U∩  
1 – LIBRERIAS\_ASOCIADAS está en BCNF? No, porque existe df1 donde {idLibreria} no es superclave de LIBRERIAS\_ASOCIADAS.

Entonces particionamos LIBRERIAS\_ASOCIADAS y nos quedaría:

**R1(idLibreria, nombreLibreria)**

**R2 ( idLibreria, idArticulo, nombreArticulo, idComponente, nombreComponente, idFabricanteArticulo, idDueño)**

-Se perdió información? No, porque de R1∩R2 = {idLibreria} y esta es clave de R1

-Se perdieron dependencias funcionales? No, porque vale la df1 en r1 y valen df2 y df3 en R2

-R1 está en BCNF? Sí, porque vale df1 donde {idLibreria} es superclave de R1

-R2 está en BCNF? No, porque vale df2 donde {idLibreria, idArticulo} no es superclave de R2

Particionamos R2 y nos queda:

**R3(idLibreria, idArticulo, nombreArticulo)**

**R4 ( idLibreria, idArticulo, idComponente, nombreComponente, idFabricanteArticulo, idDueño)**

-Se perdió información? No porque R3∩R4 = {idLibreria, idArticulo} y esta es clave en R3.

-Se perdieron dependencias funcionales? No, porque vale df2 en R3 y df3 en R4

-R3 está en BCNF? Sí, porque vale df2 donde {idLibreria, idArticulo} es superclave de R3

-R4 está en BCNF? No, porque vale df3 donde {idLibreria, idArticulo, idComponente} no es superclave de R4

Particionamos R4 y nos queda:

**R5(idLibreria, idArticulo, idComponente, nombreComponente)**

**R6( idLibreria, idArticulo, idComponente, idFabricanteArticulo, idDueño)**

-Se perdió información? No, porque R5∩R6 = {idLibreria, idArticulo, idComponente} y esta es superclave en R5.

-Se perdieron dependencias funcionales? No, porque vale df3 en R5

-R5 está en BCNF? Sí, porque vale df3 donde {idLibreria, idArticulo, idComponente} es superclave de R5.

-R6 está en BCNF? Sí, porque las únicas dependencias que hay son triviales.

Particiones que quedaron en BCNF:

**R1(idLibreria, nombreLibreria)**

**R3(idLibreria, idArticulo, nombreArticulo)**

**R5(idLibreria, idArticulo, idComponente, nombreComponente)**

**R6( idLibreria, idArticulo, idComponente, idFabricanteArticulo, idDueño)**

Clave: (idLibreria, idArticulo, idComponente, idFabricanteArtículo, idDueño)

-Existen dependencias multivaluadas en R6?

Dm1 : idLibreria ->> idDueño

Dm2 : idLibreria, idArticulo ->> idComponente

Dm3 : idLibreria, idArticulo ->> idFabricanteArticulo

R6 cumple con 4FN? No, porque vale la dm1 que no es trivial en R6.

Particionamos R6 y nos queda:

**R7( idLibreria, idDueño)**

**R8( idLibreria, idArticulo, idComponente, idFabricanteArticulo)**

-R7 cumple con 4FN? Sí, porque vale la dm1 y esta es trivial.

-R8 cumple con 4FN? No, porque vale la dm2 que no es trivial en R8.

Particionamos R8 y nos queda:

**R9( idLibreria, idArticulo, idComponente)**

**R10( idLibreria, idArticulo, idFabricanteArticulo)**

-R9 cumple con 4FN? Sí, porque vale la dm2 y esta es trivial.

-R10 cumple con 4FN? Sí, porque vale la dm3 y esta es trivial.

Finalmente nos queda:

**R1(idLibreria, nombreLibreria)**

**R3(idLibreria, idArticulo, nombreArticulo)**

**R5(idLibreria, idArticulo, idComponente, nombreComponente)**

**R7( idLibreria, idDueño)**

**R9( idLibreria, idArticulo, idComponente)**

**R10( idLibreria, idArticulo, idFabricanteArticulo)**

2 – EMPLEADO está en BCNF? No, porque existe dfs tal que el determinante de la df1 no es superclave del esquema.

Entonces particionamos EMPLEADO  
y nos quedaría:

**R1(idOficina, nombreOficina)**

**R2 ( idEmpleado,  
nombreEmpleado, idOficina, idResponsableOficina,  
cargaHorariaEnOficina, nombreResponsableOficina, añoingresoOficina, idActividadEmpleadoOficina,  
nombreActividadOficina, dniEmpleado)**

-Se perdió información? No,  
porque de R1∩R2 = es clave de R1

-Se perdieron dependencias  
funcionales? No, porque vale la df1 en r1 y valen df2 y df3, df4, df5 en R2

-R1 está en BCNF? Sí, porque  
vale df1 donde {idOficina} es superclave de R1

-R2 está en BCNF? No, porque  
existen dfs validas donde por ej el determinante de la df2 no es superclave del  
esquema, entonces.. Particiono usando df2

Particionamos R2 y nos queda:

**R3(idResponsableOficina, idOficina, nombreResponsableOficina)**

**R4 ( idEmpleado,  
nombreEmpleado, idOficina, idResponsableOficina,  
cargaHorariaEnOficina, añoingresoOficina, idActividadEmpleadoOficina,  
nombreActividadOficina, dniEmpleado)**

-Se perdió información? No  
porque R3∩R4 = clave en R3.

-Se perdieron dependencias  
funcionales? No, porque vale df2 en R3 y df3, df4, df5 en R4

-R3 está en BCNF? Sí, porque  
vale df2 donde {idResponsableOficina, idOficina} es superclave de R3

-R4 está en BCNF? No, porque  
existe df3, df4, df5 validas en R4 y su determinante no es superclave en el  
esquema, voy a particionar usando df3

**R5(idEmpleado, nombreEmpleado, idOficina,  
añoingresoOficina, dniEmpleado, cargaHorariaEnOficina)**

**R6 (idEmpleado,  
idResponsableOficina, idActividadEmpleadoOficina, nombreActividadOficina)**

-Se perdió información? No,  
porque R5∩R6 = clave en R5.

-Se perdieron dependencias  
funcionales? No, porque vale df3 y df4 en R5

Y df5 es valida en R6

¿Cuáles son las claves candidatas de R5? Cc1 {dniEmpleado} cc2{idEmpleado}

-R5 está en BCNF? Sí,  
porque vale df3,y df4 y sus determinantes  
son superclave en R5.

-R6 está en BCNF? No, porque existe df5 valida  
en R6 y su determinante no es superclave en el esquema, voy a particionar  
usando df5

Nueva tabla R7:

**R7(idActividadEmpleadoOficina, nombreActividadOficina )**

**R8 (idEmpleado,  
idResponsableOficina, idActividadEmpleadoOficina,)**

-Se perdió información? No,  
porque R7∩R8 = clave en R7.

-Se perdieron dependencias  
funcionales? No, porque vale df5 y en R7.

-R7 está en BCNF? Sí, porque vale df5 y su determinantes es superclave en R7.

-R8 está en BCNF? Si, porque todas las dfs  
validas son triviales.

Particiones que quedaron en BCNF: R1, R3, R5, R7, R8

Clave: **(idEmpleado, idResponsableOficina, idActividadEmpleadoOficina)**

-Existen dependencias  
multivaluadas en R8?

Dm1: idEmpleado ->> idActividadEmpleadoOficina

R8 cumple con 4FN? No,  
porque vale la dm1 que no es trivial en R8.

Particionamos utilizando DM1:

**R9 (idEmpleado,  
idResponsableOficina, idActividadEmpleadoOficina)**

**R10(idEmpleado,  
idResponsableOficina)**

-R9 cumple con 4FN? Sí, porque vale la dm1 y esta es trivial.

DM2) ø->> idEmpleado

DM3) ø->> idResponsableOficina

**R11 (idEmpleado) Está en 4FN porque vale la dm2 y es trivial.**

**R12(idResponsableOficina)  
Está en 4FN porque vale la dm3 y es trivial.**

R9, R11, R12 -> No  
intermedias y ultimas -> Multivaluadas.

R11 es proyección de R5

R12 es proyección de R3

Tablas en 4FN-> R1, R3, R5, R7, R9

**3 - INFORME\_MEDICO (idMedico, apynMedico, tipoDocM, nroDocM, fechaNacM, matricula, direcciónM, teléfonoM,**

**idPaciente, apynPaciente, tipoDocP, nroDocP, fechaNacP, idObraSoc, nroAfiliado, direcciónP, teléfonoP,**

**nombreOS, direcciónOS, teléfonoOS, idÓrgano, descripción, idEstudio, resultado, fechaEstudio, informe)**

Cc: {idEstudio**}**

Dependencias funcionales:

Df1: idMedico -> apynMedico, tipoDocM, nroDocM, fechaNacM, matricula, direcciónM, teléfonoM

Df2: matricula -> idMedico, apynMedico, tipoDocM, nroDocM, fechaNacM, matricula, direcciónM, teléfonoM

Df3: tipoDocM, nroDocM -> matricula, idMedico, apynMedico, fechaNacM, matricula, direcciónM, teléfonoM

Df4: idPaciente -> apynPaciente, tipoDocP, nroDocP, fechaNacP, idObraSoc, nroAfiliado, direcciónP, teléfonoP

Df5: tipoDocP, nroDocP -> idPaciente, apynPaciente, fechaNacP, idObraSoc, nroAfiliado, direcciónP, teléfonoP

Df6: idObraSoc, nroAfiliado -> idPaciente, apynPaciente, tipoDocP, nroDocP, fechaNacP, direcciónP, teléfonoP

Df7: idObraSoc -> nombreOS, direcciónOS, teléfonos

Df8: idOrgano -> descripcion

Df9: idEstudio -> idPaciente, idMedico, idOrgano, resultado, fechaEstudio, informe

Df10: idEstudio -> idPaciente, matricula, idOrgano, resultado, fechaEstudio, informe

Df11: idEstudio -> idPaciente, tipoDocM, nroDocM, idOrgano, resultado, fechaEstudio, informe

Df12: idEstudio -> tipoDocP, nroDocP, tipoDocM, nroDocM, idOrgano, resultado, fechaEstudio, informe

Df13: idEstudio -> idObraSoc, nroAfiliado, tipoDocM, nroDocM, idOrgano, resultado, fechaEstudio, informe

Df14: idEstudio -> idObraSoc, nroAfiliado, matricula, idOrgano, resultado, fechaEstudio, informe

Df15: idEstudio -> idObraSoc, nroAfiliado, idMedico, idOrgano, resultado, fechaEstudio, informe

Df16: idEstudio -> tipoDocP, nroDocP, idMedico, idOrgano, resultado, fechaEstudio, informe

Df17: idEstudio -> tipoDocP, nroDocP, matricula, idOrgano, resultado, fechaEstudio, informe

-INFORME\_MEDICO cumple BCNF? No, porque vale df1 donde {idMedico} no es superclave de INFORME\_MEDICO.

Particionamos y queda:

**R1(idMedico, apynMedico, tipoDocM, nroDocM, fechaNacM, matricula, direcciónM, teléfonoM)**

**R2(idMedico, idPaciente, apynPaciente, tipoDocP, nroDocP, fechaNacP, idObraSoc, nroAfiliado, direcciónP, teléfonoP,**

**nombreOS, direcciónOS, teléfonoOS, idÓrgano, descripción, idEstudio, resultado, fechaEstudio, informe)**

-Se perdió información? No, porque de R1∩R2 = {idMedico} y esta es clave de R1

-Se perdieron dependencias funcionales? No, porque vale la df1, df2 y df3 en r1 y valen df4 a df17 en R2

-R1 está en BCNF? Sí, porque vale df1, df2 y df3 cuyos determinantes son clave del esquema, un caso especial de superclave.

-R2 está en BCNF? No, porque valen de df4 a df17, donde por ej idObraSoc no es superclave. Se particiona usando la df7

**R3(idObraSoc, nombreOS, direcciónOS, teléfonoOS)**

**R4(idMedico, idPaciente, apynPaciente, tipoDocP, nroDocP, fechaNacP, idObraSoc, nroAfiliado, direcciónP, teléfonoP, idÓrgano, descripción, idEstudio, resultado, fechaEstudio, informe)**

-Se perdio información? No, porque de R3∩R4 = {idObraSoc} y esta es clave en R3.

-Se perdieron dependencias funcionales? No, porque vale df7 en R3 y df4, df5, df6, df8 a df17 en R4.

-R3 cumple BCNF? Si, porque vale df7, donde {idObraSoc} es superclave de R3.

-R4 cumple BCNF? No, porque valen

**R5(idÓrgano, descripción)**

**R6(idMedico, idPaciente, apynPaciente, tipoDocP, nroDocP, fechaNacP, idObraSoc, nroAfiliado, direcciónP, teléfonoP, idÓrgano, idEstudio, resultado, fechaEstudio, informe)**

-Se perdio información? No, porque R5∩R6 = {idOrgano} y esta es superclave de R5.

-Se perdieron dependencias funcionales? No, porque vale df8 en R5 y valen df4, df5, df6, df9 a df17 en R6

-R5 cumple BCNF? Si, porque vale df8, donde {idOrgano} es superclave de R5.

-R6 cumple BCNF? No, porque valen df4, df5, df6, df9 a df17, donde por ej idPaciente no es superclave. Particionamos usando df4

**R7(idPaciente, apynPaciente, tipoDocP, nroDocP, fechaNacP, idObraSoc, nroAfiliado, direcciónP, teléfonoP)**

**R8(idEstudio, idPaciente, idMedico,idÓrgan, resultado, fechaEstudio, informe)**

-Se perdio información? No, porque R7∩R8 = {idPaciente} y esta es superclave de R7.

-Se perdieron dependencias funcionales? No, porque valen df4, df5 y df6 en R7 y df9 a df17 en R8.

-R7 cumple BCNF? Si, porque valen df4, df5 y df6 cuyos determinantes son clave del esquema, caso especial de superclave.

-R8 cumple BCNF? Si, porque vale df9, donde {id\_estudio} es superclave. Df10 a df17 estan contempladas en el esquema también, ya que el determinante es clave del esquema.

-Hay dependencias multivaluadas? No, no hay atributos primos.

Cumple con 4FN? Si, porque no existen dependencias multivaluadas.

Esquemas en BCNF:

**R1(idMedico, apynMedico, tipoDocM, nroDocM, fechaNacM, matricula, direcciónM, teléfonoM)  
R3(idObraSoc, nombreOS, direcciónOS, teléfonoOS)  
R5(idÓrgano, descripción)  
R7(idPaciente, apynPaciente, tipoDocP, nroDocP, fechaNacP, idObraSoc, nroAfiliado, direcciónP, teléfonoP)**

**R8(idEstudio, idPaciente, idMedico,idÓrgan, resultado, fechaEstudio, informe)**

4 - AEROPUERTO (#aeropuerto, #pista, fecha, #avion)

Df1: #avion, #aeropuerto,fecha -> #pista  
-AEROPUERTO cumple BCNF? Si, porque vale la df1 donde #aeropuero, fecha, #avion son superclave de AEROPUERTO

Hay dependencias multivaludas? No.

Cumple con 4FN? Si, porque no existen dependencias multivaluadas.

Clave: {#aeropuero, fecha, #avion}

5 - **DISPOSITIVOS (Marca\_id, descripMarca, modelo\_id, descripModelo, equipo\_tipo\_id, descripEquipoTipo,empresa\_id, nombreEmpresa, cuit, direcciónEmpresa, usuario\_id, apyn, nro\_doc, direcciónUsuario, cuil,plan\_id, descripPlan, importe, equipo\_id, imei, fec\_alta, fec\_baja, observaciones, línea\_id, nroContrato,fec\_alta\_linea, fec\_baja\_linea)**

Dependencias funcionales:  
 df1: Marca\_id -> descripMarca  
 df2: modelo\_id -> descripModelo, Marca\_id  
 df3: equipo\_id -> equipo\_tipo\_id, modelo\_id, imei, fec\_alta, fec\_baja, observaciones  
 df4: imei -> equipo\_tipo\_id, modelo\_id, equipo\_id, fec\_alta, fec\_baja, observacioes  
 df5: equipo\_tipo\_id -> descripEquipoTipo  
 df6: plan\_id -> empresa\_id, descripPlan, importe  
 df7: empresa\_id -> nombreEmpresa, cuit, direccionEmpresa  
 df8: cuit -> nombreEmpresa, empresa\_id, direccionEmpresa  
 df9: usuario\_id -> apyn, nro\_doc, direccionUsuario, cuil  
 df10: nro\_doc -> usuario\_id, apyn, direccionUsuario, cuil  
 df11: cuil -> nro\_doc, usuario\_id, apyn, direccionUsuario  
 df12: linea\_id -> nroContrato, plan\_id, fec\_alta\_linea, fec\_baja\_linea, equipo\_id, usuario\_id  
 df13: linea\_id -> nroContrato, plan\_id, fec\_alta\_linea, fec\_baja\_linea, imei, usuario\_id  
 df14: linea\_id -> nroContrato, plan\_id, fec\_alta\_linea, fec\_baja\_linea, equipo\_id, nro\_doc  
 df15: linea\_id -> nroContrato, plan\_id, fec\_alta\_linea, fec\_baja\_linea, equipo\_id, cuil  
 df16: linea\_id -> nroContrato, plan\_id, fec\_alta\_linea, fec\_baja\_linea, imei, nro\_doc  
 df17: linea\_id -> nroContrato, plan\_id, fec\_alta\_linea, fec\_baja\_linea, imei, cuil

-DISPOSITIVOS cumple BCNF? No, porque valen df1 a df11, donde por ej Marca\_id no es superclave. Particionamos usando df1

**R1(Marca\_id, descripMarca)**

**R2(Marca\_id, modelo\_id, descripModelo, equipo\_tipo\_id, descripEquipoTipo,empresa\_id, nombreEmpresa, cuit, direcciónEmpresa, usuario\_id, apyn, nro\_doc, direcciónUsuario, cuil,plan\_id, descripPlan, importe, equipo\_id, imei, fec\_alta, fec\_baja, observaciones, línea\_id, nroContrato,fec\_alta\_linea, fec\_baja\_linea)**

-Se perdio información? No, porque R1∩R2 = {Marca\_id}, la cual es superclave de R1.  
-Se perdieron dependencias funcionales? No, porque vale df1 en R1 y valen df2 a df17 en R2.  
-R1 cumple BCNF? Si, porque vale df1 donde Marca\_id es superclave del esquema.  
-R2 cumple BCNF? No, porque valen las df2 a df17, donde por ej modelo\_id no es superclave. Particionamos usando df2

**R3(modelo\_id, descripModelo, Marca\_id)**

**R4(modelo\_id, equipo\_tipo\_id, descripEquipoTipo,empresa\_id, nombreEmpresa, cuit, direcciónEmpresa, usuario\_id, apyn, nro\_doc, direcciónUsuario, cuil,plan\_id, descripPlan, importe, equipo\_id, imei, fec\_alta, fec\_baja, observaciones, línea\_id, nroContrato,fec\_alta\_linea, fec\_baja\_linea)**

-Se perdio información? No, porque R3∩R4 = {modelo\_id}, la cual es superclave de R3.  
-Se perdieron dependencias funcionales? No, porque vale df2 en R3 y df3 a df17 en R4.  
-R3 cumple BCNF? Si, porque vale df2 donde {modelo\_id} es superclave del esquema.  
-R4 cumple BCNF? No, porque valen df3 a df17, donde por ej equipo\_tipo\_id no es superclave. Particionamos usando df5.

**R5(equipo\_tipo\_id, descripEquipoTipo)**

**R6(modelo\_id, equipo\_tipo\_id,empresa\_id, nombreEmpresa, cuit, direcciónEmpresa, usuario\_id, apyn, nro\_doc, direcciónUsuario, cuil,plan\_id, descripPlan, importe, equipo\_id, imei, fec\_alta, fec\_baja, observaciones, línea\_id, nroContrato,fec\_alta\_linea, fec\_baja\_linea)**

-Se perdio información? No, porque R5∩R6 = {equipo\_tipo\_id}, la cual es superclave de R5.  
-Se perdieron dependencias funcionales? No, porque vale df5 en R5 y valen df3, df4, df6 a df17 en R6.  
-R5 cumple BCNF? Si, porque vale df5, donde {equipo\_tipo\_id} es superclave.  
-R6 cumple BCNF? No, porque valen df3, df4, df6 a df17, donde por ej equipo\_id no es superclave. Particionamos usando df3

**R7(equipo\_id, modelo\_id, equipo\_tipo\_id, imei, fec\_alta, fec\_baja, observaciones)**

**R8(empresa\_id, nombreEmpresa, cuit, direcciónEmpresa, usuario\_id, apyn, nro\_doc, direcciónUsuario, cuil,plan\_id, descripPlan, importe, equipo\_id, línea\_id, nroContrato, fec\_alta\_linea, fec\_baja\_linea)**

-Se perdio información? No, porque R7∩R8 = {equipo\_id}, la cual es superclave de R7.  
-Se perdieron dependencias funcionales? No, porque valen df3 y df4 en R7 y valen df6 a df17 en R8  
-R7 cumple BCNF? Si, porque valen df3 y df4, cuyos determinantes son clave de R7, caso especial de superclave.  
-R8 cumple BCNF? No, porque valen df6 a df17, donde por ej empresa\_id no es superclave. Particionamos usando df7.

**R9(empresa\_id, nombreEmpresa, cuit, direcciónEmpresa)**

**R10(empresa\_id, usuario\_id, apyn, nro\_doc, direcciónUsuario, cuil,plan\_id, descripPlan, importe, equipo\_id, línea\_id, nroContrato, fec\_alta\_linea, fec\_baja\_linea)**

-Se perdio información? No, porque R9∩R10 = {empresa\_id}, la cual es superclave de R9.  
-Se perdieron dependencias funcionales? No, porque valen df7 y df8 en R9 y df6, df9 a df17 en R10.  
-R9 cumple BCNF? Si, porque vale df7 y df8, cuyos determinantes son clave, un caso especial de superclave.  
R10 cumple BCNF? No, porque valen df6, df9 a df17, donde por ej plan\_id no es superclave. Particionamos usando df6

**R11(plan\_id, empresa\_id, descripPlan, importe)**

**R12(usuario\_id, apyn, nro\_doc, direcciónUsuario, cuil,plan\_id, equipo\_id, línea\_id, nroContrato, fec\_alta\_linea, fec\_baja\_linea)**

-Se perdio información? No, porque R11∩R12 = {plan\_id}, la cual es superclave de R11.  
-Se perdieron dependencias funcionales? No, porque vale df6 en R11 y valen df9 a df17 en R12.  
-R11 cumple BCNF? Si, porque vale df6, donde plan\_id es superclave.  
-R12 cumple BCNF? No, porque vale df9 a df17, donde por ej usuario\_id no es superclave. Particionamos usando df9.

**R13(usuario\_id, apyn, nro\_doc, direcciónUsuario, cuil)**

**R14(línea\_id, usuario\_id,plan\_id, equipo\_id, nroContrato, fec\_alta\_linea, fec\_baja\_linea)**

-Se perdio información? No, porque R13∩R14 = {usuario\_id}, la cual es superclave de R13.  
-Se perdieron dependencias funcionales? No, porque vale df9, df10 y df11 en R13 y valen df12 a df17 en R14.  
-R13 cumple BCNF? Si, porque vale df9, df10 y df11, cuyos determinantes son clave de R13, caso especia de superclave.  
-R14 cumple BCNF? Si, porque vale df12 a df17, cuyos determinantes son clave del esquema.

-Hay dependencias multivaluadas? No, porque hay atributos primos.  
-R14 esta en 4FN? Si, porque no hay dependencias multivaludas.

Esquemas que estan en BCNF:  
**R1(Marca\_id, descripMarca)  
R3(modelo\_id, descripModelo, Marca\_id)  
R5(equipo\_tipo\_id, descripEquipoTipo)  
R7(equipo\_id, modelo\_id, equipo\_tipo\_id, imei, fec\_alta, fec\_baja, observaciones)  
R9(empresa\_id, nombreEmpresa, cuit, direcciónEmpresa)  
R11(plan\_id, empresa\_id, descripPlan, importe)  
R13(usuario\_id, apyn, nro\_doc, direcciónUsuario, cuil)  
R14(línea\_id, usuario\_id,plan\_id, equipo\_id, nroContrato, fec\_alta\_linea, fec\_baja\_linea)**

6 - TOMAS\_FOTOGRAFICAS ( idElemento, descripcionElemento, idFoto, fechaFoto, obturacionCamaraFoto,idCamara, caracteristicaTecnicaCamara, descripcionCaracteristica)

Dependencias funcionales:   
 df1: idFoto -> idCamara, fechaFoto, obturacionCamaraFoto  
 df2: idElemento -> descripcionElemento  
 df3: caracteristicaTecnicaCamara -> idCamara, descripcionCaracteristica  
   
Cc:{idFoto, idElemento, caracteristicaTecnicaCamara}

TOMAS\_FOTOGRAFICAS cumple BCNF? No, porque existe df1 a df3, donde por ej idElemento no es superclave. Particionamos usando df2

**R1(idElemento, descripcionElemento)**

**R2 ( idElemento, idFoto, fechaFoto, obturacionCamaraFoto,idCamara, caracteristicaTecnicaCamara, descripcionCaracteristica)**

-Se perdio información? No, porque R1∩R2 = {idElemento}, la cual es clave en R1.  
-Se perdieron dependencias funcionales? No, porque vale df2 en R1 y valen df1 y df3 en R2  
-R1 cumple BCNF? Si, porque vale df1, donde idElemento es superclave.  
-R2 cumple BCNF? No, porque valen df1 y df3, donde por ej idFoto no es superclave. Particionamos usando df1.

**R3(idFoto, fechaFoto, obturacionCamaraFoto,idCamara)**

**R4 ( idElemento, idFoto, caracteristicaTecnicaCamara, descripcionCaracteristica)**

-Se perdio información? No, porque R3∩R4 = {idFoto}, la cual es superclave de R3.  
-Se perdieron dependencias funcionales? En R3 vale df1 pero df3 no vale en ningun esquema, aplicamos el algoritmo de perdida de dependencias funcionales:

Res = {caracteristicaTecnicaCamara}  
i=1  
Res = {caracteristicaTecnicaCamara} U {(({caracteristicaTecnicaCamara} ∩ {idFoto, idCamara, fechaFoto, obturacionCamaraFoto})+ ∩ {idFoto, idCamara, fechaFoto, obturacionCamaraFoto})} = {caracteristicaTecnicaCamara}

i=2  
Res = {caracteristicaTecnicaCamara} U {(({caracteristicaTecnicaCamara} ∩ {idElemento, descripcionElemento})+ ∩ {idElemento, descripcionElemento})} = {caracteristicaTecnicaCamara}

i=3  
Res = {caracteristicaTecnicaCamara} U {(({caracteristicaTecnicaCamara} ∩ {caracteristicaTecnicaCamara, idCamara, descripcionCaracteristica})+ ∩ {caracteristicaTecnicaCamara, idCamara, descripcionCaracteristica})}  
Res = {caracteristicaTecnicaCamara} U {(({caracteristicaTecnicaCamara})+ ∩ {caracteristicaTecnicaCamara, idCamara, descripcionCaracteristica})}

Hallamos ({caracteristicaTecnicaCamara})+:  
Result = {caracteristicaTecnicaCamara}  
1-Tomamos la df1 idFoto -> idCamara, fechaFoto, obturacionCamaraFoto, cómo idFoto no esta incluido en Result, no agrego nada a Result.  
2-Tomamos df2: idElemento -> descripcionElemento, cómo idElemento no esta incluido en Result, no agrego nada a Result.  
3-Tomamos df3: caracteristicaTecnicaCamara -> idCamara, descripcionCaracteristica, cómo caracteristicaTecnicaCamara esta incluido en Result, le agrego idCamara, descripcionCaracteristica.  
Result = {caracteristicaTecnicaCamara, idCamara, descripcionCaracteristica}  
Cómo cambio Result, vuelvo a iterar  
1-Tomamos la df1 idFoto -> idCamara, fechaFoto, obturacionCamaraFoto, cómo idFoto no esta incluido en Result, no agrego nada a Result.  
2-Tomamos df2: idElemento -> descripcionElemento, cómo idElemento no esta incluido en Result, no agrego nada a Result.

Volvi a recorrer las df y result no cambio. Retorna Result = {caracteristicaTecnicaCamara, idCamara, descripcionCaracteristica}

Res = {caracteristicaTecnicaCamara} U {({caracteristicaTecnicaCamara, idCamara, descripcionCaracteristica} ∩ {caracteristicaTecnicaCamara, idCamara, descripcionCaracteristica})}  
Res = {caracteristicaTecnicaCamara} U {caracteristicaTecnicaCamara, idCamara, descripcionCaracteristica}  
Res = {caracteristicaTecnicaCamara, idCamara, descripcionCaracteristica}  
Cómo Res cambio, vuelvo a iterar  
i=1  
Res = {caracteristicaTecnicaCamara, idCamara, descripcionCaracteristica} U {(({caracteristicaTecnicaCamara, idCamara, descripcionCaracteristica} ∩ {idFoto, idCamara, fechaFoto, obturacionCamaraFoto})+ ∩ {idFoto, idCamara, fechaFoto, obturacionCamaraFoto})}  
Res = {caracteristicaTecnicaCamara, idCamara, descripcionCaracteristica} U {(({idCamara})+ ∩ {idFoto, idCamara, fechaFoto, obturacionCamaraFoto})}

Hallamos ({idCamara})+  
Result = {idCamara}  
1-Tomamos df1 idFoto -> idCamara, fechaFoto, obturacionCamaraFoto, cómo idFoto no esta incluido en Result, no agrego nada a Result.  
2-Tomamos df2: idElemento -> descripcionElemento, cómo idElemento no esta incluido en Result, no agrego nada a Result.  
3-Tomamos df3: caracteristicaTecnicaCamara -> idCamara, descripcionCaracteristica, cómo caracteristicaTecnicaCamara no esta incluido en Result, no agrego nada.  
Cómo Result no cambia, termina la iteracion y retorna Result = {idCamara}

Res = {caracteristicaTecnicaCamara, idCamara, descripcionCaracteristica} U {({idCamara})}  
Res = {caracteristicaTecnicaCamara, idCamara, descripcionCaracteristica}  
Cómo Res no cambia, ya que junto con df2 no hay atributos en comun, termina la iteracion. No se encontraron todos los atributos, por lo tanto se perdio la df3.

**Aplico pasaje a 3FN  
  
R5(caracteristicaTecnicaCamara, descripcionCaracteristica, idCamara)**

**R6 ( idElemento, idFoto, caracteristicaTecnicaCamara)**

Finalmente nos quedan las relaciones en 3FN:  
**R1(idElemento, descripcionElemento)  
R3(idFoto, fechaFoto, obturacionCamaraFoto,idCamara)**  
**R5(caracteristicaTecnicaCamara, descripcionCaracteristica, idCamara)  
R6 ( idElemento, idFoto, caracteristicaTecnicaCamara)**

Clave: ( idElemento, idFoto, caracteristicaTecnicaCamara)  
Dependencias multivaluadas:  
 dm1: idFoto ->> idElemento  
 dm2: idFoto ->> caracteristicaTecnicaCamara

-R6 cumple 4FN? No, porque existe dm1 y dm2 que no son triviales. Particionamos usando dm1.  
  
**R7( idElemento, idFoto)**

**R8( idFoto, caracteristicaTecnicaCamara)**

-R7 cumple 4FN? Si, porque vale dm1 y es trivial.  
-R8 cumple 4FN? Si, porque vale dm2 y es trivial.

Entonces quedan las siguientes particiones:  
**R1(idElemento, descripcionElemento)  
R3(idFoto, fechaFoto, obturacionCamaraFoto,idCamara)**  
**R5(caracteristicaTecnicaCamara, descripcionCaracteristica, idCamara)  
R7( idElemento, idFoto)**  
**R8( idFoto, caracteristicaTecnicaCamara)**

7 - EMPRESA\_COLECTIVO (#Línea, #Ramal, #Colectivo, dniChofer, dniInspector, dniEmpleado, nombreLinea,nombreChofer, nombreInspector, nombreEmpleado)

Dependencias funcionales:  
 df1: dniChofer -> nombreChofer, #Ramal  
 df2: #Linea -> nombreLinea   
 df3: dniInspector -> nombreInspector  
 df4: dniEmpleado -> nombreEmpleado  
 df5: #Colectivo, #Linea -> #Ramal  
 df6: #Ramal -> #Linea

Cc: (dniChofer,dniInspector,dniEmpleado, #Colectivo)

-EMPRESA\_COLECTIVO cumple BCNF? No, porque valen df1 a df6, donde por ej #Linea no es superclave. Particionamos usando df2.

**R1(#Línea, nombreLinea)**

**R2(#Línea, #Ramal, #Colectivo, dniChofer, dniInspector, dniEmpleado,nombreChofer, nombreInspector, nombreEmpleado)**

-Se perdio información? No, porque R1∩R2 = {#Linea}, la cual es superclave de R1.  
-Se perdieron dependencias funcionales? No, porque vale df2 en R1 y df1 y de df3 a df6 en R2.  
-R1 cumple BCNF? Si, porque vale df2, donde #Linea es superclave.  
-R2 cumple BCNF? No, porque vale df1 y de df3 a df6, donde por ej dniInspector no es superclave. Particionamos usando df3.

**R3(dniInspector, nombreInspector)**

**R4(#Línea, #Ramal, #Colectivo, dniChofer, dniInspector, dniEmpleado,nombreChofer, nombreEmpleado)**

-Se perdio información? No, porque R3∩R4 = {dniInspector}, la cual es superclave de R3.  
-Se perdieron dependencias funcionales? No, porque vale df3 en R3 y vale df1 y de de df4 a df6 en R4.  
-R3 cumple BCNF? Si, porque vale df3, donde dniInspector es superclave.  
-R4 cumple BCNF? No, porque valen df1 y de df4 a df6, donde por ej dniEmpleado no es superclave. Particionamos usando df4.

**R5(dniEmpleado,nombreEmpleado)**

**R6(#Línea, #Ramal, #Colectivo, dniChofer, dniInspector, dniEmpleado, nombreChofer)**

-Se perdio información? No, porque R5∩R6 = {dniEmpleado}, la cual es superclave de R5.  
-Se perdieron dependencias funcionales? No, porque vale df4 en R5 y valen df1, df5 y df6 en R6.  
-R5 cumple BCNF? Si, porque vale df4, donde dniEmpleado es superclave.  
-R6 cumple BCNF? No, porque vale df1, df5 y df6, donde por ej #Ramal no es superclave. Particionamos usando df6.

**R7(#Ramal, #Línea)**

**R8(#Ramal, #Colectivo, dniChofer, dniInspector, dniEmpleado, nombreChofer)**

-Se perdio información? No, porque R7∩R8 = {#Ramal}, la cual es superclave de R7.  
-Se perdieron dependencias funcionales? No, porque vale df6 en R7 y valen df1 y df5 en R8.  
-R7 cumple BCNF? Si, porque vale df6, donde #Ramal es superclave.  
-R8 cumple BCNF? No, porque valen df1 y df5, donde por ej dniChofer no es superclave. Particionamos usando df1.

**R9(dniChofer, nombreChofer, #Ramal)**

**R10(#Colectivo, dniChofer, dniInspector, dniEmpleado)**

-Se perdio información? No, porque R9∩R10 = {dniChofer}, la cual es superclave de R9.  
-Se perdieron dependencias funcionales? Vale df1 en R9 pero no vale df5 en ningun esquema. Ejecutamos el algoritmo de perdida de dependencias funcionales.

Res = {#Colectivo, #Linea}  
i = 1  
Res = {#Colectivo, #Linea} U {(({#Colectivo, #Linea} ∩ {dniChofer, nombreChofer, #Ramal})+ ∩ {dniChofer, nombreChofer, #Ramal})} = {#Colectivo, #Linea}

i = 2  
Res = {#Colectivo, #Linea} U {(({#Colectivo, #Linea} ∩ {#Linea, nombreLinea})+ ∩ {#Linea, nombreLinea})}  
Res = {#Colectivo, #Linea} U {(({#Linea})+ ∩ {#Linea, nombreLinea})}

Hallamos (#Linea)+  
Result = {#Linea}  
1- Tomamos df1: dniChofer -> nombreChofer, #Ramal, cómo dniChofer no esta incluido en Result, no agrego nada en el mismo.  
2- Tomamos df2: #Linea -> nombreLinea, cómo #Linea esta incluido en Result, agregamos nombreLinea en Result. Result = {#Linea, nombreLinea}  
3 - Tomamos df3: dniInspector -> nombreInspector, cómo dniInspector no esta incluido en Result, no agregamos nada al mismo.  
4 - Tomamos df4: dniEmpleado -> nombreEmpleado, cómo dniEmpleado no esta incluido en Result, no agregamos nada al mismo.  
5- Tomamos df5: #Colectivo, #Linea -> #Ramal, cómo #Colectivo, #Linea no esta incluido en Result, no agregamos nada al mismo.   
6- Tomamos df6: #Ramal -> #Linea, cómo #Ramal no esta incluido en Result, no agregamos nada al mismo.  
  
Cómo Result cambio, volvemos a iterar.  
Result = {#Linea, nombreLinea}  
1- Tomamos df1: dniChofer -> nombreChofer, #Ramal, cómo dniChofer no esta incluido en Result, no agrego nada en el mismo.  
2- Tomamos df2: #Linea -> nombreLinea, cómo #Linea esta incluido en Result, pero no se agrega más nada en Result. Result = {#Linea, nombreLinea}  
3 - Tomamos df3: dniInspector -> nombreInspector, cómo dniInspector no esta incluido en Result, no agregamos nada al mismo.  
4 - Tomamos df4: dniEmpleado -> nombreEmpleado, cómo dniEmpleado no esta incluido en Result, no agregamos nada al mismo.  
5- Tomamos df5: #Colectivo, #Linea -> #Ramal, cómo #Colectivo, #Linea no esta incluido en Result, no agregamos nada al mismo.   
6- Tomamos df6: #Ramal -> #Linea, cómo #Ramal no esta incluido en Result, no agregamos nada al mismo.

Cómo result no cambio, corta la iteracion y se devuelve Result = {#Linea, nombreLinea}.

Res = {#Colectivo, #Linea} U {({#Linea, nombreLinea} ∩ {#Linea, nombreLinea})}  
Res = {#Colectivo, #Linea} U {#Linea, nombreLinea}  
Res = {#Colectivo, #Linea, nombreLinea}

i = 3  
Res = {#Colectivo, #Linea, nombreLinea} U {(({#Colectivo, #Linea, nombreLinea} ∩ {dniInspector, nombreInspector})+ ∩ {dniInspector, nombreInspector})} = {#Colectivo, #Linea, nombreLinea}

i = 4  
Res = {#Colectivo, #Linea, nombreLinea} U {(({#Colectivo, #Linea, nombreLinea} ∩ {dniEmpleado, nombreEmpleado})+ ∩ {dniEmpleado, nombreEmpleado})} = {#Colectivo, #Linea, nombreLinea

i = 5  
Res = {#Colectivo, #Linea, nombreLinea} U {(({#Colectivo, #Linea, nombreLinea} ∩ {#Colectivo, #Linea, #Ramal})+ ∩ {#Colectivo, #Linea, #Ramal})}  
Res = {#Colectivo, #Linea, nombreLinea} U {(({#Colectivo, #Linea})+ ∩ {#Colectivo, #Linea, #Ramal})}  
  
Hallamos ({#Colectivo, #Linea})+  
Result = {#Colectivo, #Linea}  
1- Tomamos df1: dniChofer -> nombreChofer, #Ramal, cómo dniChofer no esta incluido en Result, no agrego nada en el mismo.  
2- Tomamos df2: #Linea -> nombreLinea, cómo #Linea esta incluido en Result se agrega nombreLinea en Result. Result = {#Colectivo, #Linea, nombreLinea}  
3 - Tomamos df3: dniInspector -> nombreInspector, cómo dniInspector no esta incluido en Result, no agregamos nada al mismo.  
4 - Tomamos df4: dniEmpleado -> nombreEmpleado, cómo dniEmpleado no esta incluido en Result, no agregamos nada al mismo.  
5- Tomamos df5: #Colectivo, #Linea -> #Ramal, cómo #Colectivo, #Linea esta incluido en Result, se agrega #Ramal. Result = {#Colectivo, #Linea, nombreLinea, #Ramal}  
6- Tomamos df6: #Ramal -> #Linea, cómo #Ramal esta incluido en Result, no agregamos #Linea porque ya esta incluido.

Cómo Result cambio, volvemos a iterar.  
Result = {#Colectivo, #Linea, nombreLinea, #Ramal}  
1- Tomamos df1: dniChofer -> nombreChofer, #Ramal, cómo dniChofer no esta incluido en Result, no agrego nada en el mismo.  
2- Tomamos df2: #Linea -> nombreLinea, cómo #Linea esta incluido en Result no se agrega nombreLinea porque ya esta incluido.  
3 - Tomamos df3: dniInspector -> nombreInspector, cómo dniInspector no esta incluido en Result, no agregamos nada al mismo.  
4 - Tomamos df4: dniEmpleado -> nombreEmpleado, cómo dniEmpleado no esta incluido en Result, no agregamos nada al mismo.  
5- Tomamos df5: #Colectivo, #Linea -> #Ramal, cómo #Colectivo, #Linea esta incluido en Result, no se agrega #Ramal porque ya esta incluido.  
6- Tomamos df6: #Ramal -> #Linea, cómo #Ramal esta incluido en Result, no agregamos #Linea porque ya esta incluido.

Cómo Result no cambio, corta la iteracion y se devuelve {#Colectivo, #Linea, nombreLinea, #Ramal}.

Res = {#Colectivo, #Linea, nombreLinea} U {(({#Colectivo, #Linea, nombreLinea, #Ramal})+ ∩ {#Colectivo, #Linea, #Ramal})}  
Res = {#Colectivo, #Linea, nombreLinea} U {({#Colectivo, #Linea,#Ramal})}  
Res = {#Colectivo, #Linea, nombreLinea,#Ramal}

i = 6  
Res = {#Colectivo, #Linea, nombreLinea,#Ramal} U {(({#Colectivo, #Linea, nombreLinea,#Ramal} ∩ {#Ramal, #Linea})+ ∩ {#Ramal, #Linea})}  
Res = {#Colectivo, #Linea, nombreLinea,#Ramal} U {(({#Ramal, #Linea})+ ∩ {#Ramal, #Linea})}  
  
Hallamos ({#Ramal, #Linea})+.  
Result = {#Ramal, #Linea}  
1- Tomamos df1: dniChofer -> nombreChofer, #Ramal, cómo dniChofer no esta incluido en Result, no agrego nada en el mismo.  
2- Tomamos df2: #Linea -> nombreLinea, cómo #Linea esta incluido en Result, se agrega nombreLinea. Result = {#Ramal, #Linea, nombreLinea}  
3 - Tomamos df3: dniInspector -> nombreInspector, cómo dniInspector no esta incluido en Result, no agregamos nada al mismo.  
4 - Tomamos df4: dniEmpleado -> nombreEmpleado, cómo dniEmpleado no esta incluido en Result, no agregamos nada al mismo.  
5- Tomamos df5: #Colectivo, #Linea -> #Ramal, cómo #Colectivo, #Linea no esta incluido en Result, no se agrega nada.  
6- Tomamos df6: #Ramal -> #Linea, cómo #Ramal esta incluido en Result, no agregamos #Linea porque ya esta incluido.

Cómo Result cambio, se vuelve a iterar.  
Result = {#Ramal, #Linea, nombreLinea}  
1- Tomamos df1: dniChofer -> nombreChofer, #Ramal, cómo dniChofer no esta incluido en Result, no agrego nada en el mismo.  
2- Tomamos df2: #Linea -> nombreLinea, cómo #Linea esta incluido en Result, no se agrega nombreLinea porque también esta incluido.  
3 - Tomamos df3: dniInspector -> nombreInspector, cómo dniInspector no esta incluido en Result, no agregamos nada al mismo.  
4 - Tomamos df4: dniEmpleado -> nombreEmpleado, cómo dniEmpleado no esta incluido en Result, no agregamos nada al mismo.  
5- Tomamos df5: #Colectivo, #Linea -> #Ramal, cómo #Colectivo, #Linea no esta incluido en Result, no se agrega nada.  
6- Tomamos df6: #Ramal -> #Linea, cómo #Ramal esta incluido en Result, no agregamos #Linea porque ya esta incluido.

Cómo Result no cambio, se corta la iteracion y se devuelve {#Ramal, #Linea, nombreLinea}.

Res = {#Colectivo, #Linea, nombreLinea,#Ramal} U {({#Ramal, #Linea, nombreLinea} ∩ {#Ramal, #Linea})}  
Res = {#Colectivo, #Linea, nombreLinea,#Ramal} U {{#Ramal, #Linea}}  
Res = {#Colectivo, #Linea, nombreLinea,#Ramal}

Cómo Res cambio, se vuelve a iterar  
Res = {#Colectivo, #Linea, nombreLinea,#Ramal}  
i = 1  
Res = {#Colectivo, #Linea, nombreLinea,#Ramal} U {(({#Colectivo, #Linea, nombreLinea,#Ramal} ∩ {dniChofer, nombreChofer, #Ramal})+ ∩ {dniChofer -> nombreChofer, #Ramal})}  
Res = {#Colectivo, #Linea, nombreLinea,#Ramal} U {(({#Ramal})+ ∩ {dniChofer, nombreChofer, #Ramal})}

Hallamos ({#Ramal})+.  
Result = {#Ramal}  
1- Tomamos df1: dniChofer -> nombreChofer, #Ramal, cómo dniChofer no esta incluido en Result, no agrego nada en el mismo.  
2- Tomamos df2: #Linea -> nombreLinea, cómo #Linea no esta incluido en Result, no se agrega nada.  
3 - Tomamos df3: dniInspector -> nombreInspector, cómo dniInspector no esta incluido en Result, no agregamos nada al mismo.  
4 - Tomamos df4: dniEmpleado -> nombreEmpleado, cómo dniEmpleado no esta incluido en Result, no agregamos nada al mismo.  
5- Tomamos df5: #Colectivo, #Linea -> #Ramal, cómo #Colectivo, #Linea no esta incluido en Result, no se agrega nada.  
6- Tomamos df6: #Ramal -> #Linea, cómo #Ramal esta incluido en Result, agregamos #Linea. Result = {#Ramal,#Linea}

Cómo cambio Result, se vuelve a iterar.  
Result = {#Ramal,#Linea}  
1- Tomamos df1: dniChofer -> nombreChofer, #Ramal, cómo dniChofer no esta incluido en Result, no agrego nada en el mismo.  
2- Tomamos df2: #Linea -> nombreLinea, cómo #Linea esta incluido en Result, agregamos nombreLinea. Result = {#Ramal,#Linea,nombreLinea}  
3 - Tomamos df3: dniInspector -> nombreInspector, cómo dniInspector no esta incluido en Result, no agregamos nada al mismo.  
4 - Tomamos df4: dniEmpleado -> nombreEmpleado, cómo dniEmpleado no esta incluido en Result, no agregamos nada al mismo.  
5- Tomamos df5: #Colectivo, #Linea -> #Ramal, cómo #Colectivo, #Linea no esta incluido en Result, no se agrega nada.  
6- Tomamos df6: #Ramal -> #Linea, #Ramal esta incluido en Result, no agregamos #Linea porque ya esta incluido en Result.

Cómo Result cambio, volvemos a iterar  
Result = {#Ramal,#Linea,nombreLinea}  
1- Tomamos df1: dniChofer -> nombreChofer, #Ramal, cómo dniChofer no esta incluido en Result, no agrego nada en el mismo.  
2- Tomamos df2: #Linea -> nombreLinea, #Linea esta incluido en Result, no agregamos nombreLinea porque esta incluido también.  
3 - Tomamos df3: dniInspector -> nombreInspector, cómo dniInspector no esta incluido en Result, no agregamos nada al mismo.  
4 - Tomamos df4: dniEmpleado -> nombreEmpleado, cómo dniEmpleado no esta incluido en Result, no agregamos nada al mismo.  
5- Tomamos df5: #Colectivo, #Linea -> #Ramal, cómo #Colectivo, #Linea no esta incluido en Result, no se agrega nada.  
6- Tomamos df6: #Ramal -> #Linea, #Ramal esta incluido en Result, no agregamos #Linea porque ya esta incluido en Result.

Cómo no cambio Result, corta la iteracion y devuelve {#Ramal,#Linea,nombreLinea}

Res = {#Colectivo, #Linea, nombreLinea,#Ramal} U {({#Ramal})}  
Res = {#Colectivo, #Linea, nombreLinea,#Ramal}

i = 2  
Res = {#Colectivo, #Linea, nombreLinea,#Ramal} U {(({#Colectivo, #Linea, nombreLinea,#Ramal} ∩ {#Linea, nombreLinea})+ ∩ {#Linea, nombreLinea})}  
Res = {#Colectivo, #Linea, nombreLinea,#Ramal} U {(({#Linea, nombreLinea})+ ∩ {#Linea, nombreLinea})}  
  
Cuando ejecuto ({#Linea, nombreLinea})+, no voy a obtener datos distintos a lo contenido en Res. Por lo tanto

Res = {#Colectivo, #Linea, nombreLinea,#Ramal}

i = 3  
Res = {#Colectivo, #Linea, nombreLinea,#Ramal} U {(({#Colectivo, #Linea, nombreLinea,#Ramal} ∩ {dniInspector, nombreInspector})+ ∩ {dniInspector, nombreInspector})} = {#Colectivo, #Linea, nombreLinea,#Ramal}

i = 4  
Res = {#Colectivo, #Linea, nombreLinea,#Ramal} U {(({#Colectivo, #Linea, nombreLinea,#Ramal} ∩ {dniEmpleado, nombreEmpleado})+ ∩ {dniEmpleado, nombreEmpleado})} = {#Colectivo, #Linea, nombreLinea,#Ramal}

i = 5  
Res = {#Colectivo, #Linea, nombreLinea,#Ramal} U {(({#Colectivo, #Linea, nombreLinea,#Ramal} ∩ {#Colectivo, #Linea, #Ramal})+ ∩ {#Colectivo, #Linea, #Ramal})}  
Res = {#Colectivo, #Linea, nombreLinea,#Ramal} U {(({#Colectivo, #Linea, #Ramal})+ ∩ {#Colectivo, #Linea, #Ramal})}

Cuando ejecuto ({#Colectivo, #Linea, #Ramal})+, no voy a obtener datos distintos a lo contenido en Res. Por lo tanto

Res = {#Colectivo, #Linea, nombreLinea,#Ramal}

i = 6  
Res = {#Colectivo, #Linea, nombreLinea,#Ramal} U {(({#Colectivo, #Linea, nombreLinea,#Ramal} ∩ {#Ramal, #Linea})+ ∩ {#Ramal, #Linea})}  
Res = {#Colectivo, #Linea, nombreLinea,#Ramal} U {(({#Ramal, #Linea})+ ∩ {#Ramal, #Linea})}

Cuando ejecuto ({#Ramal, #Linea})+, no voy a obtener datos distintos a lo contenido en Res. Por lo tanto

Res = {#Colectivo, #Linea, nombreLinea,#Ramal}

Cómo Res no cambio, se corta la iteracion

**Finalmente Res = {#Colectivo, #Linea, nombreLinea,#Ramal} y no se encontraron todos los atributos de EMPRESA\_COLECTIVO, quiere decir que se perdio la df5, por lo tanto no se puede llevar a BCNF y aplico 3FN.**

**R11(#Colectivo, #Linea, #Ramal)**

**R12(#Colectivo, dniChofer, dniInspector, dniEmpleado)**

Finalmente nos quedan los siguientes esquemas en 3FN.  
**R1(#Línea, nombreLinea)  
R3(dniInspector, nombreInspector)**  
**R5(dniEmpleado,nombreEmpleado)**  
**R7(#Ramal, #Línea)**  
**R9(dniChofer, nombreChofer, #Ramal)**  
**R11(#Colectivo, #Linea, #Ramal)  
R12(#Colectivo, dniChofer, dniInspector, dniEmpleado)**

Dependencias multivaluadas:  
 dm1: ø ->> #Colectivo  
 dm2: ø ->> dniChofer  
 dm3: ø ->> dniInspector  
 dm4: ø ->> dniEmpleado

-R12 cumple 4FN? No, porque existen dm1 a dm4, donde por ej dm1 no es trivial. Particionamos usando dm1.

**R13(#Colectivo)**

**R14(dniChofer, dniInspector, dniEmpleado)**

-R13 cumple 4FN? Si, porque vale dm1 y es trivial.  
-R14 cumple 4FN? No, porque valen dm2 a dm4, donde por ej dm2 no es trivial. Particionamos usando dm2.

**R15(dniChofer)**

**R16(dniInspector, dniEmpleado)**

-R15 cumple 4FN? Si, porque vale dm2 y es trivial.  
-R15 cumple 4FN? No, porque valen dm3 y dm4, donde por ej dm3 no es trivial. Particionamos usando dm3.

**R17(dniInspector)**

**R18(dniEmpleado)**

-R17 cumple 4FN? Si, porque vale dm3 y es trivial.  
-R18 cumple 4FN? Si, porque vale dm4 y es trivial.

Cómo R13, R15, R17 y R18 son proyecciones de atributos claves de esquemas anteriores, finalmente te queda:

**R1(#Línea, nombreLinea)  
R3(dniInspector, nombreInspector)**  
**R5(dniEmpleado,nombreEmpleado)**  
**R7(#Ramal, #Línea)**  
**R9(dniChofer, nombreChofer, #Ramal)**  
**R11(#Colectivo, #Linea, #Ramal)**

8 - INTERNACION (codHospital, cantidadHabitaciones, direcciónInternacionPaciente, telefonoInternacionPaciente, dniPaciente, domicilioPaciente, nombreApellidoPaciente, domicilioHospital,ciudadHospital, directorHospital, fechaInicioInternacion, cantDiasIntenacion, doctorQueAtiendePaciente,insumoEmpleadoInternación)

Dependencias funcionales:  
 df1: codHospital -> cantidadHabitaciones, directorHospital,domicilioHospital,ciudadHospital  
 df2: dniPaciente, fechaInicioInternacion -> domicilioHospital, ciudadHospital, cantDiasInternacion, direcciónInternacionPaciente, telefonoInternacionPaciente  
 df3: dniPaciente, fechaInicioInternacion -> codHospital,cantDiasInternacion, direcciónInternacionPaciente, telefonoInternacionPaciente  
 df4: codHospital, dniPaciente, fechaInicioInternacion -> direcciónInternacionPaciente, telefonoInternacionPaciente  
 df5: dniPaciente -> domicilioPaciente, nombreApellidoPaciente

Cc:{dniPaciente,fechaInternacion,doctorQueAtiendePaciente,insumoEmpleadoInternación}

-INTERNACION cumple BCNF? No, porque existe df1 a df5, donde por ej dniPaciente no es superclave. Particionamos usando df5.

**R1(dniPaciente, domicilioPaciente, nombreApellidoPaciente)**

**R2(codHospital, cantidadHabitaciones, direcciónInternacionPaciente, telefonoInternacionPaciente, dniPaciente, domicilioHospital,ciudadHospital, directorHospital, fechaInicioInternacion, cantDiasIntenacion, doctorQueAtiendePaciente, insumoEmpleadoInternación)**

-Se perdio información? No, porque R1∩R2 = {dniPaciente}, la cual es superclave de R1.  
-Se perdieron dependencias funcionales? No, porque vale df5 en R1 y valen df1 a df4 en R2.  
-R1 cumple BCNF? Si, porque vale df5, donde dniPaciente es superclave.  
-R2 cumple BCNF? No, porque valen df1 a df4, donde por ej codHospital no es superclave. Particionamos usando df1.

**R3(codHospital, cantidadHabitaciones,domicilioHospital,ciudadHospital, directorHospital)**

**R4(codHospital, direcciónInternacionPaciente, telefonoInternacionPaciente, dniPaciente, fechaInicioInternacion, cantDiasIntenacion, doctorQueAtiendePaciente, insumoEmpleadoInternación)**

-Se perdio información? No, porque R3∩R4 = {codHospital}, la cual es superclave de R3.  
-Se perdieron dependencias funcionales? Vale df1 en R4 y valen df3 y df4 en R4. Se perdio df2? Aplicamos el algoritmo de perdida de dependencias funcionales.

Algoritmo de perdida de dependencias funcionales.

Res = {dniPaciente, fechaInicioInternacion}  
i = 1  
Res = {dniPaciente, fechaInicioInternacion} U (({dniPaciente, fechaInicioInternacion} ∩ {codHospital, cantidadHabitaciones, directorHospital,domicilioHospital,ciudadHospital})+ ∩ {codHospital, cantidadHabitaciones, directorHospital,domicilioHospital,ciudadHospital}) = {dniPaciente, fechaInicioInternacion}

i = 2  
Res = {dniPaciente, fechaInicioInternacion} U (({dniPaciente, fechaInicioInternacion} ∩ {dniPaciente, fechaInicioInternacion, domicilioHospital, ciudadHospital, cantDiasInternacion, direcciónInternacionPaciente, telefonoInternacionPaciente})+ ∩ {dniPaciente, fechaInicioInternacion, domicilioHospital, ciudadHospital, cantDiasInternacion, direcciónInternacionPaciente, telefonoInternacionPaciente})  
Res = {dniPaciente, fechaInicioInternacion} U (({dniPaciente, fechaInicioInternacion})+ ∩ {dniPaciente, fechaInicioInternacion, domicilioHospital, ciudadHospital, cantDiasInternacion, direcciónInternacionPaciente, telefonoInternacionPaciente})

Algoritmo de clausura ({dniPaciente, fechaInicioInternacion})+:  
Result = {dniPaciente, fechaInicioInternacion}  
-Tomamos df1: codHospital -> cantidadHabitaciones, directorHospital, domicilioHospital, ciudadHospital. Cómo codHospital no esta incluido en Result, no agregamos nada en Result.  
-Tomamos df2: dniPaciente, fechaInicioInternacion -> domicilioHospital, ciudadHospital, cantDiasInternacion, direcciónInternacionPaciente, telefonoInternacionPaciente. Cómo dniPaciente, fechaInicioInternacion esta incluido en Result, agregamos los determinados en Result.   
Result = {dniPaciente, fechaInicioInternacion, domicilioHospital, ciudadHospital, cantDiasInternacion, direcciónInternacionPaciente, telefonoInternacionPaciente}  
-Tomamos df3: dniPaciente, fechaInicioInternacion -> codHospital,cantDiasInternacion, direcciónInternacionPaciente, telefonoInternacionPaciente. Cómo {dniPaciente, fechaInicioInternacion} esta incluido en Result, agregamos los determinados no repetidos en Result.  
Result = {dniPaciente, fechaInicioInternacion, domicilioHospital, ciudadHospital, cantDiasInternacion, direcciónInternacionPaciente, telefonoInternacionPaciente,codHospital}   
-Tomamos df4: codHospital, dniPaciente, fechaInicioInternacion -> direcciónInternacionPaciente, telefonoInternacionPaciente. {codHospital, dniPaciente, fechaInicioInternacion} esta incluido en Result, pero cómo los determinados también estan incluidos, no agregamos nada en Result.  
-Tomamos df5: dniPaciente -> domicilioPaciente, nombreApellidoPaciente. Cómo {dniPaciente} esta incluido en Result, agregamos {domicilioPaciente, nombreApellidoPaciente} en Result.   
Result = {dniPaciente, fechaInicioInternacion, domicilioHospital, ciudadHospital, cantDiasInternacion, direcciónInternacionPaciente, telefonoInternacionPaciente,codHospital,domicilioPaciente, nombreApellidoPaciente}

Cómo Result cambio, se vuelve a iterar.  
Result = {dniPaciente, fechaInicioInternacion, domicilioHospital, ciudadHospital, cantDiasInternacion, direcciónInternacionPaciente, telefonoInternacionPaciente,codHospital,domicilioPaciente, nombreApellidoPaciente}  
-Tomamos df1: codHospital -> cantidadHabitaciones, directorHospital, domicilioHospital, ciudadHospital. Cómo {codHospital} esta incluido en Result, agregamos los determinados en Result.  
Result = {dniPaciente, fechaInicioInternacion, domicilioHospital, ciudadHospital, cantDiasInternacion, direcciónInternacionPaciente, telefonoInternacionPaciente,codHospital,domicilioPaciente, nombreApellidoPaciente,cantidadHabitaciones,directorHospital}

Se obtuvieron todos los atributos, por lo tanto se retorna Result = {dniPaciente, fechaInicioInternacion, domicilioHospital, ciudadHospital, cantDiasInternacion, direcciónInternacionPaciente, telefonoInternacionPaciente,codHospital,domicilioPaciente, nombreApellidoPaciente,cantidadHabitaciones,directorHospital}

Res = {dniPaciente, fechaInicioInternacion} U ({dniPaciente, fechaInicioInternacion, domicilioHospital, ciudadHospital, cantDiasInternacion, direcciónInternacionPaciente, telefonoInternacionPaciente,codHospital,domicilioPaciente, nombreApellidoPaciente,cantidadHabitaciones,directorHospital} ∩ {dniPaciente, fechaInicioInternacion, domicilioHospital, ciudadHospital, cantDiasInternacion, direcciónInternacionPaciente, telefonoInternacionPaciente})  
Res = {dniPaciente, fechaInicioInternacion} U {dniPaciente, fechaInicioInternacion, domicilioHospital, ciudadHospital, cantDiasInternacion, direcciónInternacionPaciente, telefonoInternacionPaciente}  
Res = {dniPaciente, fechaInicioInternacion, domicilioHospital, ciudadHospital, cantDiasInternacion, direcciónInternacionPaciente, telefonoInternacionPaciente}

Cómo se encontraron todos los atributos determinados en df2, no se pierde la dependencia funcional.

-R3 cumple BCNF? Si, porque vale df1, donde codHospital es superclave.  
-R4 cumple BCNF? No, porque vale df2 a df4, donde por ej {dniPaciente, fechaInicioInternacion} no es superclave. Particionamos usando df3.

**R5(dniPaciente, fechaInicioInternacion,codHospital, direcciónInternacionPaciente, telefonoInternacionPaciente, cantDiasIntenacion)**

**R6(dniPaciente, fechaInicioInternacion, doctorQueAtiendePaciente, insumoEmpleadoInternación)**

-Se perdio información? No, porque R5∩R6 = {dniPaciente, fechaInicioInternacion}, la cual es superclave de R5.  
-Se perdieron dependencias funcionales? Vale df3 en R5. Se perdieron df2 y df4?

Algoritmo de perdida de dependencias funcionales.  
(Uso el determinante de df4, ya que df2 esta incluido en df4)  
Res = {codHospital, dniPaciente, fechaInicioInternacion}  
i = 1  
Res = {codHospital, dniPaciente, fechaInicioInternacion} U (({codHospital, dniPaciente, fechaInicioInternacion} ∩ {codHospital, cantidadHabitaciones, directorHospital,domicilioHospital,ciudadHospital})+ ∩ {codHospital, cantidadHabitaciones, directorHospital,domicilioHospital,ciudadHospital})  
Res = {codHospital, dniPaciente, fechaInicioInternacion} U (({codHospital})+ ∩ {codHospital, cantidadHabitaciones, directorHospital,domicilioHospital,ciudadHospital})

Algoritmo de clausura ({codHospital})+:  
Result = {codHospital}  
-Tomamos df1: codHospital -> cantidadHabitaciones, directorHospital, domicilioHospital, ciudadHospital. Cómo {codHospital} esta incluido en Result, agregamos los determinados en Result.  
Result = {codHospital,cantidadHabitaciones, directorHospital, domicilioHospital, ciudadHospital}  
-Tomamos df2: dniPaciente, fechaInicioInternacion -> domicilioHospital, ciudadHospital, cantDiasInternacion, direcciónInternacionPaciente, telefonoInternacionPaciente. Cómo {dniPaciente, fechaInicioInternacion} no esta incluido en Result, no se agrega nada.  
-Tomamos df3: dniPaciente, fechaInicioInternacion -> codHospital,cantDiasInternacion, direcciónInternacionPaciente, telefonoInternacionPaciente. Cómo {dniPaciente, fechaInicioInternacion} no esta incluido en Result, no se agrega nada.  
-Tomamos df4: codHospital, dniPaciente, fechaInicioInternacion -> direcciónInternacionPaciente, telefonoInternacionPaciente. Cómo {codHospital, dniPaciente, fechaInicioInternacion} no esta incluido en Result, no agregamos nada.  
-Tomamos df5: dniPaciente -> domicilioPaciente, nombreApellidoPaciente. Cómo {dniPaciente} no esta incluido en Result, no agregamos nada.

Result cambio, entonces se debe volver a iterar. Pero cómo ya se sabe que el mismo no va a cambiar, se retorna Result = {codHospital,cantidadHabitaciones, directorHospital, domicilioHospital, ciudadHospital}

Res = {codHospital, dniPaciente, fechaInicioInternacion} U ({codHospital,cantidadHabitaciones, directorHospital, domicilioHospital, ciudadHospital} ∩ {codHospital, cantidadHabitaciones, directorHospital,domicilioHospital,ciudadHospital})  
Res = {codHospital, dniPaciente, fechaInicioInternacion} U {codHospital,cantidadHabitaciones, directorHospital, domicilioHospital, ciudadHospital}  
Res = {codHospital, dniPaciente, fechaInicioInternacion,cantidadHabitaciones, directorHospital, domicilioHospital, ciudadHospital}

i = 2  
Res = {codHospital, dniPaciente, fechaInicioInternacion,cantidadHabitaciones, directorHospital, domicilioHospital, ciudadHospital} U (({codHospital, dniPaciente, fechaInicioInternacion,cantidadHabitaciones, directorHospital, domicilioHospital, ciudadHospital} ∩ {dniPaciente, fechaInicioInternacion, domicilioHospital, ciudadHospital, cantDiasInternacion, direcciónInternacionPaciente, telefonoInternacionPaciente})+ ∩ {dniPaciente, fechaInicioInternacion, domicilioHospital, ciudadHospital, cantDiasInternacion, direcciónInternacionPaciente, telefonoInternacionPaciente})  
Res = {codHospital, dniPaciente, fechaInicioInternacion,cantidadHabitaciones, directorHospital, domicilioHospital, ciudadHospital} U (({dniPaciente, fechaInicioInternacion, domicilioHospital, ciudadHospital})+ ∩ {dniPaciente, fechaInicioInternacion, domicilioHospital, ciudadHospital, cantDiasInternacion, direcciónInternacionPaciente, telefonoInternacionPaciente})

Algoritmo de clausura ({dniPaciente, fechaInicioInternacion, domicilioHospital, ciudadHospital})+:  
Result = {dniPaciente, fechaInicioInternacion, domicilioHospital, ciudadHospital}  
-Tomamos df1: codHospital -> cantidadHabitaciones, directorHospital, domicilioHospital, ciudadHospital. Cómo {codHospital} no esta incluido en Result, no se agrega nada.  
-Tomamos df2: dniPaciente, fechaInicioInternacion -> domicilioHospital, ciudadHospital, cantDiasInternacion, direcciónInternacionPaciente, telefonoInternacionPaciente. Cómo {dniPaciente, fechaInicioInternacion} esta incluido en Result, agregamos los determinados al mismo.  
Result = {dniPaciente, fechaInicioInternacion, domicilioHospital, ciudadHospital, cantDiasInternacion, direcciónInternacionPaciente, telefonoInternacionPaciente}  
-Tomamos df3: dniPaciente, fechaInicioInternacion -> codHospital,cantDiasInternacion, direcciónInternacionPaciente, telefonoInternacionPaciente. Cómo {dniPaciente, fechaInicioInternacion}. Cómo {dniPaciente, fechaInicioInternacion} esta incluido en Result, agregamos los determinados no repetidos en Result.  
Result = {dniPaciente, fechaInicioInternacion, domicilioHospital, ciudadHospital, cantDiasInternacion, direcciónInternacionPaciente, telefonoInternacionPaciente,codHospital}  
-Tomamos df4: codHospital, dniPaciente, fechaInicioInternacion -> direcciónInternacionPaciente, telefonoInternacionPaciente.{codHospital, dniPaciente, fechaInicioInternacion} esta incluido en Result, pero cómo los determinados también estan incluido, no se agrega nada.  
-Tomamos df5: dniPaciente -> domicilioPaciente, nombreApellidoPaciente. Cómo {dniPaciente} esta incluido en Result, agregamos los determinados al mismo.   
Result = {dniPaciente, fechaInicioInternacion, domicilioHospital, ciudadHospital, cantDiasInternacion, direcciónInternacionPaciente, telefonoInternacionPaciente,codHospital,domicilioPaciente, nombreApellidoPaciente}

Cómo Result cambio, se vuelve a iterar.  
Result = {dniPaciente, fechaInicioInternacion, domicilioHospital, ciudadHospital, cantDiasInternacion, direcciónInternacionPaciente, telefonoInternacionPaciente,codHospital,domicilioPaciente, nombreApellidoPaciente}  
-Tomamos df1: codHospital -> cantidadHabitaciones, directorHospital, domicilioHospital, ciudadHospital. Cómo {codHospital} esta incluido en Result, agregamos los determinados no repetidos en el mismo. Result = {dniPaciente, fechaInicioInternacion, domicilioHospital, ciudadHospital, cantDiasInternacion, direcciónInternacionPaciente, telefonoInternacionPaciente,codHospital,domicilioPaciente, nombreApellidoPaciente, cantidadHabitaciones,directorHospital}

Se encontraron todos los atributos, por lo tanto se retorna Result = {dniPaciente, fechaInicioInternacion, domicilioHospital, ciudadHospital, cantDiasInternacion, direcciónInternacionPaciente, telefonoInternacionPaciente,codHospital,domicilioPaciente, nombreApellidoPaciente, cantidadHabitaciones,directorHospital}.

Res = {codHospital, dniPaciente, fechaInicioInternacion,cantidadHabitaciones, directorHospital, domicilioHospital, ciudadHospital} U ({dniPaciente, fechaInicioInternacion, domicilioHospital, ciudadHospital, cantDiasInternacion, direcciónInternacionPaciente, telefonoInternacionPaciente, codHospital,domicilioPaciente, nombreApellidoPaciente, cantidadHabitaciones,directorHospital} ∩ {dniPaciente, fechaInicioInternacion, domicilioHospital, ciudadHospital, cantDiasInternacion, direcciónInternacionPaciente, telefonoInternacionPaciente})  
Res = {codHospital, dniPaciente, fechaInicioInternacion,cantidadHabitaciones, directorHospital, domicilioHospital, ciudadHospital} U {dniPaciente, fechaInicioInternacion, domicilioHospital, ciudadHospital, cantDiasInternacion, direcciónInternacionPaciente, telefonoInternacionPaciente}  
Res = {codHospital, dniPaciente, fechaInicioInternacion,cantidadHabitaciones, directorHospital, domicilioHospital, ciudadHospital, cantDiasInternacion, direcciónInternacionPaciente, telefonoInternacionPaciente}

Cómo en Res ya tengo todos los atributos determinados en df4, no se pierde la dependencia funcional.

-R5 cumple BCNF? Si, porque existe df2, donde {dniPaciente, fechaInicioInternacion} es superclave.  
-R6 cumple BCNF? Si, porque las unicas dependencias funcionales que hay, son triviales

Esquemas en BCNF:  
**R1(dniPaciente, domicilioPaciente, nombreApellidoPaciente)  
R3(codHospital, cantidadHabitaciones,domicilioHospital,ciudadHospital, directorHospital)  
R5(dniPaciente, fechaInicioInternacion,codHospital, direcciónInternacionPaciente, telefonoInternacionPaciente, cantDiasIntenacion)  
R6(dniPaciente, fechaInicioInternacion, doctorQueAtiendePaciente, insumoEmpleadoInternación)**

Clave: {dniPaciente, fechaInicioInternacion, doctorQueAtiendePaciente, insumoEmpleadoInternación}

Dependencias multivaluadas:  
 dm1: dniPaciente, fechaInicioInternacion ->> doctorQueAtiendePaciente  
 dm2: dniPaciente, fechaInicioInternacion ->> insumoEmpleadoInternación

-R6 cumple 4FN? No, porque existe dm1 y esta no es trivial. Particionamos usando dm1.

**R7(dniPaciente, fechaInicioInternacion, doctorQueAtiendePaciente)**

**R8(dniPaciente, fechaInicioInternacion, insumoEmpleadoInternación)**

-R7 cumple 4FN? Si, porque existe dm1 y es trivial.  
-R8 cumple 4FN? Si, porque existe dm2 y es trivial.

Finalmente, nos quedan los siguiente esquemas en 4FN:  
**R1(dniPaciente, domicilioPaciente, nombreApellidoPaciente)  
R3(codHospital, cantidadHabitaciones,domicilioHospital,ciudadHospital, directorHospital)  
R5(dniPaciente, fechaInicioInternacion,codHospital, direcciónInternacionPaciente, telefonoInternacionPaciente, cantDiasIntenacion)  
R7(dniPaciente, fechaInicioInternacion, doctorQueAtiendePaciente)  
R8(dniPaciente, fechaInicioInternacion, insumoEmpleadoInternación)**

9 - INFRACCIONES\_REALIZADAS (#auto, modeloAuto, #cedula, #conductor, fechaVto, #propietario, #infraccion,fechaInfraccion, tipoInfraccion)

Dependencias funcionales:  
 df1: #cedula -> #auto, #conductor, fechaVto  
 df2: #infraccion -> #cedula, fechaInfraccion, tipoInfraccion  
 df3: #auto -> modeloAuto

Cc: {#infraccion,#propietario}

-INFRACCIONES\_REALIZADAS cumple BCNF? No, porque existen df1 a df3, donde por ej #auto no es superclave. Particionamos usando df3.

**R1(#auto, modeloAuto)**

**R2(#auto, #cedula, #conductor, fechaVto, #propietario, #infraccion,fechaInfraccion, tipoInfraccion)**

-Se perdio información? No, porque R1∩R2 = {#auto}, la cual es superclave de R1.  
-Se perdieron dependencias funcionales? No, porque vale df3 en R1 y valen df1 y df2 en R2.  
-R1 cumple BCNF? Si, porque vale df3, donde {#auto} es superclave.  
-R2 cumple BCNF? No, porque valen df1 y df2, donde por ej {#cedula} no es superclave. Particionamos usando df1.

**R3(#cedula,#auto,#conductor, fechaVto)**

**R4( #cedula, #propietario, #infraccion,fechaInfraccion, tipoInfraccion)**

-Se perdio información? No, porque R3∩R4 = {#cedula}, la cual es superclave de R3.  
-Se perdieron dependencias funcionales? No, porque vale df1 en R1 y vale df2 en R4.  
-R3 cumple BCNF? Si, porque vale df1, donde {#cedula} es superclave.  
-R4 cumple BCNF? No, porque valen df2, donde {#infraccion} no es superclave. Particionamos usando df2.

**R5(#infraccion,#cedula,fechaInfraccion, tipoInfraccion)**

**R6(#propietario, #infraccion)**

-Se perdio información? No, porque R5∩R6 = {#infraccion}, la cual es superclave de R5.  
-Se perdieron dependencias funcionales? No, porque vale df2 en R5.  
-R5 cumple BCNF? Si, porque vale df2, donde {#infraccion} es superclave.  
-R6 cumple BCNF? Si, porque las dependencias funcionales que hay son triviales.

Esquemas en BCNF:  
**R1(#auto, modeloAuto)**  
**R3(#cedula,#auto,#conductor, fechaVto)**  
**R5(#infraccion,#cedula,fechaInfraccion, tipoInfraccion)  
R6(#propietario, #infraccion)**

Clave: {#propietario,#infraccion}

Dependencias multivaluadas:  
 dm1: #propietario ->> #infraccion

-R6 cumple 4FN? Si, porque existe dm1, la cual es trivial.

Esquemas es 4FN:  
**R1(#auto, modeloAuto)**  
**R3(#cedula,#auto,#conductor, fechaVto)**  
**R5(#infraccion,#cedula,fechaInfraccion, tipoInfraccion)  
R6(#propietario, #infraccion)**

10 - RESERVA (#Reserva, #Agencia, nombreAgencia, fechaReservaVuelo, ciudadOrigen, ciudadDestino,tipoPago, nombreAerolínea, #Vuelo, dniPasajero, nombrePasajero, dirPasajero, telPasajero, clase, fechaPartida,fechaLlegada, horaPartida, horaLlegada, modeloAvión, #Asiento, tipoComida, compañíaPasajero, dirCompañía,telCompañía)

Dependencias funcionales.  
 df1: #Vuelo,nombreAerolinea -> modeloAvion, ciudadOrigen, ciudad Destino, horaPartida,horaLlegada  
 df2: #Reserva,nombreAerolinea -> #Agencia  
 df3: #Agencia-> nombreAgencia  
 df4: #Reserva,nombreAerolinea, dniPasajero -> dirPasajero  
 df5: dniPasajero -> nombrePasajero, telPasajero  
 df6: #Reserva, #Agencia, nombreAerolinea -> clase,fechaReservaVuelo, fechaPartida, fechaLlegada  
 df7: #Reserva, #Agencia, nombreAerolinea, dniPasajero -> tipoPago, #asiento

Cc:{#Vuelo,nombreAerolinea,#Reserva,dniPasajero,tipoComida, compañíaPasajero, dirCompañía,telCompañía}  
  
-RESERVA esta en BCNF? No, porque existen df1 a df7, donde por ej #Vuelo,nombreAerolinea no es superclave. Particionamos usando df1.

**R1(nombreAerolínea, #Vuelo,horaPartida, horaLlegada, modeloAvión,ciudadOrigen, ciudadDestino)**

**R2(#Reserva, #Agencia, nombreAgencia, fechaReservaVuelo,tipoPago, nombreAerolínea, #Vuelo, dniPasajero, nombrePasajero, dirPasajero, telPasajero, clase, fechaPartida,fechaLlegada, #Asiento, tipoComida, compañíaPasajero, dirCompañía,telCompañía)**

-Se perdio información? No, porque R1∩R2 = {nombreAerolinea,#Vuelo}, la cual es superclave en R1.  
-Se perdieron dependencias funcionales? No, porque vale df1 en R1 y valen de df2 a df7 en R2.  
-R1 esta en BCNF? Si, porque vale df1, donde {nombreAerolinea,#Vuelo} es superclave.  
-R2 esta en BCNF? No, porque vale df2 a df7, donde por ej #Agencia no es superclave. Particionamos usando df3.

**R3(#Agencia, nombreAgencia)**

**R4(#Reserva, #Agencia, fechaReservaVuelo,tipoPago, nombreAerolínea, #Vuelo, dniPasajero, nombrePasajero, dirPasajero, telPasajero, clase, fechaPartida,fechaLlegada, #Asiento, tipoComida, compañíaPasajero, dirCompañía,telCompañía)**

-Se perdio información? No, porque R3∩R4 = {#Agencia}, la cual es superclave en R3.  
-Se perdieron dependencias funcionales? No, porque vale df3 en R3 y valen df2 y de df4 a df7 en R4.  
-R3 esta en BCNF? Si, porque vale df3, donde {#Agencia} es superclave.  
-R4 esta en BCNF? No, porque vale df2 y de df4 a df7, donde por ej #Reserva,nombreAerolinea, dniPasajero no es superclave. Particionamos usando df4.

**R5(#Reserva,nombreAerolínea,dniPasajero,dirPasajero)**

**R6(#Reserva, #Agencia,fechaReservaVuelo,tipoPago, nombreAerolínea, #Vuelo, dniPasajero, nombrePasajero, telPasajero, clase, fechaPartida,fechaLlegada, #Asiento, tipoComida, compañíaPasajero, dirCompañía,telCompañía)**

-Se perdio información? No, porque R5∩R6 = {#Reserva,nombreAerolinea, dniPasajero}, la cual es superclave en R5.  
-Se perdieron dependencias funcionales? No, porque vale df4 en R5 y valen df2 y de df5 a df7 en R4.  
-R5 esta en BCNF? Si, porque vale df4, donde {#Reserva,nombreAerolinea, dniPasajero} es superclave.  
-R6 esta en BCNF? No, porque vale df2 y de df5 a df7, donde por ej dniPasajero no es superclave. Particionamos usando df5.

**R7(dniPasajero,nombrePasajero, telPasajero)**

**R8(#Reserva, #Agencia,fechaReservaVuelo,tipoPago, nombreAerolínea, #Vuelo,dniPasajero, clase, fechaPartida,fechaLlegada, #Asiento, tipoComida, compañíaPasajero, dirCompañía,telCompañía)**

-Se perdio información? No, porque R7∩R8 = {dniPasajero}, la cual es superclave en R7.  
-Se perdieron dependencias funcionales? No, porque vale df5 en R7 y valen df2, df6 y df7 en R8.  
-R7 esta en BCNF? Si, porque vale df5, donde {dniPasajero} es superclave.  
-R8 esta en BCNF? No, porque vale df2, df6 y df7, donde por ej #Reserva, #Agencia, nombreAerolinea no es superclave. Particionamos usando df6.

**R9(#Reserva, #Agencia,nombreAerolínea,fechaReservaVuelo,clase, fechaPartida,fechaLlegada)**

**R10(#Reserva, #Agencia,tipoPago, nombreAerolínea, #Vuelo,dniPasajero, #Asiento, tipoComida, compañíaPasajero, dirCompañía,telCompañía)**

-Se perdio información? No, porque R9∩R10 = {#Reserva, #Agencia, nombreAerolinea}, la cual es superclave en R9.  
-Se perdieron dependencias funcionales? No, porque vale df6 en R9 y valen df2 y df7 en R10.  
-R9 esta en BCNF? Si, porque vale df6, donde {#Reserva, #Agencia, nombreAerolinea} es superclave.  
-R10 esta en BCNF? No, porque vale df2 y df7, donde por ej #Reserva, #Agencia, nombreAerolinea, dniPasajero no es superclave. Particionamos usando df7.

**R11(#Reserva,#Agencia,nombreAerolínea,dniPasajero,tipoPago,#Asiento)**

**R12(#Reserva, #Agencia, nombreAerolínea, #Vuelo,dniPasajero, tipoComida, compañíaPasajero, dirCompañía,telCompañía)**

-Se perdio información? No, porque R11∩R12 = {#Reserva, #Agencia, nombreAerolinea, dniPasajero}, la cual es superclave en R11.  
-Se perdieron dependencias funcionales? No, porque vale df7 en R9 y vale df2 en R12.  
-R11 esta en BCNF? Si, porque vale df7, donde {#Reserva, #Agencia, nombreAerolinea,dniPasajero} es superclave.  
-R12 esta en BCNF? No, porque vale df2, donde {#Reserva, nombreAerolinea} no es superclave. Particionamos usando df2.

**R13(#Reserva,nombreAerolínea, #Agencia)**

**R14(#Reserva, nombreAerolínea, #Vuelo,dniPasajero, tipoComida, compañíaPasajero, dirCompañía,telCompañía)**

-Se perdio información? No, porque R13∩R14 = {#Reserva, nombreAerolinea}, la cual es superclave en R13.  
-Se perdieron dependencias funcionales? No, porque vale df2 en R13.  
-R13 esta en BCNF? Si, porque vale df2, donde {#Reserva,nombreAerolinea} es superclave.  
-R14 esta en BCNF? Si, porque las unicas dependencias que hay son triviales.

Esquemas que quedaron en BCNF:  
**R1(nombreAerolínea, #Vuelo,horaPartida, horaLlegada, modeloAvión,ciudadOrigen, ciudadDestino)  
R3(#Agencia, nombreAgencia)  
R5(#Reserva,nombreAerolínea,dniPasajero,dirPasajero)  
R7(dniPasajero,nombrePasajero, telPasajero)  
R9(#Reserva, #Agencia,nombreAerolínea,fechaReservaVuelo,clase, fechaPartida,fechaLlegada)  
R11(#Reserva,#Agencia,nombreAerolínea,dniPasajero,tipoPago,#Asiento)  
R13(#Reserva,nombreAerolínea, #Agencia)  
R14(#Reserva, nombreAerolínea, #Vuelo,dniPasajero, tipoComida, compañíaPasajero, dirCompañía,telCompañía)**

Clave:(#Reserva, nombreAerolínea, #Vuelo,dniPasajero, tipoComida, compañíaPasajero, dirCompañía,telCompañía)

Dependencias multivaluadas:  
 dm1: #Reserva ->> dniPasajero  
 dm2: #Reserva ->> #Vuelo  
 dm3: dniPasajero ->> #Reserva  
 dm4: #Reserva ->> nombreAerolinea  
 dm5: #Vuelo ->> tipoComida  
 dm6: #Reserva, nombreAerolinea, dniPasajero ->> compañiaPasajero  
 dm7: compañiaPasajero ->> dirCompañia  
 dm8: dirCompañia ->> telCompañia

-R14 cumple 4FN? No, porque existe df8 y no es trivial. Particionamos usando dm8.

**R15(dirCompañía,telCompañía)**

**R16(#Reserva, nombreAerolínea, #Vuelo,dniPasajero, tipoComida, compañíaPasajero, dirCompañía)**

-R15 cumple 4FN? Si, porque existe df8 y es trivial.  
-R16 cumple 4FN? No, porque existe df7 y no es trivial. Particionamos usando dm7.

**R17(compañíaPasajero, dirCompañía)**

**R18(#Reserva, nombreAerolínea, #Vuelo,dniPasajero, tipoComida, compañíaPasajero)**

-R17 cumple 4FN? Si, porque existe df7 y es trivial.  
-R18 cumple 4FN? No, porque existe df6 y no es trivial. Particionamos usando dm6.

**R19(#Reserva, nombreAerolínea,dniPasajero,compañíaPasajero)**

**R20(#Reserva, nombreAerolínea, #Vuelo,dniPasajero, tipoComida)**

-R19 cumple 4FN? Si, porque existe df6 y es trivial.  
-R20 cumple 4FN? No, porque existe df5 y no es trivial. Particionamos usando dm5.

**R21(#Vuelo,tipoComida)**

**R22(#Reserva, nombreAerolínea, #Vuelo,dniPasajero)**

-R21 cumple 4FN? Si, porque existe df5 y es trivial.  
-R22 cumple 4FN? No, porque existe df4 y no es trivial. Particionamos usando dm4.

**R23(#Reserva, nombreAerolínea)**

**R24(#Reserva, #Vuelo,dniPasajero)**

-R23 cumple 4FN? Si, porque existe df4 y es trivial.  
-R24 cumple 4FN? No, porque existe df2 y no es trivial. Particionamos usando dm2.

**R25(#Reserva, #Vuelo)**

**R26(#Reserva,dniPasajero)**

-R25 cumple 4FN? Si, porque existe df2 y es trivial.  
-R26 cumple 4FN? Si, porque existe df1 y df3 y son triviales.

Finalmente, los esquemas en 4FN son:  
**R1(nombreAerolínea, #Vuelo,horaPartida, horaLlegada, modeloAvión,ciudadOrigen, ciudadDestino)  
R3(#Agencia, nombreAgencia)  
R5(#Reserva,nombreAerolínea,dniPasajero,dirPasajero)  
R7(dniPasajero,nombrePasajero, telPasajero)  
R9(#Reserva, #Agencia,nombreAerolínea,fechaReservaVuelo,clase, fechaPartida,fechaLlegada)  
R11(#Reserva,#Agencia,nombreAerolínea,dniPasajero,tipoPago,#Asiento)  
R13(#Reserva,nombreAerolínea, #Agencia)  
R15(dirCompañía,telCompañía)  
R17(compañíaPasajero, dirCompañía)  
R19(#Reserva, nombreAerolínea,dniPasajero,compañíaPasajero)  
R21(#Vuelo,tipoComida)  
R23(#Reserva, nombreAerolínea)  
R25(#Reserva, #Vuelo)  
R26(#Reserva,dniPasajero)**

U∩  
11 - BUQUE (nombreBuque, nYApDueño, dniDueño, tipoBuque, tonelaje, tipoCasco, #Viaje, puertoOrigen,puertoDestino, puertoIntermedio, nomPaísPuertoDestino, nombrePaisPuertoOrigen, nombrePaisPuertoIntermedio, posicionActual, fechaPosicionActual, nYApPasajero, dniPasajero, dirPasajero,puertoInicioPasajero, puertoFinalPasajero)

Dependencias funcionales:  
 df1: nombreBuque -> tipoBuque  
 df2: tipoBuque -> tonelaje, tipoCasco  
 df3: nombreBuque, fechaPosicionActual -> posicionActual  
 df4: nombreBuque, #Viaje-> puertoOrigen, puertoDestino  
 df5: dniPasajero -> nYApPasajero, dirPasajero  
 df6: nombreBuque, #Viaje, dniPasajero -> puertoInicioPasajero, puertoFinalPasajero  
 df7: dniDueño -> nYApDueño  
 df8: puertoOrigen -> nomPaisPuertoOrigen  
 df9: puertoDestino -> nomPaisPuertoDestino  
 df10: puertoIntermedio -> nomPaisPuertoIntermedio

Cc: {nombreBuque,fechaPosicionActual,#Viaje,dniPasajero,dniDueño,puertoIntermedio}

-BUQUE cumple BCNF? No, porque existen df1 a df10, donde por ej {tipoBuque} no es superclave. Particionamos usando df2.

**R1(tipoBuque, tonelaje, tipoCasco)**

**R2(nombreBuque, nYApDueño, dniDueño, tipoBuque, #Viaje, puertoOrigen,puertoDestino, puertoIntermedio, nomPaísPuertoDestino, nombrePaisPuertoOrigen, nombrePaisPuertoIntermedio, posicionActual, fechaPosicionActual, nYApPasajero, dniPasajero, dirPasajero,puertoInicioPasajero, puertoFinalPasajero)**

-Se perdio información? No, porque R1∩R2 = {tipoBuque}, la cual es superclave de R1  
-Se perdieron dependencias funcionales? No, porque vale df2 en R1 y vale df1 y df3 a df10 en R2.  
-R1 cumple BCNF? Si, porque vale df1, donde {tipoBuque} es superclave.  
-R2 cumple BCNF? No, porque vale df1 y df3 a df10, donde por ej {nombreBuque,fechaPosicionActual} no es superclave. Particionamos usando df3.

**R3(nombreBuque, fechaPosicionActual,posicionActual)**

**R4(nombreBuque, nYApDueño, dniDueño, tipoBuque, #Viaje, puertoOrigen,puertoDestino, puertoIntermedio, nomPaísPuertoDestino, nombrePaisPuertoOrigen, nombrePaisPuertoIntermedio, fechaPosicionActual, nYApPasajero, dniPasajero, dirPasajero,puertoInicioPasajero, puertoFinalPasajero)**

-Se perdio información? No, porque R3∩R4 = {nombreBuque, fechaPosicionActual}, la cual es superclave de R3  
-Se perdieron dependencias funcionales? No, porque vale df3 en R3 y vale df1 y df4 a df10 en R4.  
-R3 cumple BCNF? Si, porque vale df3, donde {nombreBuque, fechaPosicionActual} es superclave.  
-R4 cumple BCNF? No, porque vale df1 y df4 a df10, donde por ej {puertoOrigen} no es superclave. Particionamos usando df8.

**R5(puertoOrigen,nombrePaisPuertoOrigen)**

**R6(nombreBuque, nYApDueño, dniDueño, tipoBuque, #Viaje, puertoOrigen,puertoDestino, puertoIntermedio, nomPaísPuertoDestino,nombrePaisPuertoIntermedio, fechaPosicionActual, nYApPasajero, dniPasajero, dirPasajero,puertoInicioPasajero, puertoFinalPasajero)**

-Se perdio información? No, porque R5∩R6 = {puertoOrigen}, la cual es superclave de R5.  
-Se perdieron dependencias funcionales? No, porque vale df8 en R5 y vale df1, df4 a df7, df9 y df10 en R6.  
-R5 cumple BCNF? Si, porque vale df8, donde {puertoOrigen} es superclave.  
-R6 cumple BCNF? No, porque vale df1, df4 a df7, df9 y df10, donde por ej {puertoDestino} no es superclave. Particionamos usando df9.

**R7(puertoDestino,nomPaísPuertoDestino)**

**R8(nombreBuque, nYApDueño, dniDueño, tipoBuque, #Viaje, puertoOrigen,puertoDestino, puertoIntermedio,nombrePaisPuertoIntermedio, fechaPosicionActual, nYApPasajero, dniPasajero, dirPasajero,puertoInicioPasajero, puertoFinalPasajero)**

-Se perdio información? No, porque R7∩R8 = {puertoDestino}, la cual es superclave de R7.  
-Se perdieron dependencias funcionales? No, porque vale df9 en R7 y vale df1, df4 a df7 y df10 en R8.  
-R7 cumple BCNF? Si, porque vale df9, donde {puertoDestino} es superclave.  
-R8 cumple BCNF? No, porque vale df1, df4 a df7 y df10, donde por ej {puertoIntermedio} no es superclave. Particionamos usando df10.

**R9(puertoIntermedio,nombrePaisPuertoIntermedio)**

**R10(nombreBuque, nYApDueño, dniDueño, tipoBuque, #Viaje, puertoOrigen,puertoDestino, puertoIntermedio, fechaPosicionActual, nYApPasajero, dniPasajero, dirPasajero,puertoInicioPasajero, puertoFinalPasajero)**

-Se perdio información? No, porque R9∩R10 = {puertoIntermedio}, la cual es superclave de R9.  
-Se perdieron dependencias funcionales? No, porque vale df10 en R9 y vale df1, df4 a df7 en R10.  
-R9 cumple BCNF? Si, porque vale df10, donde {puertoIntermedio} es superclave.  
-R10 cumple BCNF? No, porque vale df1, df4 a df7, donde por ej {nombreBuque} no es superclave. Particionamos usando df1.

**R11(nombreBuque, tipoBuque)**

**R12(nombreBuque, nYApDueño, dniDueño, #Viaje, puertoOrigen,puertoDestino, puertoIntermedio, fechaPosicionActual, nYApPasajero, dniPasajero, dirPasajero,puertoInicioPasajero, puertoFinalPasajero)**

-Se perdio información? No, porque R11∩R12 = {nombreBuque}, la cual es superclave de R11.  
-Se perdieron dependencias funcionales? No, porque vale df1 en R11 y vale df4 a df7 en R12.  
-R11 cumple BCNF? Si, porque vale df1, donde {nombreBuque} es superclave.  
-R12 cumple BCNF? No, porque vale df4 a df7, donde por ej {nombreBuque,#Viaje} no es superclave. Particionamos usando df4.

**R13(nombreBuque,#Viaje,puertoOrigen,puertoDestino)**

**R14(nombreBuque, nYApDueño, dniDueño, #Viaje, puertoIntermedio, fechaPosicionActual, nYApPasajero, dniPasajero, dirPasajero,puertoInicioPasajero, puertoFinalPasajero)**

-Se perdio información? No, porque R13∩R14 = {nombreBuque,#Viaje}, la cual es superclave de R13.  
-Se perdieron dependencias funcionales? No, porque vale df4 en R13 y vale df5 a df7 en R14.  
-R13 cumple BCNF? Si, porque vale df4, donde {nombreBuque,#Viaje} es superclave.  
-R14 cumple BCNF? No, porque vale df5 a df7, donde por ej {dniPasajero} no es superclave. Particionamos usando df5.

**R15(dniPasajero,nYApPasajero,dirPasajero)**

**R16(nombreBuque, nYApDueño, dniDueño, #Viaje, puertoIntermedio, fechaPosicionActual, dniPasajero,puertoInicioPasajero, puertoFinalPasajero)**

-Se perdio información? No, porque R15∩R16 = {dniPasajero}, la cual es superclave de R15.  
-Se perdieron dependencias funcionales? No, porque vale df5 en R15 y vale df6 y df7 en R16.  
-R15 cumple BCNF? Si, porque vale df5, donde {dniPasajero} es superclave.  
-R16 cumple BCNF? No, porque vale df6 y df7, donde por ej {nombreBuque, #Viaje,dniPasajero} no es superclave. Particionamos usando df6.

**R17(nombreBuque,#Viaje,dniPasajero,puertoInicioPasajero, puertoFinalPasajero)**

**R18(nombreBuque, nYApDueño, dniDueño, #Viaje, puertoIntermedio, fechaPosicionActual, dniPasajero)**

-Se perdio información? No, porque R17∩R18 = {nombreBuque,#Viaje,dniPasajero}, la cual es superclave de R17.  
-Se perdieron dependencias funcionales? No, porque vale df6 en R17 y vale df7 en R18.  
-R17 cumple BCNF? Si, porque vale df6, donde {nombreBuque,#Viaje,dniPasajero} es superclave.  
-R18 cumple BCNF? No, porque vale df7, donde {dniDueño} no es superclave. Particionamos usando df7.

**R19(dniDueño, nYApDueño)**

**R20(nombreBuque, dniDueño, #Viaje, puertoIntermedio, fechaPosicionActual, dniPasajero)**

-Se perdio información? No, porque R19∩R20 = {dniDueño}, la cual es superclave de R19.  
-Se perdieron dependencias funcionales? No, porque vale df7 en R19.  
-R19 cumple BCNF? Si, porque vale df7, donde {dniDueño} es superclave.  
-R20 cumple BCNF? Si, porque las unicas dependencias que hay son triviales.

Esquemas en BCNF:  
**R1(tipoBuque, tonelaje, tipoCasco)  
R3(nombreBuque, fechaPosicionActual,posicionActual)  
R5(puertoOrigen,nombrePaisPuertoOrigen)  
R7(puertoDestino,nomPaísPuertoDestino)  
R9(puertoIntermedio,nombrePaisPuertoIntermedio)  
R11(nombreBuque, tipoBuque)  
R13(nombreBuque,#Viaje,puertoOrigen,puertoDestino)  
R15(dniPasajero,nYApPasajero,dirPasajero)  
R17(nombreBuque,#Viaje,dniPasajero,puertoInicioPasajero, puertoFinalPasajero)  
R19(dniDueño, nYApDueño)  
R20(nombreBuque, dniDueño, #Viaje, puertoIntermedio, fechaPosicionActual, dniPasajero)**

Dependencias multivaluadas:  
 dm1: nombreBuque ->> dniDueño  
 dm2: nombreBuque,#Viaje ->> puntoIntermedio  
 dm3: nombreBuque,#Viaje ->> dniPasajero  
 dm4: nombreBuque ->> fechaPosicionActual

-R20 cumple 4FN? No, porque existe dm1 y no es trivial. Particionamos usando dm1

**R21(nombreBuque, dniDueño)**

**R22(nombreBuque, #Viaje, puertoIntermedio, fechaPosicionActual, dniPasajero)**

-R21 cumple 4FN? Si, porque existe dm1 y es trivial.  
-R22 cumple 4FN? No, porque existe dm2 y no es trivial. Particionamos usando dm2.

**R23(nombreBuque, #Viaje, puertoIntermedio)**

**R24(nombreBuque, #Viaje, fechaPosicionActual, dniPasajero)**

-R23 cumple 4FN? Si, porque existe dm2 y es trivial.  
-R24 cumple 4FN? No, porque existe dm3 y no es trivial. Particionamos usando dm3.

**R25(nombreBuque, #Viaje,dniPasajero)**

**R26(nombreBuque, #Viaje, fechaPosicionActual)**

-R25 cumple 4FN? Si, porque existe dm3 y es trivial.  
-R26 cumple 4FN? Si, porque existe dm4 y es trivial.

Finalmente, los esquemas en 4FN:  
**R1(tipoBuque, tonelaje, tipoCasco)  
R3(nombreBuque, fechaPosicionActual,posicionActual)  
R5(puertoOrigen,nombrePaisPuertoOrigen)  
R7(puertoDestino,nomPaísPuertoDestino)  
R9(puertoIntermedio,nombrePaisPuertoIntermedio)  
R11(nombreBuque, tipoBuque)  
R13(nombreBuque,#Viaje,puertoOrigen,puertoDestino)  
R15(dniPasajero,nYApPasajero,dirPasajero)  
R17(nombreBuque,#Viaje,dniPasajero,puertoInicioPasajero, puertoFinalPasajero)  
R19(dniDueño, nYApDueño)  
R21(nombreBuque, dniDueño)  
R23(nombreBuque, #Viaje, puertoIntermedio)  
R25(nombreBuque, #Viaje,dniPasajero)  
R26(nombreBuque, #Viaje, fechaPosicionActual)**