```vb
Imports System.Collections.Generic

Imports Grasshopper.Kernel
Imports Rhino.Geometry
Imports Interop.gsa_8_7
Imports UsageDataCollection


Public Class GetNodeCoord
    Inherits GH_Component
    ''' <summary>
    ''' Initializes a new instance of the GetNodeCoord class.
    ''' </summary>
    Public Sub New()
        MyBase.New("Get Node Coordinates", "NodeCoord", _
                "Import Node Coordinates from GSA File", _
                "GsaComTools", "Nodes")
    End Sub

    ''' <summary>
    ''' Registers all the input parameters for this component.
    ''' </summary>
    Protected Overrides Sub RegisterInputParams(pManager As GH_Component.GH_InputParamManager)
        pManager.AddGenericParameter("Gsa", "Gsa", "Gsa COM Object", GH_ParamAccess.item)
        pManager.AddIntegerParameter("Nodes", "Nodes", "Node Numbers", GH_ParamAccess.list)
    End Sub

    ''' <summary>
    ''' Registers all the output parameters for this component.
    ''' </summary>
    Protected Overrides Sub RegisterOutputParams(pManager As GH_Component.GH_OutputParamManager)
        pManager.AddGenericParameter("Gsa", "Gsa", "Gsa COM Object", GH_ParamAccess.item)
        pManager.AddNumberParameter("X", "X", "X Coordinates", GH_ParamAccess.list)
        pManager.AddNumberParameter("Y", "Y", "Y Coordinates", GH_ParamAccess.list)
        pManager.AddNumberParameter("Z", "Z", "Z Coordinates", GH_ParamAccess.list)
    End Sub

    ''' <summary>
    ''' This is the method that actually does the work.
    ''' </summary>
    ''' <param name="DA">The DA object is used to retrieve from inputs and store in outputs.</param>
    Protected Overrides Sub SolveInstance(DA As IGH_DataAccess)
        'Declare Variables
        Dim FileName As String = ""
        Dim NodeRefs As New List(Of Integer)
        Dim Nodes() As GsaNode
        Dim NodeXCoords As New List(Of Double)
        Dim NodeYCoords As New List(Of Double)
        Dim NodeZCoords As New List(Of Double)
        Dim s As Short
        Dim myDataCollector As New DataCollector
        Dim myObjectWrapper As Types.GH_ObjectWrapper

        'Import GSA File Name
        If (Not DA.GetData(0, myObjectWrapper)) Then Return
        If (Not DA.GetDataList(1, NodeRefs)) Then Return

        Dim GsaObj As ComAuto = TryCast(myObjectWrapper.Value, ComAuto)

        'Get Data
        s = GsaObj.Nodes(NodeRefs.ToArray, Nodes)
        For Each item In Nodes
            NodeXCoords.Add(item.Coor(0))
            NodeYCoords.Add(item.Coor(1))
            NodeZCoords.Add(item.Coor(2))
        Next

        ' Assign the data to the output parameter.
        DA.SetData(0, GsaObj)
        DA.SetDataList(1, NodeXCoords)
        DA.SetDataList(2, NodeYCoords)
        DA.SetDataList(3, NodeZCoords)

        'Send Usage Data
        myDataCollector.UsageStatistics("GetNodeCoord")

    End Sub

    ''' <summary>
    ''' Provides an Icon for every component that will be visible in the User Interface.
    ''' Icons need to be 24x24 pixels.
    ''' </summary>
    Protected Overrides ReadOnly Property Icon() As System.Drawing.Bitmap
        Get
            'You can add image files to your project resources and access them like this:
            ' return Resources.IconForThisComponent;
            Return My.Resources.Node_Coords
        End Get
    End Property

    ''' <summary>
    ''' Gets the unique ID for this component. Do not change this ID after release.
    ''' </summary>
    Public Overrides ReadOnly Property ComponentGuid() As Guid
        Get
            Return New Guid("{5a890e3c-8398-4441-9b7a-49976a2c0161}")
        End Get
    End Property
End Class
```