorb - 2 + margen) / Escala, (-2
+ valori + valora + margen) / Escala, (valorj + valorb - 2 + margen) / Escala, (-1 +
valori + valora + margen) / Escala)
```

```
Set line2D2(linea) = factory2D1.CreateLine((valorj + valorb - 2 + margen) / Escala, (-1
+ valori + valora + margen) / Escala, (valorj + valorb - 1 + margen) / Escala, (-1 +
valori + valora + margen) / Escala)
```

```
Set line2D3(linea) = factory2D1.CreateLine((valorj + valorb - 1 + margen) / Escala, (-1
+ valori + valora + margen) / Escala, (valorj + valorb - 1 + margen) / Escala, (-2 +
valori + valora + margen) / Escala)
```

```
Set line2D4(linea) = factory2D1.CreateLine((valorj + valorb - 1 + margen) / Escala, (-2 + valori + valora + margen) / Escala, (valorj + valorb - 2 + margen) / Escala, (-2 + valori + valora + margen) / Escala)
```

```
linea = linea + 1
```

```
'frmCargarDisposicion.imaDisposicion.Line (valorj + valorb - 2, 2 - valori - valora)-(valorj + valorb - 1, 1 - valori - valora), RGB(255 / numeroforma, 255 * Sin(numeroforma), 255 * Cos(numeroforma)), BF
```

```
Loop
```

```
Close #3
```

```
Close #4
```

```
ElseIf numerogiro = 6 Then
```

```
Do While Not EOF(3)
```

```
Line Input #3, valori1
```

```
Line Input #4, valorj1
```

```
valori = Val(valori1) - yminima + 1
```

```
valorj = Val(valorj1) - xminima + 1
```

```
valori = ymaxima - yminima - valori + 2
```

```
ReDim line2D1(1 To linea) As Line2D
```

```
ReDim line2D2(1 To linea) As Line2D
```

```
ReDim line2D3(1 To linea) As Line2D
```

```
ReDim line2D4(1 To linea) As Line2D
```

```
Set line2D1(linea) = factory2D1.CreateLine((valori + valorb - 2 + margen) / Escala, (-xminima + xmaxima + valora - valorj + margen) / Escala, (valori + valorb - 2 + margen) / Escala, (-xminima + xmaxima + valora - valorj + 1 + margen) / Escala)
```

```
Set line2D2(linea) = factory2D1.CreateLine((valori + valorb - 2 + margen) / Escala, (-xminima + xmaxima + valora - valorj + 1 + margen) / Escala, (valori + valorb - 1 + margen) / Escala, (-xminima + xmaxima + valora - valorj + 1 + margen) / Escala)
```

```
Set line2D3(linea) = factory2D1.CreateLine((valori + valorb - 1 + margen) / Escala, (-xminima + xmaxima + valora - valorj + 1 + margen) / Escala, (valori + valorb - 1 + margen) / Escala, (-xminima + xmaxima + valora - valorj + margen) / Escala)
```

```
Set line2D4(linea) = factory2D1.CreateLine((valori + valorb - 1 + margen) / Escala, (-
xminima + xmaxima + valora - valorj + margen) / Escala, (valori + valorb - 2 + margen)
/ Escala, (-xminima + xmaxima + valora - valorj + margen) / Escala)
```

```
linea = linea + 1
```

```
'frmCargarDisposicion.imaDisposicion.Line (valori + valorb - 2, xminima - xmaxima -
valora + valorj)-(valori + valorb - 1, xminima - xmaxima - valora + valorj - 1),
RGB(255 / numeroforma, 255 * Sin(numeroforma), 255 * Cos(numeroforma)), BF
```

```
Loop
```

```
Close #3
```

```
Close #4
```

```
ElseIf numerogiro = 7 Then
```

```
Do While Not EOF(3)
```

```
Line Input #3, valori1
```

```
Line Input #4, valorj1
```

```
valori = Val(valori1) - yminima + 1
```

```
valorj = Val(valorj1) - xminima + 1
```

```
valori = ymaxima - yminima - valori + 2
```

```
ReDim line2D1(1 To linea) As Line2D
```

```
ReDim line2D2(1 To linea) As Line2D
```

```
ReDim line2D3(1 To linea) As Line2D
```

```
ReDim line2D4(1 To linea) As Line2D
```

```
Set line2D1(linea) = factory2D1.CreateLine((xmaxima - xminima - valorj + valorb +
margen) / Escala, (-yminima - valori + ymaxima + valora + margen) / Escala, (xmaxima
- xminima - valorj + valorb + margen) / Escala, (-yminima - valori + ymaxima + valora
+ 1 + margen) / Escala)
```

```
Set line2D2(linea) = factory2D1.CreateLine((xmaxima - xminima - valorj + valorb +
margen) / Escala, (-yminima - valori + ymaxima + valora + 1 + margen) / Escala,
(xmaxima - xminima - valorj + valorb + 1 + margen) / Escala, (-yminima - valori +
ymaxima + valora + 1 + margen) / Escala)
```

```
Set line2D3(linea) = factory2D1.CreateLine((xmaxima - xminima - valorj + valorb + 1
+ margen) / Escala, (-yminima - valori + ymaxima + valora + 1 + margen) / Escala,
```

(xmaxima - xminima - valorj + valorb + 1 + margen) / Escala, (-yminima - valori + ymaxima + valora + margen) / Escala)

Set line2D4(linea) = factory2D1.CreateLine((xmaxima - xminima - valorj + valorb + 1 + margen) / Escala, (-yminima - valori + ymaxima + valora + margen) / Escala, (xmaxima - xminima - valorj + valorb + margen) / Escala, (-yminima - valori + ymaxima + valora + margen) / Escala)

linea = linea + 1

'frmCargarDisposicion.imaDisposicion.Line (xmaxima - xminima - valorj + valorb, yminima + valori - ymaxima - valora)-(xmaxima - xminima - valorj + valorb + 1, yminima + valori - ymaxima - valora - 1), RGB(255 / numeroforma, 255 * Sin(numeroforma), 255 * Cos(numeroforma)), BF

Loop

Close #3

Close #4

ElseIf numerogiro = 8 Then

Do While Not EOF(3)

Line Input #3, valori1

Line Input #4, valorj1

valori = Val(valori1) - yminima + 1

valorj = Val(valorj1) - xminima + 1

valori = ymaxima - yminima - valori + 2

ReDim line2D1(1 To linea) As Line2D

ReDim line2D2(1 To linea) As Line2D

ReDim line2D3(1 To linea) As Line2D

ReDim line2D4(1 To linea) As Line2D

Set line2D1(linea) = factory2D1.CreateLine((ymaxima - yminima - valori + valorb + margen) / Escala, (-2 + valora + valorj + margen) / Escala, (ymaxima - yminima - valori + valorb + margen) / Escala, (-1 + valora + valorj + margen) / Escala)

Set line2D2(linea) = factory2D1.CreateLine((ymaxima - yminima - valori + valorb + margen) / Escala, (-1 + valora + valorj + margen) / Escala, (ymaxima - yminima - valori + valorb + 1 + margen) / Escala, (-1 + valora + valorj + margen) / Escala)


```
Set line2D3(linea) = factory2D1.CreateLine((ymaxima - yminima - valori + valorb + 1 + margen) / Escala, (-1 + valora + valorj + margen) / Escala, (ymaxima - yminima - valori + valorb + 1 + margen) / Escala, (-2 + valora + valorj + margen) / Escala)
```

```
Set line2D4(linea) = factory2D1.CreateLine((ymaxima - yminima - valori + valorb + 1 + margen) / Escala, (-2 + valora + valorj + margen) / Escala, (ymaxima - yminima - valori + valorb + margen) / Escala, (-2 + valora + valorj + margen) / Escala)
```

```
linea = linea + 1
```

```
'frmCargarDisposicion.imaDisposicion.Line (ymaxima - yminima - valori + valorb, 2 - valora - valorj)-(ymaxima - yminima - valori + valorb + 1, 1 - valora - valorj), RGB(255 / numeroforma, 255 * Sin(numeroforma), 255 * Cos(numeroforma)), BF
```

```
Loop
```

```
Close #3
```

```
Close #4
```

```
End If
```

```
End If
```

```
Loop
```

```
'Wend
```

```
Close #1
```

```
hide
```

```
End If
```

```
End Sub
```

```
Private Sub cmdNuevadisposición_Click()
```

```
If CarpetaDisposicion = "" Then
```

```
hide
```

```
frmIntroduzcaCarpeta.Show
```

```
Else
```

```
Open "C:\Archivos de módulo\matriz.m" For Output As #1
```

```
Print #1, "function [A,cant,orientacionz]=matriz(k);"
```

```
Close #1
```

```
hide
```

```
frmAgregarformas.Show
```

```
End If
```

```
End Sub
```

```
Private Sub txtCarpetaDisposicion_Change()
```

```
CarpetaDisposicion = txtCarpetaDisposicion.Text
```

```
End Sub
```

A.3.4. Formulario frmIntroduzcaCarpeta

```
Option Explicit
```

```
Private Sub cmdAceptar_Click()
```

```
hide
```

```
frmInicio.Show
```

```
End Sub
```

A.3.5. Formulario frmIntroduzcaDimensiones

```
Option Explicit
```

```
Private Sub cmdAceptar_Click()
```

```
hide
```

```
frmDimensionesHoja.Show
```

```
End Sub
```

A.3.6. Formulario frmIntroduzcaNumerodePiezas

Option Explicit

Private Sub cmdAceptar_Click()

hide

frmSeleccionarForma1.Show

End Sub

A.3.7. Formulario frmIntroduzcaPunto

Option Explicit

Private Sub cmdAceptar_Click()

hide

frmSeleccionarForma1.Show

End Sub

A.3.8. Formulario frmIntroduzcaRadio

Option Explicit

Private Sub cmdAceptar_Click()

hide

frmSeleccionarForma1.Show

End Sub

A.3.9. Formulario frmOrientacionAutomatica

Dim Npunto

Dim PuntoX()

Dim PuntoY()

Dim Radio

Dim a() As Variant

Dim Tamaño(1 To 18) As Double

Dim delta2 As Double

Dim delta1 As Double

Dim cantidaddepiezasporforma

Dim Znegativa As Double

Option Explicit

Private Sub cmdActivarZNegativa_Click()

Znegativa = 1

txtEstadoZNegativa.Text = "Activado"

End Sub

Private Sub cmdAgregarFigura_Click()

If Radio = 0 Then

hide

frmIntroduzcaRadio.Show

ElseIf Npunto = 0 Then

hide

frmIntroduzcaPunto.Show

ElseIf cantidaddepiezasporforma = 0 Then

hide

frmIntroduzcaNumerodePiezas.Show

Else

Dim orientacionrad As Double

Dim d1 As Double

Dim d2 As Double

Dim dx As Double

Dim dxprima As Double

Dim dy As Double

Dim dyprima As Double

Dim j

Dim i

Dim xmaxima(1 To 18) As Double

Dim ymaxima(1 To 18) As Double

Dim xminima(1 To 18) As Double

Dim yminima(1 To 18) As Double

Dim grados(1 To 18) As Double

Dim direccionx As Double

Dim direcciony As Double

Dim distancia As Double

ReDim oa(1 To 18, 1 To 2 * Radio, 1 To 2 * Radio) As Double

Dim documents1 As Documents

Set documents1 = CATIA.Documents

Dim partDocument1 As PartDocument

Set partDocument1 = documents1.Open(frmSeleccionarForma1.Pieza)

Dim part1 As Part

Set part1 = partDocument1.Part

Dim hybridShapeFactory1 As HybridShapeFactory

Set hybridShapeFactory1 = part1.HybridShapeFactory

Dim PuntoInspeccion() As HybridShapePointCoord

ReDim PuntoInspeccion(1 To Npunto)

Dim orientacion As Double

orientacion = 1

While orientacion < 19

grados(orientacion) = (orientacion - 1) * 5

xmaxima(orientacion) = 1

ymaxima(orientacion) = 1

xminima(orientacion) = 2 * Radio

yminima(orientacion) = 2 * Radio

Dim alfa

alfa = 1

While alfa < Npunto + 1

Set PuntoInspeccion(alfa) =

hybridShapeFactory1.AddNewPointCoord(Val(PuntoX(alfa)), Val(PuntoY(alfa)), 0#)

Dim bodies1 As Bodies

Set bodies1 = part1.Bodies

Dim body1 As Body

Set body1 = bodies1.Item("PartBody")

body1.InsertHybridShape PuntoInspeccion(alfa)

part1.InWorkObject = PuntoInspeccion(alfa)

part1.Update

i = 1

While i < 2 * Radio + 1

dyprima = Radio + 0.5 - i

```
j = 1
```

```
While j < 2 * Radio + 1
```

```
orientacionrad = Val((orientacion - 1) * 5) * 2 * 3.141592 / 360
```

```
dxprima = j - 0.5 - Radio
```

```
dx = dxprima * Cos(orientacionrad) + dyprima * Sin(orientacionrad)
```

```
dy = -dxprima * Sin(orientacionrad) + dyprima * Cos(orientacionrad)
```

```
d1 = dx - Val(PuntoX(alfa))
```

```
d2 = dy - Val(PuntoY(alfa))
```

```
distancia = Sqr(d1 ^ 2 + d2 ^ 2)
```

```
direccionx = d1 / distancia
```

```
direcciony = d2 / distancia
```

```
Dim vector As HybridShapeDirection
```

```
Set vector = hybridShapeFactory1.AddNewDirectionByCoord(direccionx, direcciony,  
0#)
```

```
Dim vector1 As Variant
```

```
Set vector1 = vector
```

```
Dim hybridShapes1 As HybridShapes
```

```
Set hybridShapes1 = body1.HybridShapes
```

```
Dim superficiedeOperacion As Variant
```

```
Set superficiedeOperacion = hybridShapes1.Item("superficiedeoperacion")
```

```
Dim reference1 As Reference
```

```
Set reference1 = part1.CreateReferenceFromObject(superficiedeOperacion)
```

```
Dim reference2 As Reference
```

```
Set reference2 = part1.CreateReferenceFromObject(PuntoInspeccion(alfa))
```

```
Dim hybridShapePointOnSurface1 As HybridShapePointOnSurface
```

```
Set hybridShapePointOnSurface1 =  
hybridShapeFactory1.AddNewPointOnSurfaceWithReference(reference1, reference2,  
vector1, distancia)
```

```
body1.InsertHybridShape hybridShapePointOnSurface1
```

```
part1.InWorkObject = hybridShapePointOnSurface1
```

```
part1.Update
```

```
Dim ohybridShapePointOnSurface1 As Variant
```

```
Set ohybridShapePointOnSurface1 = hybridShapePointOnSurface1
```

```
Dim Coordinates1(2) As Variant
```

```
ohybridShapePointOnSurface1.GetCoordinates Coordinates1
```

```
If Coordinates1(0) < dx - 0.5 Then
```

```
ElseIf Coordinates1(0) > dx + 0.5 Then
```

```
ElseIf Coordinates1(1) < dy - 0.5 Then
```

```
ElseIf Coordinates1(1) > dy + 0.5 Then
```

```
Else
```

```
oa(orientacion, i, j) = 1
```

```
If j < xminima(orientacion) Then
```

```
xminima(orientacion) = j
```

```
Else
```

```
End If
```

```
If i < yminima(orientacion) Then
```

```
yminima(orientacion) = i
```

```
Else
```

```
End If
```

```
If j > xmaxima(orientacion) Then
```

```
xmaxima(orientacion) = j
```

```
Else
```

```
End If
```



```
If i > ymaxima(orientacion) Then
```

```
ymaxima(orientacion) = i
```

```
Else
```

```
End If
```

```
End If
```

```
j = j + 1
```

```
Wend
```

```
i = i + 1
```

```
Wend
```

```
alfa = alfa + 1
```

```
Wend
```

```
Tamaño(orientacion) = (xmaxima(orientacion) + 1 - xminima(orientacion)) *  
(ymaxima(orientacion) + 1 - yminima(orientacion))
```

```
orientacion = orientacion + 1
```

```
Wend
```

```
partDocument1.Close
```

```
delta2 = 1
```

```
delta1 = 1
```

```
While delta1 < 19
```

```
If Tamaño(delta1) < Tamaño(delta2) Then
```

```
delta2 = delta1
```

```
Else
```

```
End If
```

```
delta1 = delta1 + 1
```

```
Wend
```

```
Dim oA1() As Double
```

```
ReDim oA1(1 To ymaxima(delta2) - yminima(delta2) + 1, 1 To xmaxima(delta2) -  
xminima(delta2) + 1)
```

```
Dim linea As String
```

```
linea = " "
```

```
Open "C:\Archivos de módulo\matriz.mat" For Append As #1
```

```
Print #1, Chr(13) + Chr(10), "if k==", frmSeleccionarForma1.NumeroArchivos, ",",  
Chr(13) + Chr(10), "A=["
```

```
Open "C:\Archivos de módulo\" & frmInicio.CarpetaDisposicion & "\" &  
frmSeleccionarForma1.NumeroArchivos & ".eli" For Output As #2
```

```
Open "C:\Archivos de módulo\" & frmInicio.CarpetaDisposicion & "\" &  
frmSeleccionarForma1.NumeroArchivos & ".elj" For Output As #3
```

```
For i = yminima(delta2) To ymaxima(delta2)
```

```
For j = xminima(delta2) To xmaxima(delta2)
```

```
oA1(i + 1 - yminima(delta2), j + 1 - xminima(delta2)) = oa(delta2, i, j)
```

```
Print #1, oa(delta2, i, j), " "
```

```
linea = linea & oa(delta2, i, j) & " "
```

```
If oa(delta2, i, j) > 0 Then
```

```
Print #2, i
```

```
Print #3, j
```

```
Else
```

```
End If
```

```
Next j
```

```
Print #1, linea
```

```
If i = ymaxima(delta2) Then
```

```
Else
```

```
Print #1, ";"
```

```
End If
```

```
Next i
```

```
Print #1, "];", Chr(13) + Chr(10), "cant=", cantidaddepiezasporforma, ";", Chr(13) +  
Chr(10), "orientacionz=", Znegativa, ";", Chr(13) + Chr(10), "else", Chr(13) + Chr(10),  
"end", Chr(13) + Chr(10)
```

```
Close #1
```

```
Close #2
```

```
Close #3
```

```
Open "C:\Archivos de módulo\" & frmInicio.CarpetaDisposicion & "\" & "relacion.rel"
```

```
For Append As #4
```

```
Print #4, frmSeleccionarForma1.Pieza
```

```
Close #4
```

```
Open "C:\Archivos de módulo\" & frmInicio.CarpetaDisposicion & "\" &  
frmSeleccionarForma1.NumeroArchivos & ".size" For Output As #5
```

```
Print #5, xminima(delta2), Chr(13) + Chr(10), xmaxima(delta2), Chr(13) + Chr(10),  
yminima(delta2), Chr(13) + Chr(10), ymaxima(delta2), Chr(13) + Chr(10)
```

```
Close #5
```

```
ReDim Preserve a(1 To frmSeleccionarForma1.NumeroArchivos)
```

```
a(frmSeleccionarForma1.NumeroArchivos) = oA1
```

```
frmAgregarformas.lstListadeformas.AddItem frmSeleccionarForma1.Pieza
```

```
frmAgregarformas.lstOrientacion.AddItem grados(delta2)
```

```
frmAgregarformas.lstX.AddItem xmaxima(delta2) - xminima(delta2) + 1
```

```
frmAgregarformas.lstY.AddItem ymaxima(delta2) - yminima(delta2) + 1
```

```
hide
```

```
frmAgregarformas.Show
```

```
End If
```

```
End Sub
```

```
Private Sub cmdAgregarPuntoInspeccion_Click()
```

```
ReDim Preserve PuntoX(1 To Npunto + 1)
```

```
ReDim Preserve PuntoY(1 To Npunto + 1)
```

```
PuntoX(Npunto + 1) = txtPuntoInspeccionX.Text
```

```
PuntoY(Npunto + 1) = txtPuntoInspeccionY.Text
```

```
Npunto = Npunto + 1
```

```
txtNumeroPuntos.Text = Npunto
```

```
End Sub
```

```
Private Sub cmdDesactivarZNegativa_Click()
```

```
Znegativa = 0
```

```
txtEstadoZNegativa = "Desactivado"
```

```
End Sub
```

```
Private Sub cmdVolver5_Click()
```

```
frmSeleccionarForma1.NumeroArchivos = frmSeleccionarForma1.NumeroArchivos - 1
```

```
hide
```

```
frmSeleccionarForma1.Show
```

```
End Sub
```

```
Private Sub txtPiezas_Change()
```

```
cantidaddepiezasporforma = txtPiezas.Text
```

```
End Sub
```

```
Private Sub txtRadio_Change()
```

```
Radio = txtRadio.Text
```

```
End Sub
```

A.3.10. Formulario frmOrientacionManual

Dim Npunto

Dim PuntoX()

Dim PuntoY()

Dim orientacion

Dim Radio

Dim cantidaddepiezasporforma

Dim Znegativa As Double

Option Explicit

Private Sub cmdActivarZNegativa_Click()

Znegativa = 1

txtEstadoZNegativa.Text = "Activado"

End Sub

Private Sub cmdAgregarFigura_Click()

If Radio = 0 Then

hide

frmIntroduzcaRadio.Show

ElseIf Npunto = 0 Then

hide

frmIntroduzcaPunto.Show

ElseIf cantidaddepiezasporforma = 0 Then

hide

frmIntroduzcaNumerodePiezas.Show

Else

Dim orientacionrad As Double

Dim d1 As Double

Dim d2 As Double

Dim dx As Double

Dim dxprima As Double

Dim dy As Double

Dim dyprima As Double

Dim j

Dim i

Dim xmaxima As Double

Dim ymaxima As Double

Dim xminima As Double

Dim yminima As Double

xmaxima = 1

ymaxima = 1

xminima = 2 * Radio

yminima = 2 * Radio

Dim direccionx As Double

Dim direcciony As Double

Dim distancia As Double

ReDim oa(1 To 2 * Radio, 1 To 2 * Radio) As Double

Dim documents1 As Documents

Set documents1 = CATIA.Documents

Dim partDocument1 As PartDocument

Set partDocument1 = documents1.Open(frmSeleccionarForma1.Pieza)

Dim part1 As Part

Set part1 = partDocument1.Part

Dim hybridShapeFactory1 As HybridShapeFactory

Set hybridShapeFactory1 = part1.HybridShapeFactory

Dim alfa

alfa = 1

```
Dim PuntoInspeccion() As HybridShapePointCoord
```

```
ReDim PuntoInspeccion(1 To Npunto)
```

```
Dim bodies1 As Bodies
```

```
Set bodies1 = part1.Bodies
```

```
Dim body1 As Body
```

```
Set body1 = bodies1.Item("PartBody")
```

```
Dim vector As HybridShapeDirection
```

```
Dim vector1 As Variant
```

```
Dim hybridShapes1 As HybridShapes
```

```
Dim superficiedeOperacion As Variant
```

```
Dim reference1 As Reference
```

```
Dim reference2 As Reference
```

```
Dim hybridShapePointOnSurface1 As HybridShapePointOnSurface
```

```
Dim ohybridShapePointOnSurface1 As Variant
```

```
Dim Coordinates1(2) As Variant
```

```
While alfa < Npunto + 1
```

```
Set PuntoInspeccion(alfa) =  
hybridShapeFactory1.AddNewPointCoord(Val(PuntoX(alfa)), Val(PuntoY(alfa)), 0#)
```

```
body1.InsertHybridShape PuntoInspeccion(alfa)
```

```
part1.InWorkObject = PuntoInspeccion(alfa)
```

```
part1.Update
```

```
i = 1
```

```
While i < 2 * Radio + 1
```

dyprima = Radio + 0.5 - i

j = 1

While j < 2 * Radio + 1

orientacionrad = Val(orientacion) * 2 * 3.141592 / 360

dxprima = j - 0.5 - Radio

dx = dxprima * Cos(orientacionrad) + dyprima * Sin(orientacionrad)

dy = -dxprima * Sin(orientacionrad) + dyprima * Cos(orientacionrad)

d1 = dx - Val(PuntoX(alfa))

d2 = dy - Val(PuntoY(alfa))

distancia = Sqr(d1 ^ 2 + d2 ^ 2)

direccionx = d1 / distancia

direcciony = d2 / distancia

Set vector = hybridShapeFactory1.AddNewDirectionByCoord(direccionx, direcciony,
0#)

Set vector1 = vector

Set hybridShapes1 = body1.HybridShapes

Set superficiedeOperacion = hybridShapes1.Item("superficiedeoperacion")

Set reference1 = part1.CreateReferenceFromObject(superficiedeOperacion)

Set reference2 = part1.CreateReferenceFromObject(PuntoInspeccion(alfa))

Set hybridShapePointOnSurface1 =
hybridShapeFactory1.AddNewPointOnSurfaceWithReference(reference1, reference2,
vector1, distancia)

body1.InsertHybridShape hybridShapePointOnSurface1

part1.InWorkObject = hybridShapePointOnSurface1

part1.Update

Set ohybridShapePointOnSurface1 = hybridShapePointOnSurface1

ohybridShapePointOnSurface1.GetCoordinates Coordinates1

If Coordinates1(0) < dx - 0.5 Then

ElseIf Coordinates1(0) > dx + 0.5 Then

ElseIf Coordinates1(1) < dy - 0.5 Then

ElseIf Coordinates1(1) > dy + 0.5 Then

Else

oa(i, j) = 1

If j < xminima Then

xminima = j

Else

End If

If i < yminima Then

yminima = i

Else

End If

If j > xmaxima Then

xmaxima = j

Else

End If

If i > ymaxima Then

ymaxima = i

Else

End If

End If

j = j + 1

Wend

i = i + 1

```
Wend
```

```
alfa = alfa + 1
```

```
Wend
```

```
partDocument1.Close
```

```
Dim linea As String
```

```
linea = " "
```

```
Open "C:\Archivos de módulo\matriz.m" For Append As #1
```

```
Print #1, Chr(13) + Chr(10), "if k==", frmSeleccionarForma1.NumeroArchivos, ",",  
Chr(13) + Chr(10), "A=["
```

```
Open "C:\Archivos de módulo\" & frmInicio.CarpetaDisposicion & "\" &  
frmSeleccionarForma1.NumeroArchivos & ".eli" For Output As #2
```

```
Open "C:\Archivos de módulo\" & frmInicio.CarpetaDisposicion & "\" &  
frmSeleccionarForma1.NumeroArchivos & ".elj" For Output As #3
```

```
For i = yminima To ymaxima
```

```
For j = xminima To xmaxima
```

```
linea = linea & oa(i, j) & " "
```

```
If oa(i, j) > 0 Then
```

```
Print #2, i
```

```
Print #3, j
```

```
Else
```

```
End If
```

```
Next j
```

```
Print #1, linea
```

```
If i = ymaxima Then
```

```
Else
```

```
Print #1, ";"
```

```
End If
```

```
linea = " "
```

```
Next i
```

```
Print #1, "];", Chr(13) + Chr(10), "cant=", cantidaddepiezasporforma, ";", Chr(13) +  
Chr(10), "orientacionz=", Znegativa, ";", Chr(13) + Chr(10), "else", Chr(13) + Chr(10),  
"end", Chr(13) + Chr(10)
```

```
Close #1
```

```
Close #2
```

```
Close #3
```

```
Open "C:\Archivos de módulo\" & frmInicio.CarpetaDisposicion & "\" & "relacion.rel"
```

```
For Append As #4
```

```
Print #4, frmSeleccionarForma1.Pieza
```

```
Close #4
```

```
Open "C:\Archivos de módulo\" & frmInicio.CarpetaDisposicion & "\" &  
frmSeleccionarForma1.NumeroArchivos & ".size" For Output As #5
```

```
Print #5, xminima, Chr(13) + Chr(10), xmaxima, Chr(13) + Chr(10), yminima, Chr(13)  
+ Chr(10), ymaxima, Chr(13) + Chr(10)
```

```
Close #5
```

```
frmAgregarformas.lstListadeformas.AddItem frmSeleccionarForma1.Pieza
```

```
frmAgregarformas.lstOrientacion.AddItem orientacion
```

```
frmAgregarformas.lstX.AddItem xmaxima - xminima + 1
```

```
frmAgregarformas.lstY.AddItem ymaxima - yminima + 1
```

```
hide
```

```
frmAgregarformas.Show
```

```
End If
```

```
End Sub
```

```
Private Sub cmdAgregarPuntoInspeccion_Click()
```

```
ReDim Preserve PuntoX(1 To Npunto + 1)
```

```
ReDim Preserve PuntoY(1 To Npunto + 1)
```

```
PuntoX(Npunto + 1) = txtPuntoInspeccionX.Text
```

```
PuntoY(Npunto + 1) = txtPuntoInspeccionY.Text
```

```
Npunto = Npunto + 1
```

```
txtNumeroPuntos.Text = Npunto
```

```
End Sub
```

```
Private Sub cmdDesactivarZNegativa_Click()
```

```
Znegativa = 0
```

```
txtEstadoZNegativa = "Desactivado"
```

```
End Sub
```

```
Private Sub cmdVolver5_Click()
```

```
frmSeleccionarForma1.NumeroArchivos = frmSeleccionarForma1.NumeroArchivos - 1
```

```
hide
```

```
frmSeleccionarForma1.Show
```

```
End Sub
```

```
Private Sub txtOrientacionManual_Change()
```

```
orientacion = txtOrientacionManual.Text
```

```
End Sub
```

```
Private Sub txtPiezas_Change()
```

```
cantidaddepiezasporforma = txtPiezas.Text
```

```
End Sub
```

```
Private Sub txtRadio_Change()
```

```
Radio = txtRadio.Text
```

```
End Sub
```

A.3.11. Formulario frmSeleccionarForma1

Dim Archivo() As String

Public NumeroArchivos As Double

Public Pieza As String

Option Explicit

Private Sub cmdOrientacionAutomatica_Click()

ReDim Preserve Archivo(1 To NumeroArchivos + 1)

Archivo(NumeroArchivos + 1) = txtPathForma.Text

Pieza = txtPathForma.Text

NumeroArchivos = NumeroArchivos + 1

hide

frmOrientacionAutomatica.Show

End Sub

Private Sub cmdOrientacionManual_Click()

ReDim Preserve Archivo(1 To NumeroArchivos + 1)

Archivo(NumeroArchivos + 1) = txtPathForma.Text

Pieza = txtPathForma.Text

NumeroArchivos = NumeroArchivos + 1

hide

frmOrientacionManual.Show

End Sub

Private Sub cmdVolver4_Click()

hide

frmAgregarformas.Show

End Sub

