

Diseño

1_D_diseño_v2
rev.2

Autores:

Pedro Sánchez de Castro
Victor Cabezas Lucena
Alejandro Merlo Serrano
David Medina Godoy

Revisión	Fecha	Descripción
rev.1	19/03/12	Documento inicial de Diseño
rev.2	30/03/12	Revision diagrama de clases y de componentes

Índice

[1. Objetivos de diseño](#)

[Facilidad de uso](#)

[Flexible](#)

[Seguridad](#)

[Fiable](#)

[Extensible](#)

[Reutilizable](#)

[2. Descomposición en subsistemas](#)

[3. Diseño de objetos](#)

[Diagrama de componentes](#)

[Paquete Persona](#)

[Paquete Beneficiario](#)

[Paquete Trabajador](#)

[Paquete Donante](#)

[Diagrama Despliegue](#)

[4. Identificar nuevas clases](#)

[Diagrama de clases de diseño](#)

[5. Diagramas de secuencia](#)

[6. Asignación de las tareas](#)

[7. Control de cambios](#)

1. Objetivos de diseño

Facilidad de uso

1. **Utilizar el lenguaje de los usuarios.** El sistema debe hablar el lenguaje de los usuarios, con las palabras, las frases y los conceptos familiares, en lugar de que los términos estén orientados al sistema. Utilizar convenciones del mundo real, haciendo que la información aparezca en un orden natural y lógico.
2. **Control y libertad para el usuario.** Los usuarios eligen a veces funciones del sistema por error y necesitan a menudo una salida de emergencia claramente marcada, esto es, salir del estado indeseado sin tener que pasar por un diálogo extendido. Facilitar un botón "inicio".
3. **Minimizar la carga de la memoria del usuario.** El usuario no debería tener que recordar la información de una parte del diálogo a la otra. Es mejor mantener objetos, acciones, y las opciones visibles que memorizar.
4. **Los diálogos estéticos y diseño minimalista.** No deben contener la información que sea inaplicable o se necesite raramente. Cada unidad adicional de la información en un diálogo compite con las unidades relevantes de la información y disminuye su visibilidad relativa.
5. **Ayudar a los usuarios a reconocer, diagnosticar y recuperarse de los errores.** Que los mensajes de error se deben expresar en un lenguaje claro, se debe indicar exactamente el problema, y deben ser constructivos.
6. **Ayuda y documentación.** Aunque es mejor si el sistema se puede usar sin documentación, puede ser necesario disponer de ayuda y documentación. Ésta tiene que ser fácil de buscar, centrada en las tareas del usuario, tener información de las etapas a realizar y que no sea muy extensa.

Flexible

Se facilitará la incorporación de diferentes tecnologías para manejar la persistencia, es decir, permitir reaccionar al sistema en caso cambios previstos o no previstos,

Seguridad

Se debe garantizar la seguridad de los datos almacenados en el sistema, estos datos son de carácter personal y están amparados bajo la ley de protección de datos de carácter personal, hay que garantizar que los datos se guarden de forma segura y que no se permita el acceso a personas ajenas a la plataforma, es decir, que no sean trabajadores.

Se proporcionarán métodos para el acceso seguro mediante usuario y contraseña para cada trabajador.

En definitiva el sistema de ser capaz de denegar servicios y detectar amenazas

Fiable

Se espera que el sistema trabaje de una forma estable con un bajo tiempo de respuesta al usuario en todo momento. Además éste será capaz de gestionar errores graves o inesperados de forma satisfactoria (cortes de luz, fallo en el SO, interrupción de comunicaciones...).

Se debe garantizar que los datos se guarden de forma segura en cada sesión, evitando fugas de datos y advirtiéndolo si se van/pueden producirse.

Extensible

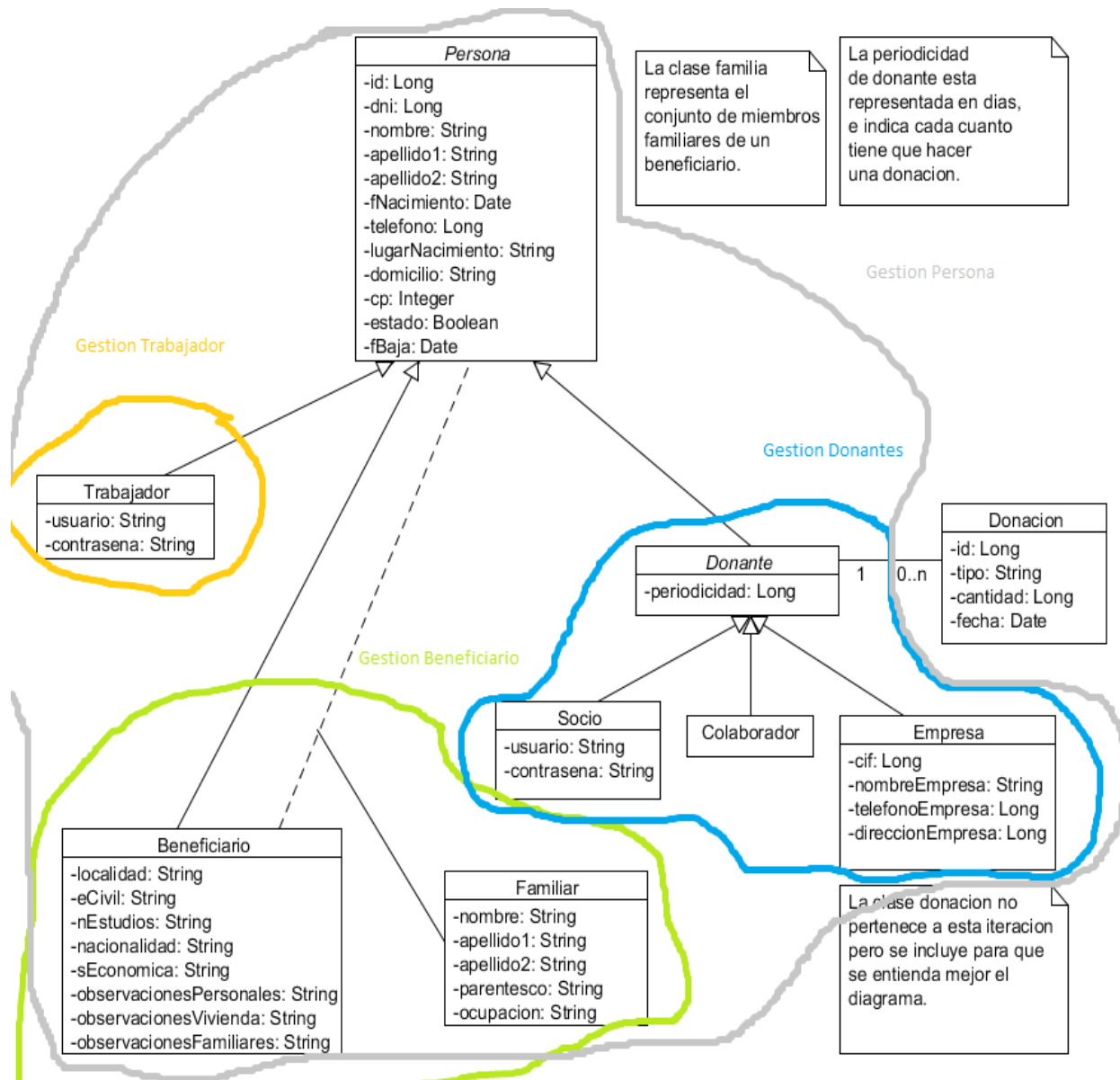
Permitirá dotar de nuevas funciones al sistema de forma cómoda.

Si el cliente quisiera que el sistema desarrolle nuevas funcionalidades el sistema debe permitirlo.

Reutilizable

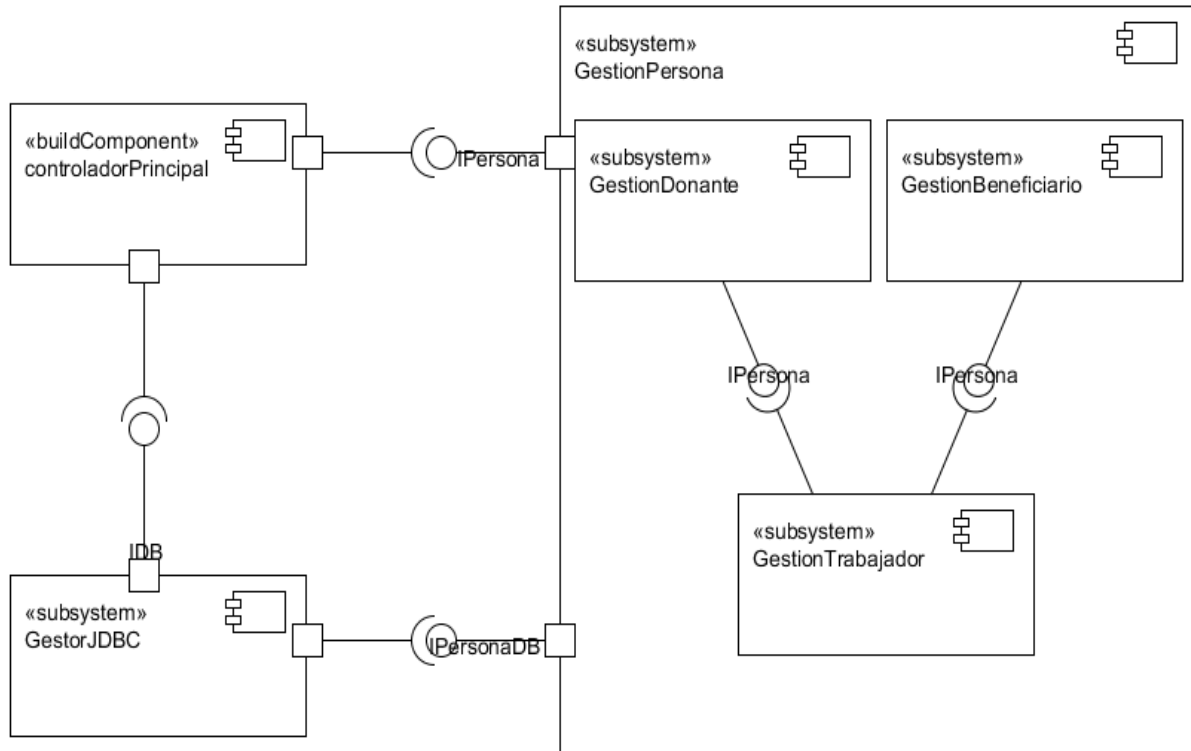
El sistema se diseñará de forma que pueda ser utilizado de nuevo para futuros proyectos que puedan estar relacionados con este.

2. Descomposición en subsistemas

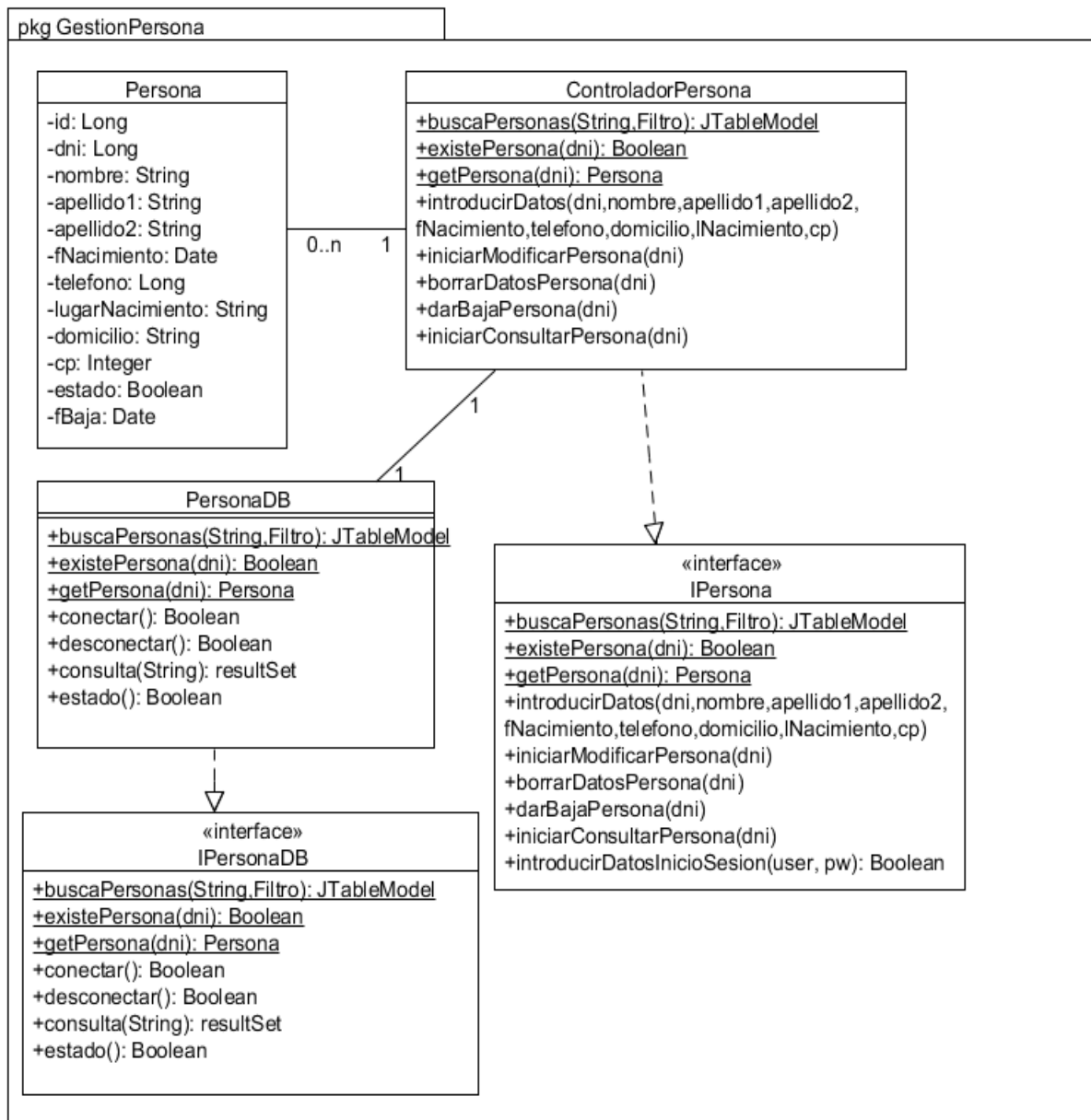


3. Diseño de objetos

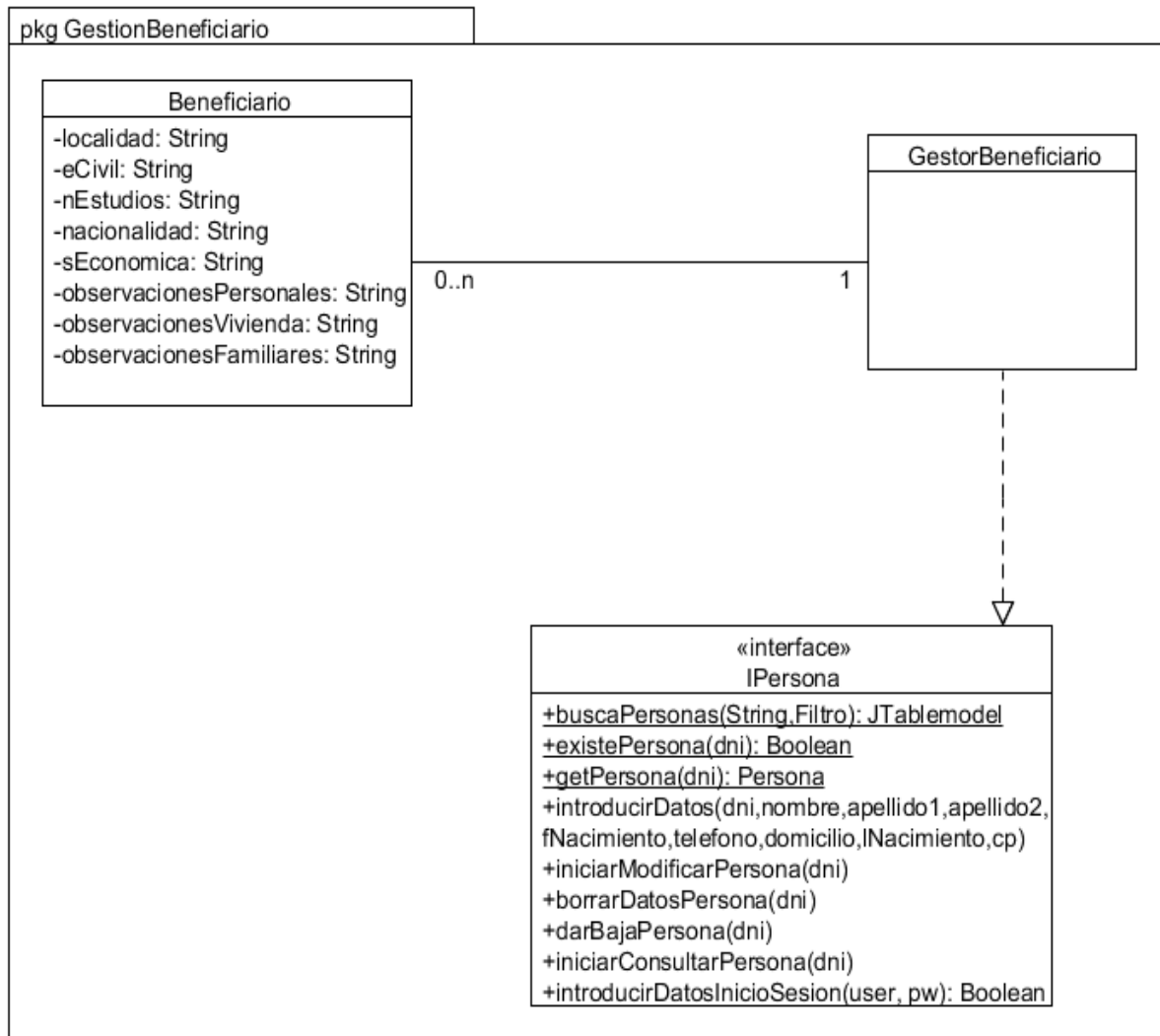
Diagrama de componentes



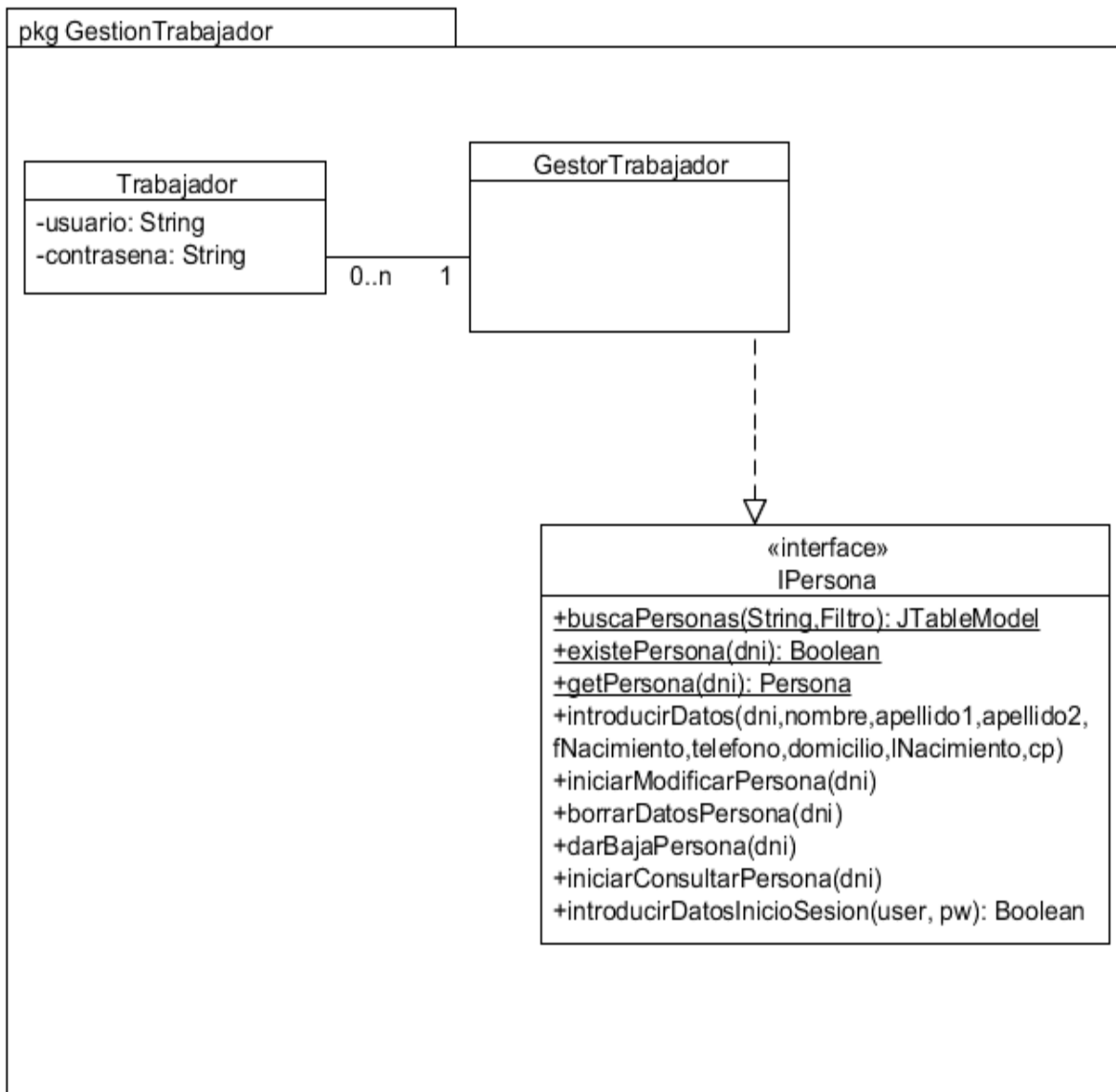
Paquete Persona



Paquete Beneficiario



Paquete Trabajador



Paquete Donante

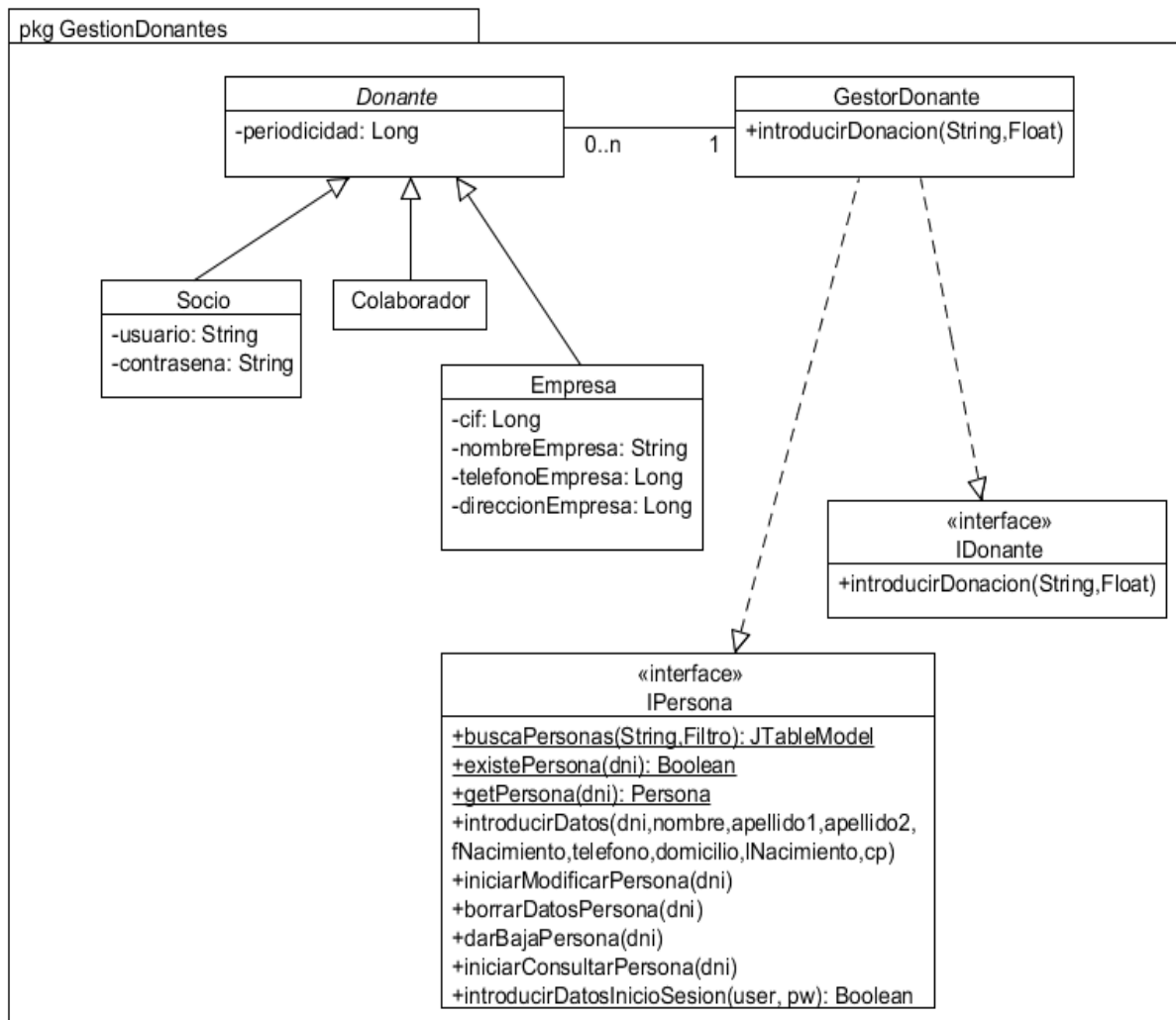
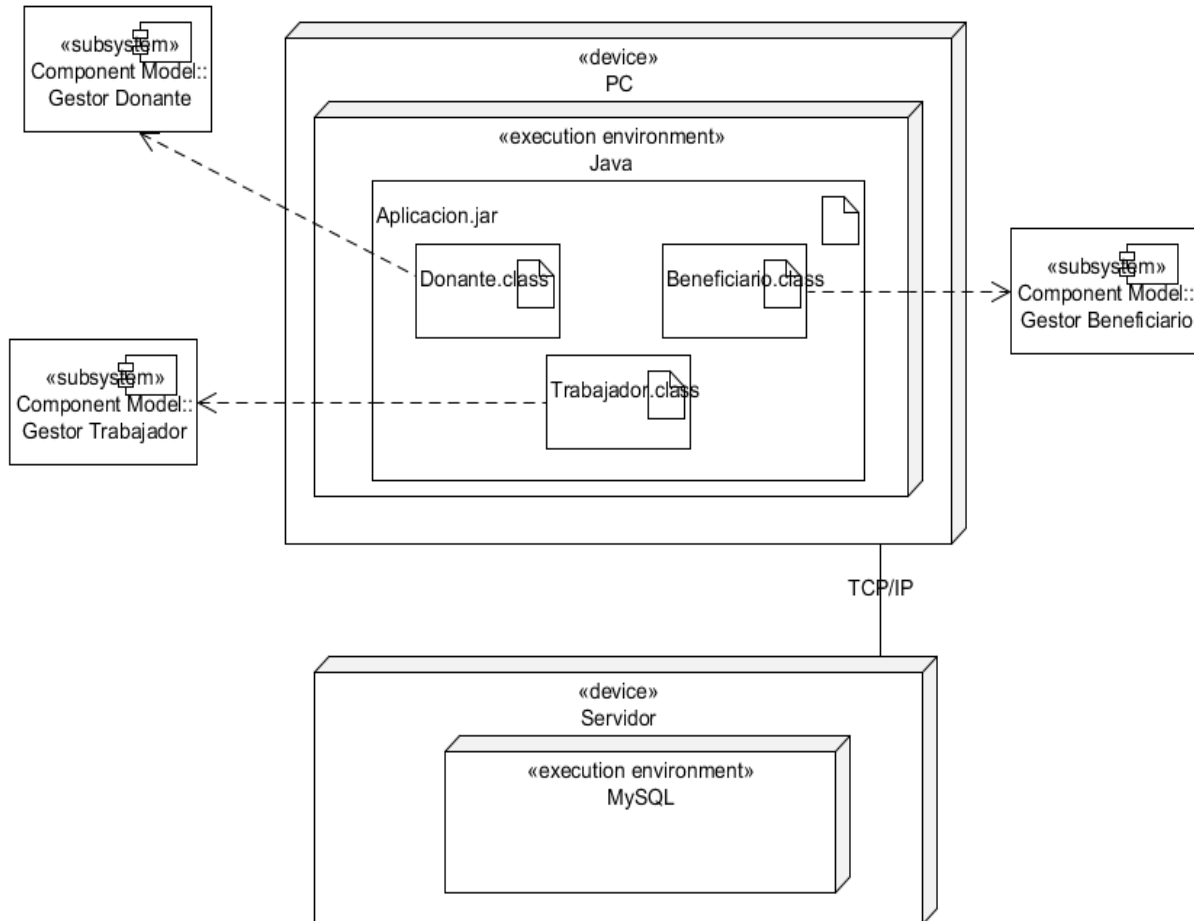


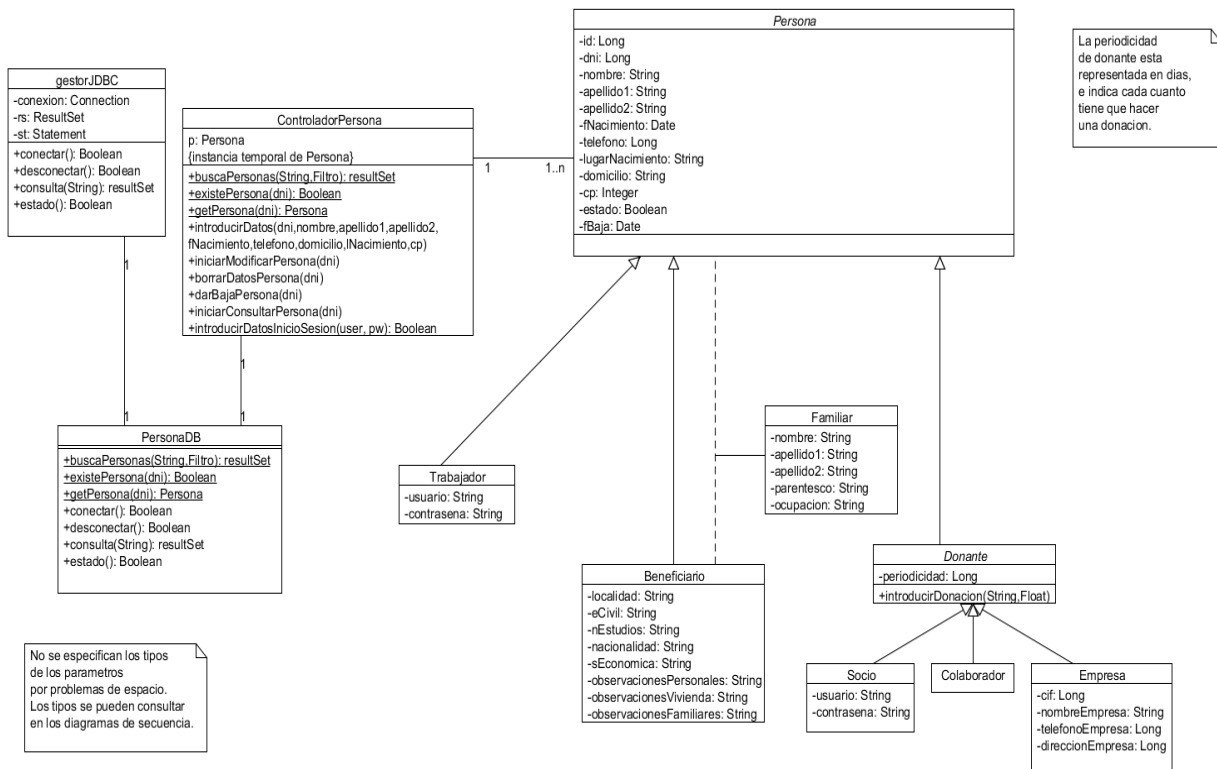
Diagrama Despliegue



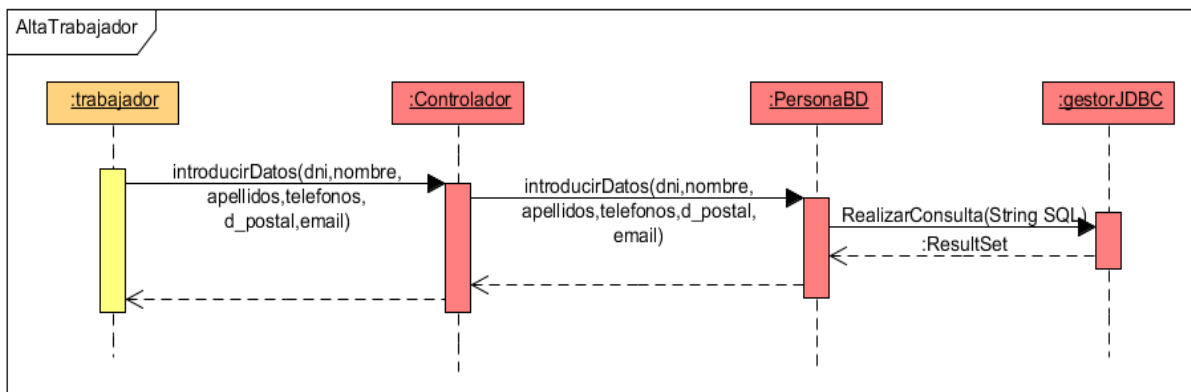
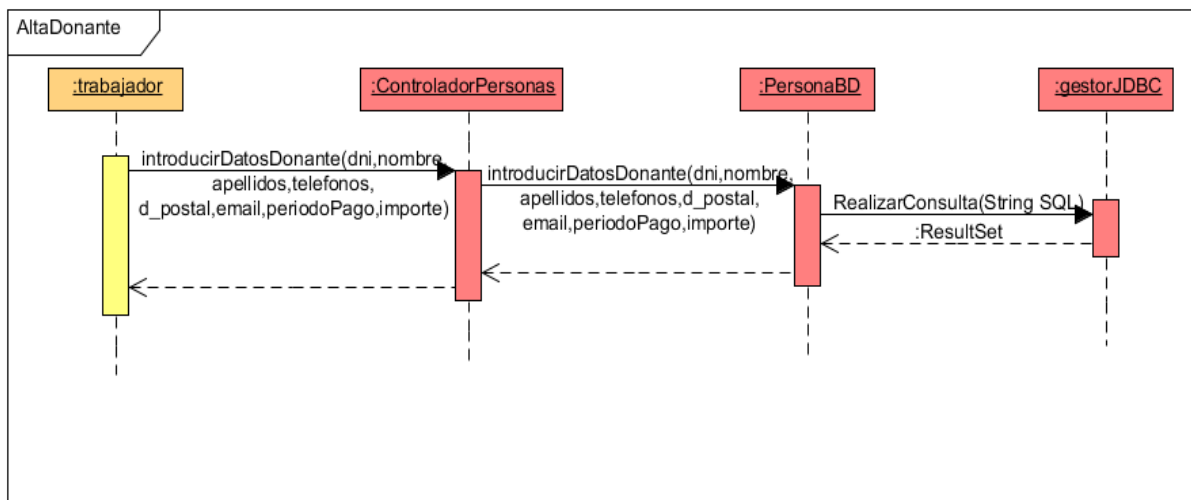
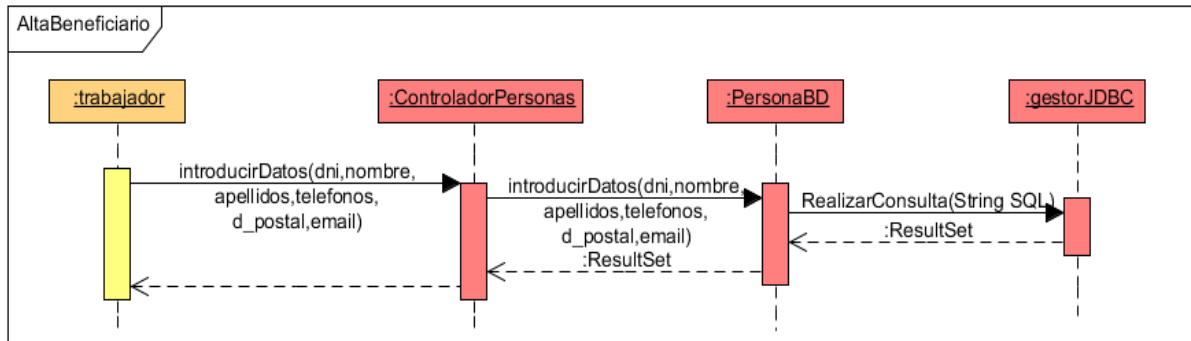
4. Identificar nuevas clases

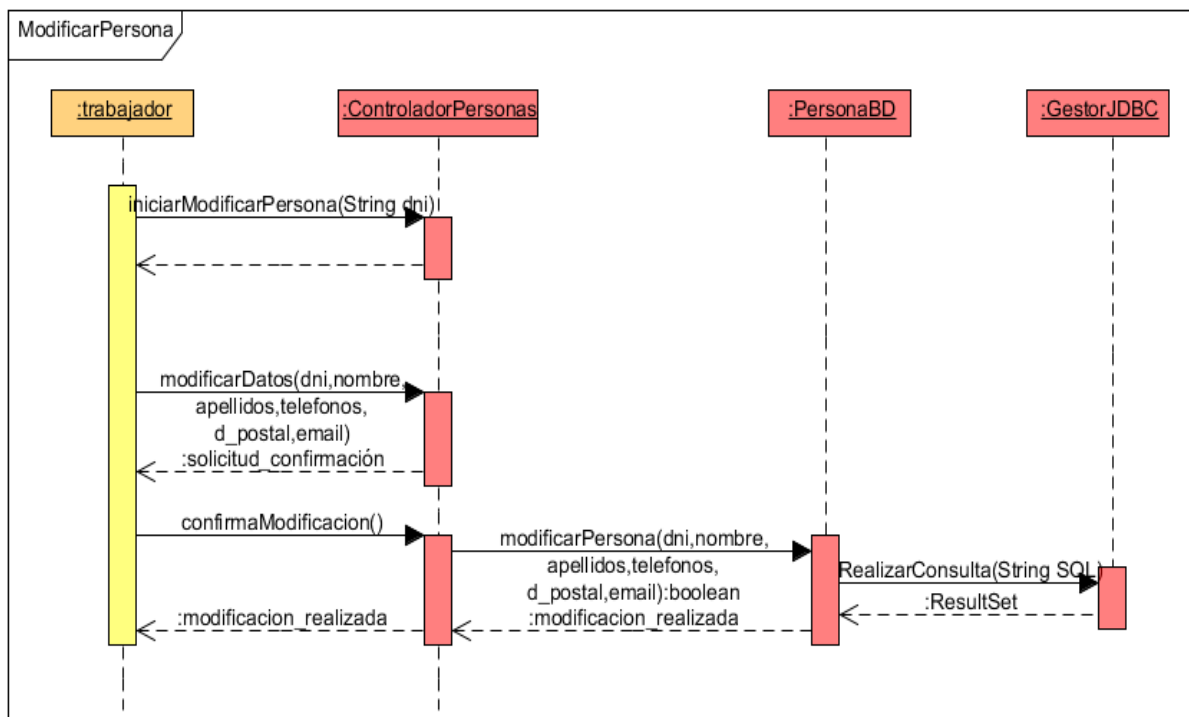
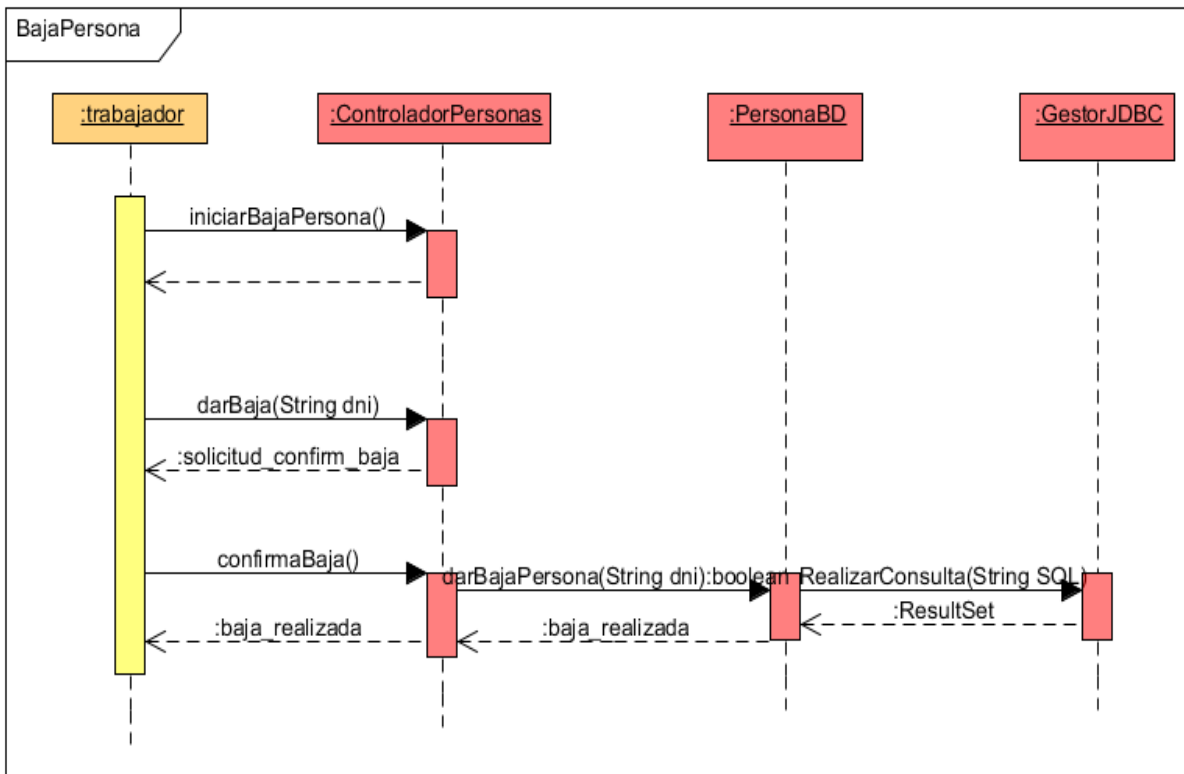
Las nuevas clases introducidas son GestorJDBC la cual gestiona la conexión con la base de datos, la clase PersonaDB, encargada de pasar los datos al mundo relacional, y la clase ControladorPersona encargada de la lógica de la aplicación y la comunicación de la interfaz de usuario con los datos.

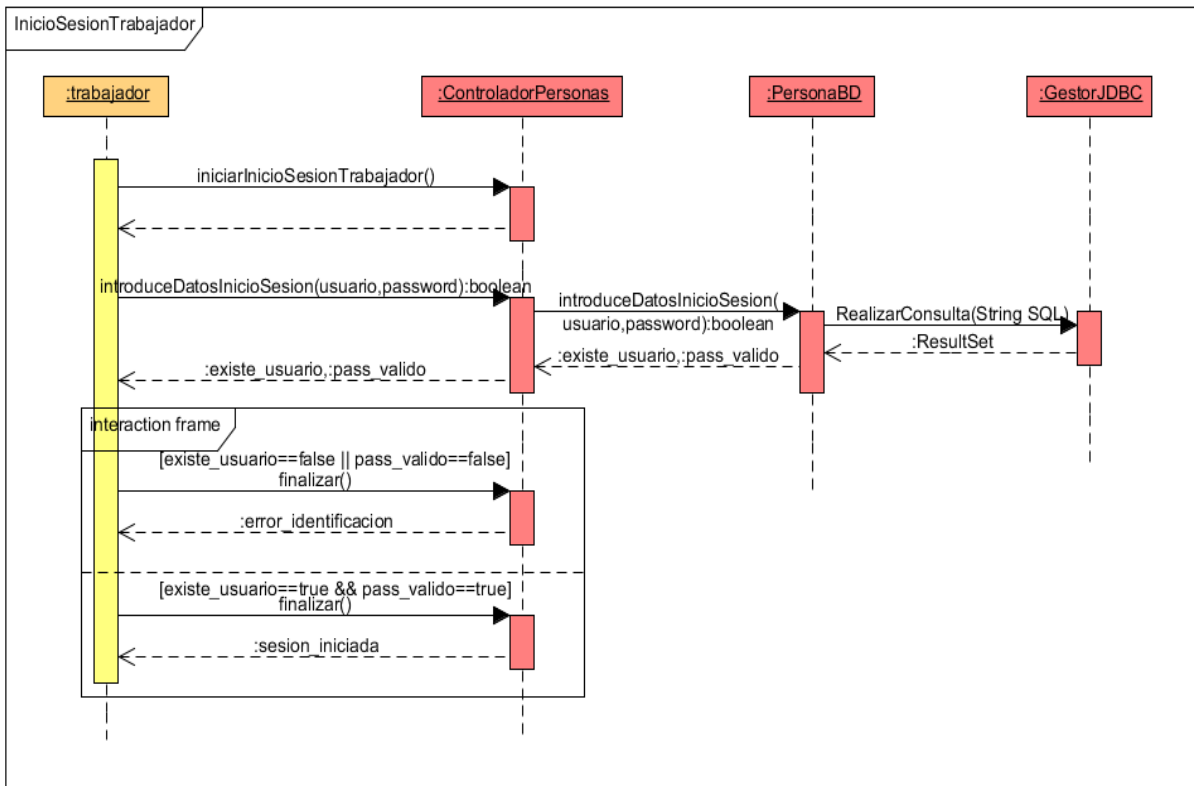
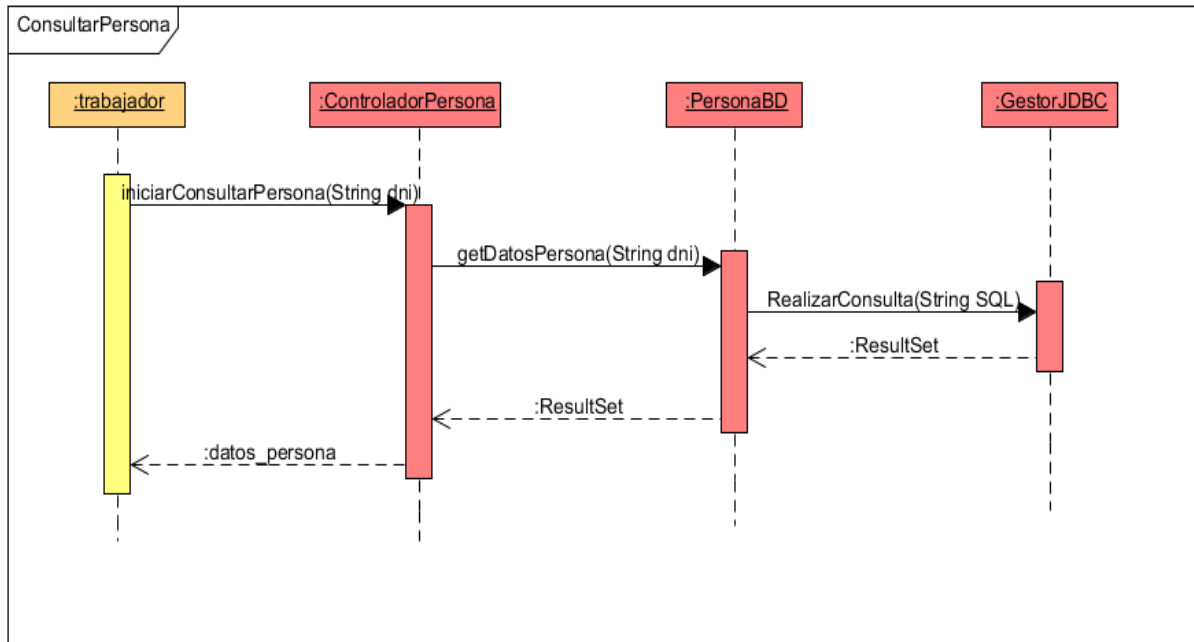
Diagrama de clases de diseño

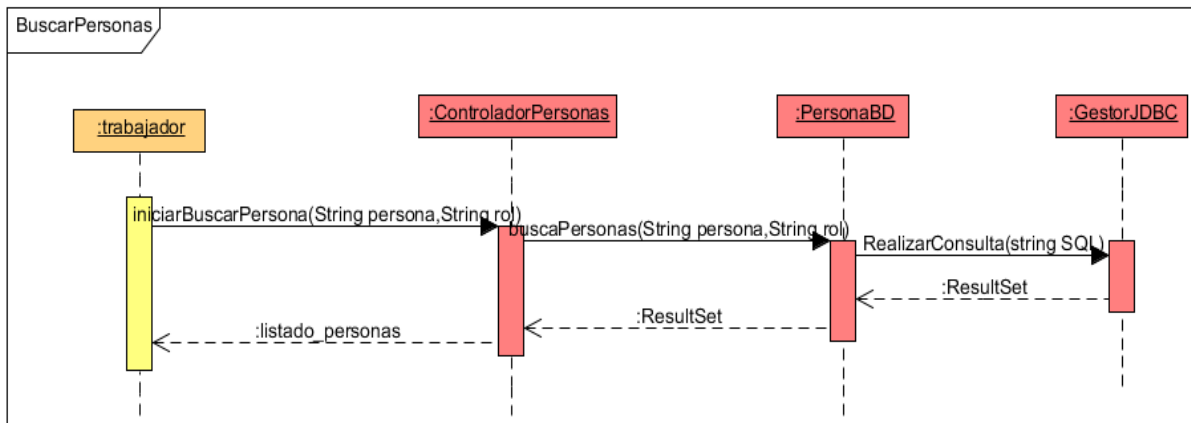
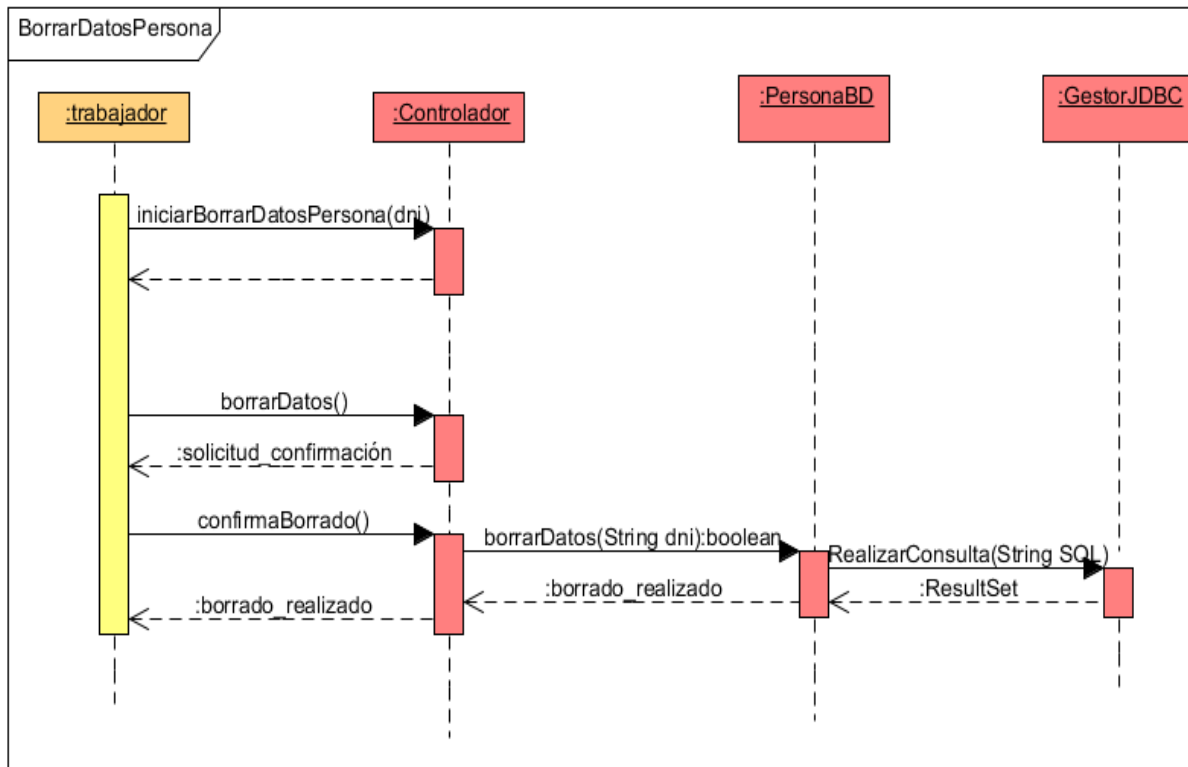


5. Diagramas de secuencia









6. Asignación de las tareas

Objetivos de diseño: Pedro Sanchez de Castro

Diagrama despliegue: David Medina Godoy y Víctor Cabezas Lucena

Diagrama de componentes: Víctor Cabezas Lucena

Diagrama clases de diseño: Víctor Cabezas Lucena y Alejandro Merlo

Diagramas de secuencia: David Medina Godoy

7. Control de cambios

Rev 2.	
Nombre de este documento	Análisis
Fecha	30/03/12
Documento afectado	1_D_diseño_v1
Breve descripción del problema	Encontradas inconsistencias con las clases “Familiar” y “Sesion”. Faltan los diagramas de secuencia de diseño.
Fecha de detección del problema	28/03/12
Impacto del problema sobre la planificación	Se ha retrasado una semana
Solución de cambio adoptada	Revisión del diagrama de clases. Inclusión diagramas de secuencia de diseño.
Anexos a este documento	