```
fun algoritmoBCargaPenalizacion (idImagen : ent, modo : tmodo, copias_objetivo : ent,
      conector : conectorBD) dev
      <idsServidores : lista de ent, copias : ent, codigoError : ent>

      casos
            ◊ modo = ARRANQUE NORMAL:
                  servidores := conectorBD.obtenerServidores(idImagen)
            ◊ modo = CREAR O EDITAR IMAGEN:
                  servidores := conectorBD.obtenerServidoresEdicion(idImagen)
            ◊ modo = DESPLEGAR IMAGEN:
                  servidores := conectorBD.obtenerServidoresCandidatos(idImagen)
      fcasos

      caracImagen := conectorBD.obtenerCaractFamilia(idImagen)
      penalizaciones := lista_vacía()

      para cada idServidor en servidores hacer
            estado_servidor := conectorBD.obtenerEstado(idServidor)
            penalizacion_cpus := calcularPenalizacionCPUs()
            penalizacion_ram := calcularPenalizacionRAM()
            penalizacion_alm := calcularPenalizacionAlmacenamiento()
            penalizacion_alm_temporal :=
                  calcularPenalizacionAlmacenamientoTemporal()

            si (penalizacion_cpus ≤ 1 + exceso_cpus y penalizacion_ram ≤ 1
                  y penalizacion_alm ≤ 1 y penalizacion_alm_temporal ≤ 1) entonces
                  penalizacion_global := calcularPenalizacionGlobal()
                  casos
                        ◊ modo = DESPLEGAR IMAGEN:
                              copias_con_cpus := estadoServidor.numCPUs /
                                    caracImagen.cpus
                              copias_con_ram := estadoServidor.tamannoRAM /
                                    caracImagen.ram
                              copias_con_alm_temp :=
                                    estadoServidor.espacio_temp_disponible /
                                          caracImagen.espacio_temp
                              copias := mín(copias_con_cpus, copias_con_ram,
                                    copias_con_alm_temp
                              penalizaciones.annadir(crear_tupla(idServidor,
                                    penalizacion_global, copias))
                         ◊ modo != DESPLEGAR IMAGEN:
                              penalizaciones.annadir(crear_tupla(idServidor,
                                    penalizacion_global))
                  fasos
            fsi
      fpara

      si es_lista_vacía(penalizaciones) entonces
            devolver (lista_vacía, 0, codigo_error)
      si no
            ordenar_por_penalizacion_global(penalizaciones)
            resultado = lista_vacía ; copias := 0
            casos
                  ◊ modo = DESPLEGAR IMAGEN:
                        mientras copias < copias_objetivo :
                              añadir_a_lista(resultado,
                                    id_cabeza(penalizaciones)
                              copias += copias_cabeza(penalizaciones)
                              quitar_cabeza(penalizaciones)
                  ◊ modo != DESPLEGAR IMAGEN:
                        resultado = lista_unitaria(id_cabeza(penalizaciones))
            fcasos
      fsi
      devolver (resultado, copias, no_error)
ffun
```