Anexo 4. Pseudocódigo del programa

```
Algoritmo Cálculo_plástico
    Leer nodos;
    Leer material;
    Leer perfil;
    Leer barras;
    Leer datos;

    nd = Lenght [nodos];
    nb = Lenght [barras];
    mT = T[1] = {{1, -q[1], 0, 0}, {q[1], 1, 0, 0}, {0, 0, 1, 0},
        {0, 0, 0, 1}};
    Para i<-1 Hasta nd-1 Con Paso 1 Hacer
        Li = Norm[nodos[[i + 1, 1]] - nodos[[i, 1]]];
        lx = (nodos[[i + 1, 1, 1]] - nodos[[i, 1, 1]])/Li;
        ly = (nodos[[i + 1, 1, 2]] - nodos[[i, 1, 2]])/Li;
        lz = 0/Li;
        T[i + 1] = {{1, -q[i + 1], 0, lx Li},
            {q[i + 1], 1, 0, ly Li}, {0, 0, 1, lz Li},
            {0, 0, 0, 1}};
        mT = mT.T[i + 1];
    Fin Para
    matrizT = mT;

    NPR = 0;
    Comp = {};
    Para i<-1 Hasta nd Con Paso 1 Hacer
        Si (nodos[[i, 2, 1]] == Fx o nodos[[i, 2, 2]] == Fy) y
            nodos[[i, 2, 3]] == 0 Entonces
            {NPR = NPR, Comp = AppendTo[Comp, 0]}
        SiNo
            {NPR += 1, Comp = AppendTo[Comp, 1]}
        Fin Si
    Fin Para

    GH = 0;
    Para i<-1 Hasta nd Con Paso 1 Hacer
        Si nodos[[i, 2, 1]] == Fx Entonces
            GH += 1;
        Fin Si
        Si nodos[[i, 2, 2]] == Fy Entonces
            GH += 1;
        Fin Si
        Si nodos[[i, 2, 3]] == Mz Entonces
            GH += 1;
        Fin Si
    Fin Para
    GH = GH - 3;
    Si Sum[nodos[[i, 1, 2]], {i, nd}] == 0 Entonces
        GH = GH - 1;
    Fin Si
    EQ = NPR - GH;
    Escribir "Ecuaciones de equilibrio", EQ;
    Escribir "Ecuaciones de compatibilidad", GH;

    mec = {};
    mec = Subsets[
    (
```

```
Para i<-1 Hasta nd Con Paso 1 Hacer
    Si Comp[[i]] == 1 Entonces
        AppendTo[mec, barras[[i, 1, 1]
    Fin Si
Fin Para
mec),
{NPR - (nd - 3), NPR}
];
Escribir "Todas las posibles combinaciones de rótulas
    plásticas son:", mec;
mecnec = mec;

Para avanza<-Lenght[mecnec[[i]]] Hasta Lenght[nodos]
    Con Paso 1 Hacer
    Repetir
        repite += 1;
        mecextra = {};
        Para i<-1 Hasta Lenght[mecnec] Con Paso 1 Hacer
            Si Lenght(mecnec[[i]])==avanza Entonces
                AppendTo[mecextra, mecnec[[i]]]
            Fin Si
        Fin Para
        mec = mecextra;
        dim = Lenght[mecextra];
        mec2 = Table[{}, dim];
        Para i<-1 Hasta dim Con Paso 1 Hacer
            mec2[[i]] = mec[[i]];
            Para j<-1 Hasta nd - Lenght(mec[[i]])
                Con Paso 1 Hacer
                AppendTo[mec2[[i]], 0]
            Fin Para
        Fin Para
        mec3 = Table[0, dim, nd];
        Para i<-1 Hasta dim Con Paso 1 Hacer
            Para j<-1 Hasta nd Con Paso 1 Hacer
                Si mec2[[i, j]] != 0 Entonces
                    mec3[[i, mec2[[i, j]]]] = mec2[[i, j]]
                Fin Si
            Fin Para
        Fin Para
        MCp = Table[{}, dim];
        Para j<-1 Hasta dim Con Paso 1 Hacer
            k = 0;
            Para i<-1 Hasta nd Con Paso 1 Hacer
                Si Comp[[i]] == 1 Entonces
                    Si mec3[[j, i]] != 0 y k == 0 Entonces
                        k = 1;
                        AppendTo[MCp[[j]], q[i] == Δv];
                    Fin Si
                    Si mec3[[j, i]] == 0 Entonces
                        AppendTo[MCp[[j]]
                    SiNo
                        q[i] == 0
                    Fin Si
                Fin Si
            Fin Para
        Fin Para

        Δv = 0.000001;
        sol = Table[0, {dim}];
        gdl = Table[q[i], {i, 1, nd}];
```

```
Repetir
    neq += 1;
    vP = {};
    Para i<-1 Hasta nd - Length[MCp[[neq]]]
        Con Paso 1 Hacer
        Si i == 1 Entonces
            AppendTo[vP, mT[[2, 4]];
        Fin Si
        Si i == 2 Entonces
            AppendTo[vP, mT[[2, 1]];
        Fin Si
        Si i == 3 Entonces
            AppendTo[vP, mT[[1, 4]];
        Fin Si
        j=i;
        Si j > 3 Entonces
            j = 3;
        Fin Si
    Fin Para
    Si j == 1 Entonces
        kk = Solve[
        Join[{vP == nodos[[nd, 1, 2]]}, MCp[[neq]]],
        gdl
        ] /. datos;
    Fin Si
    Si j == 2 Entonces
        kk = Solve[
        Join[{vP == nodos[[nd, 1, {2, 3}]]},
            MCp[[neq]]],
        gdl
        ] /. datos;
    Fin Si
    Si j == 1 Entonces
        kk = Solve[
        Join[{vP == nodos[[nd, 1, {2, 3, 1}]]},
            MCp[[neq]]],
        gdl
        ] /. datos;
    Fin Si
    s = 0;
    j = 0;
    min = 10^10;
    i = 1;
    Repetir
        i += 1;
        s += Norm[q[i]] /. kk
    Hasta Que i = nd - 1
    Repetir
        Si s[[i]] < min Entonces
            min = s[[i]];
            j = i;
        Fin Si
    Hasta Que Lenght[s]
    sol[[neq]] = kk[[j]],
Hasta Que neq = dim
sol2 = {};
Para i<-1 Hasta dim Con Paso 1 Hacer
    Si sol[[i]] == List Entonces
    SiNo
    Otro caso
        AppendTo[sol2, sol[[i]]]
```

```
        Fin Si
    Fin Para

    dim = Lenght[sol]
    d = Table[
        Table[
            ((Dot @@ Table[T[j], {j, i}])[[{1, 2, 3}, 4]])
                - nodos[[i, 1]],
            {i, nd}
        ] /. sol[[neq]] /. datos,
        {neq, dim}
    ]

    eq = Table[
        (Sum[d[[neq, i]].nodos[[i, 2]], {i, nd}] -
Sum[M[i] q[i], {i, nd}])/Δv /. sol[[neq]],
        {neq, dim}
    ] /. (0.0 -> 0) // Simplify;

    Mreal = Table[{}, {dim}];
    Table[
        Para i<-1 Hasta nd Con Paso 1 Hacer
            Si q[i] != 0 /. sol[[neq]] o Comp[[i]] == 0
                Entonces
                Mreal[[neq]] = AppendTo[Mreal[[neq]],
                Si repite == 1 Entonces
                    M[i] -> Sign[q[i]] Mp /. sol[[neq];
                SiNo
                    M[i] -> -Sign[q[i]] Mp /. sol[[neq]];
                Fin Si
            SiNo
                Mreal[[neq]] = AppendTo[Mreal[[neq]],
                    θ[i] -> 0]
            Otro caso
                Mreal[[neq]] = AppendTo[Mreal[[neq]],
                    M[i] -> 0]
            Fin Si
        Fin Para
        {neq, dim}
    ];

    cc = {};
    Para i<-1 Hasta nd Con Paso 1 Hacer
        Si Comp[[i]] == 0 Entonces
            AppendTo[cc, m[i] == 0]
        Fin Si
    Fin Para
    ec = {};
    Para i<-1 Hasta EQ Con Paso 1 Hacer
        Para j<-1 Hasta dim Con Paso 1 Hacer
            Si sol[[j, nd - i + 1, 2]] != 0 Entonces
                AppendTo[ec, eq[[j]] /. F -> 0 /.
        Table[M[i] -> m[i], {i, nd}]];
                j = dim;
                eq[[j]] = 0;
            Fin Si
        Fin Para
    Fin Para

    mvirtual = Table[
        Solve[
```

```
        Join[
            Table[
                m[i] == DiscreteDelta[j - i],
            {i, GH}], cc, {ec == Table[0, {EQ}]}
        ]
    ][[1]],
    {j, GH}
];

EC = Table[
    Table[
        pfv=0;
        Para i<-1 Hasta nb Con Paso 1 Hacer
            Li = Norm[nodos[[i + 1, 1]] -
                nodos[[i, 1]]] /. datos;
            vEi = material[[barras[[i, 2, 1]], 1]] ;
            Izi = perfil[[barras[[i, 2, 2]], 2]];
            Mi = M[i];
            Mj = M[i + 1];
            mi = m[i];
            mj = m[i + 1];
            pfv += Li/(6 vEi Izi) (mi (2 Mi + Mj) +
                mj (Mi + 2 Mj));
        Fin Para
        pfv += Sum[m[i] θ[i], {i, nd}];
        pfv = Simplify[pfv /. Mreal[[neq]] /.
            mvirtual[[j]]],
        {j, GH}
    ],
    {neq, dim}
];

adic = Table[
    Table[
        Mreal[[neq, i, 1]] == Mreal[[neq, i, 2]],
        {i, nd}
    ],
    {neq, dim}
];
incog = Join[{F}, Table[M[i], {i, nd}], Table[θ[i],
    {i, nd}]];
giros = Table[{}, {dim}];
Table[
    Para i<-1 Hasta nd Con Paso 1 Hacer
        Si Mreal[[neq, i, 1]] == M[i] Entonces
            AppendTo[giros[[neq]], θ[i]];
        Fin Si
    Fin Para
    {neq, dim}
];

Repetir
    neq += 1;
    Para i<-1 Hasta Lenght[giros[[neq]]]
        Con Paso 1 Hacer
        Para j<-1 Hasta Lenght[eq] - EQ + 1
            Con Paso 1 Hacer
            add = {};
            Para k<-j Hasta EQ + j - 1
                Con Paso 1 Hacer
                AppendTo[add, eq[[k]]]
```

```
                        Fin Para
                        sol = Solve[
                            Join[
                                {
                                EC[[neq]] == Table[0, {GH}],
                                add == Table[0, {EQ}],
                                {giros[[neq, i]]} == {0}
                                },
                                adic[[neq]]
                            ], incog
                        ][[1]] /. (0.0 -> 0);
                        Si Refine[
                            (And @@ Table[Norm[M[i]] <= Mp,
                                {i, nd}] /. sol),
                            Assumptions -> {Mp > 0, Iz > 0,
                                vE > 0}
                            ] Entonces
                            Si Refine[
                                (And @@ Table[
                              Sign[θ[i] M[i]] >= 0, {i, nd}
                        ] /. sol),
                                Assumptions -> {Mp > 0, Iz > 0,
                                    vE > 0}
                                ] Entonces
                                solucion = sol;
                                i = Length[giros[[neq]]];
                            Fin si
                        Fin Si
                    Fin Para
                Fin Para
            Hasta Que neq = dim
            Si (solucion[[1, 2]] //. datos) >= 0 Entonces
                solucion = (solucion //. datos), repite = 2;
            Fin Si
        Hasta Que repite = 2
        Si (solucion[[1, 2]] //. datos) >= 0 Entonces
            avanza = Length[nodos] + 1;
        Fin Si
    Fin Para
Fin Algoritmo
```