Universidad de Alcalá

Escuela Politécnica Superior

Universidad de Alcalá

PECL 3 – ARTEFACTO 8

Modelo de diseño

Ing. Software

Laboratorio Martes 12:00 – 14:00

Grado en Ingeniería Informática – Curso 2018/2019

Eduardo Graván Serrano – 03212337L

Marcos Barranquero Fernández – 51129104N

Sonia Rodríguez-Peral Bustos - 54302528B

Adrián Montesinos González – 51139629A

Alejandro Caballero Platas – 50891258D

INTRODUCCIÓN

Hemos diseñado cuatro casos de uso. Para la realización del Modelo de Diseño, primero hemos tenido que especificar aún más el conjunto de tecnologías que se usarán en la implementación.

Para la capa de persistencia, nos decantamos por emplear un software proyección objeto-relacional para implementar el modelo de dominio. Consideramos que la carga de trabajo del sistema no es lo suficientemente elevada como para que esto sea un problema de rendimiento y las mejoras de productividad compensan el trabajo de configuración. Para Java, el estándar de facto de la industria es Hibernate y alguno de nuestros ingenieros tiene experiencia con él, así que decidimos añadir dicha dependencia.

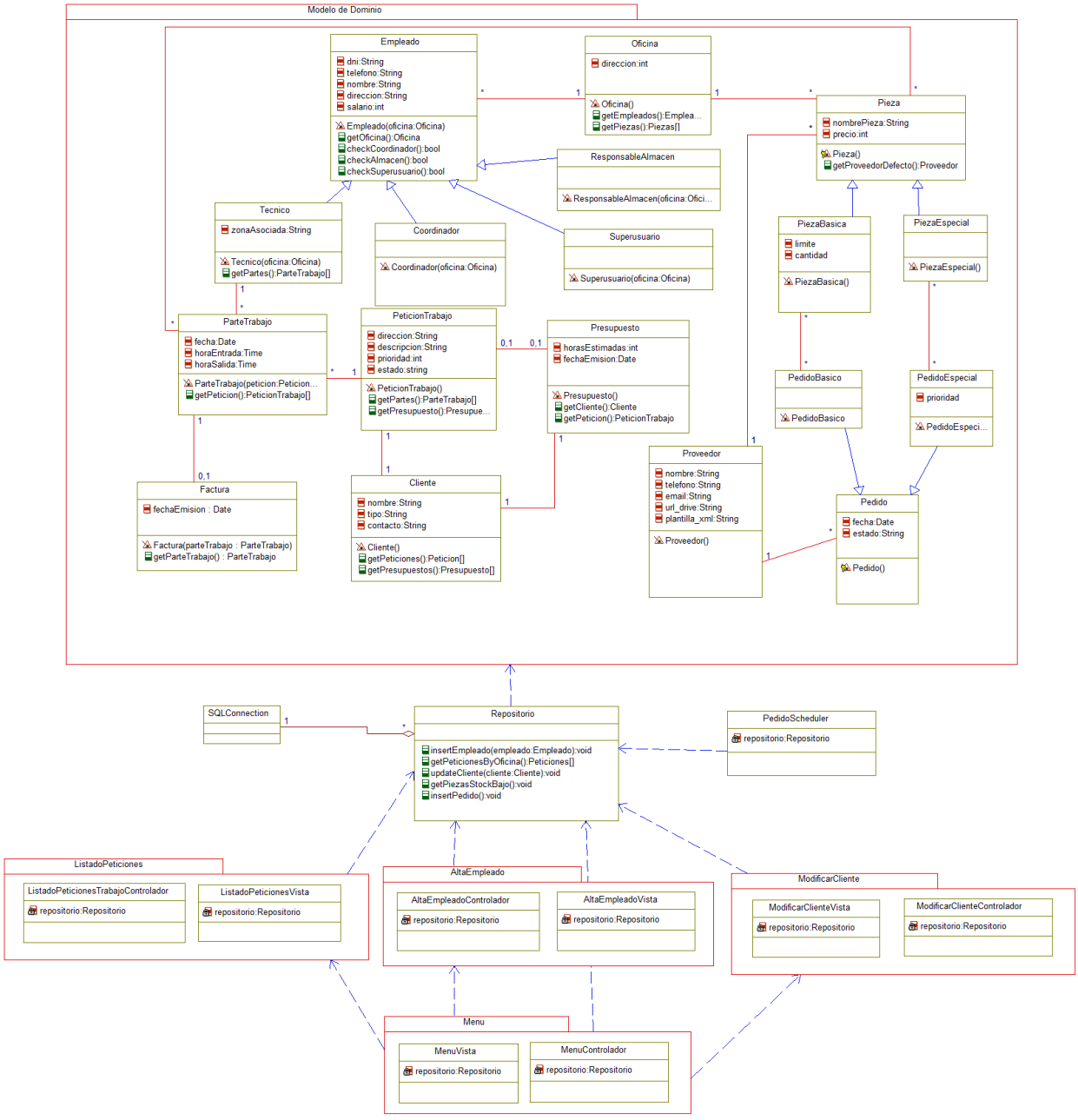
Para la interfaz, teníamos claro que queríamos dar al código de interfaz más estructura que en trabajos anteriores, así que para facilitar el trabajo en equipo decidimos emplear una estructura modelo-vista-controlador (MVC). El modelo ya lo teníamos delimitado mediante la capa de persistencia, así que solo nos quedaría implementar patrones de vista-controlador.

También por experiencia de alguno de nuestros ingenieros, tomamos una dependencia en el framework Spring: este nos da interfaces MVC listas y con herramientas extra. Además trae integración con Hibernate, que ya habíamos decidido emplear. Siendo más específicos, Spring nos permite implementar automáticamente instancias del patrón Repositorio para acceder más eficientemente a la proyección objeto-relacional. Este repositorio mantiene la conexión SQL, auto-implementa distintas consultas y centraliza la sincronización de la capa de persistencia.

Las vistas y los controladores emplean la inyección de dependencias que ofrece Spring para acceder al repositorio y al resto de instancias vista-controlador. Todo acceso y manipulación del modelo de dominio se hace mediante el repositorio.

DIAGRAMA DE CLASES

Aquí mostramos el diagrama de clases. Está compuesto por la proyección objeto-relacional, el repositorio, las parejas vista-controlador y el programador de pedidos. Nótese que solo hemos añadido al diagrama las parejas vista-controlador relevantes a los cuatro casos de uso diseñados, pero se corresponde una pareja vista-controlador con cada menú de la interfaz.



DIAGRAMAS DE COLABORACIÓN Y SECUENCIA

DIAGRAMA DE COLABORACIÓN - ALTA EMPLEADO (CU17)

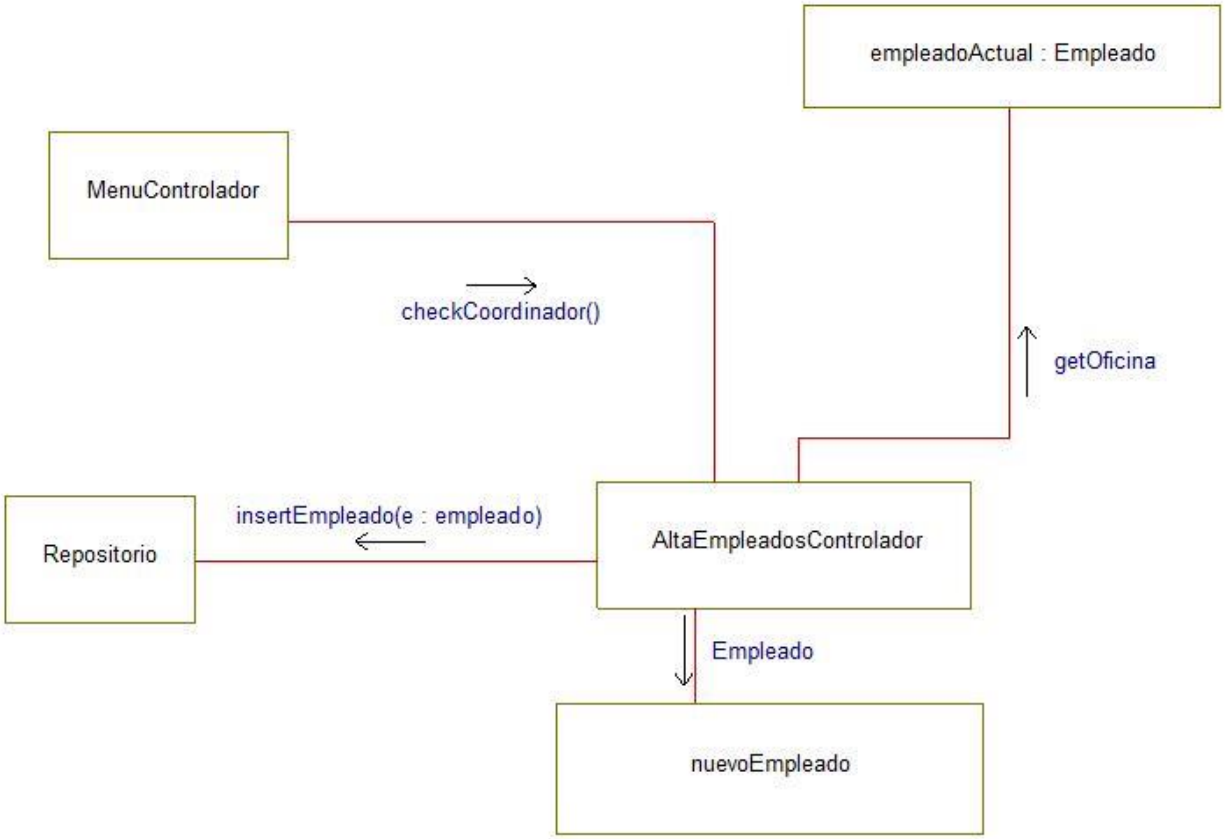


DIAGRAMA DE SECUENCIA - ALTA EMPLEADO (CU17)

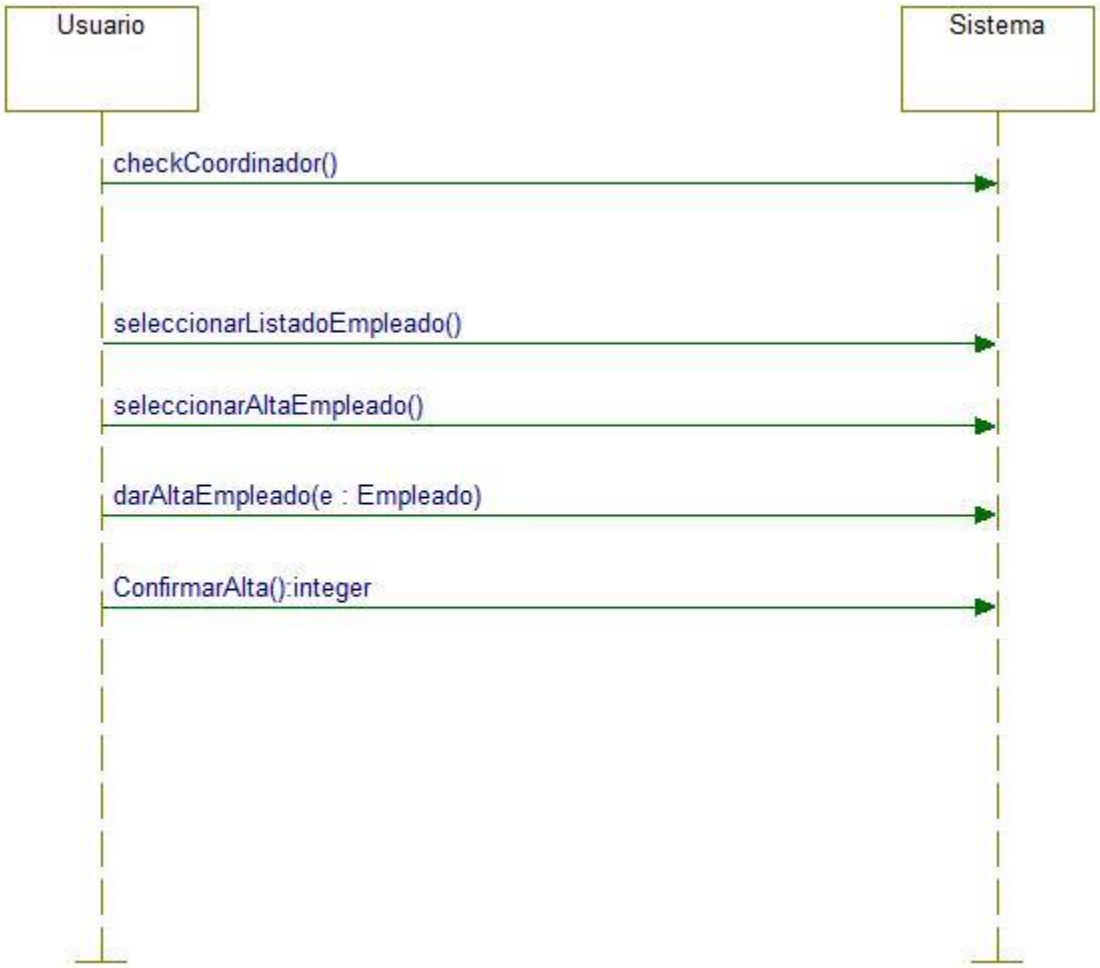


DIAGRAMA DE COLABORACIÓN - MODIFICAR CLIENTE (CU07)

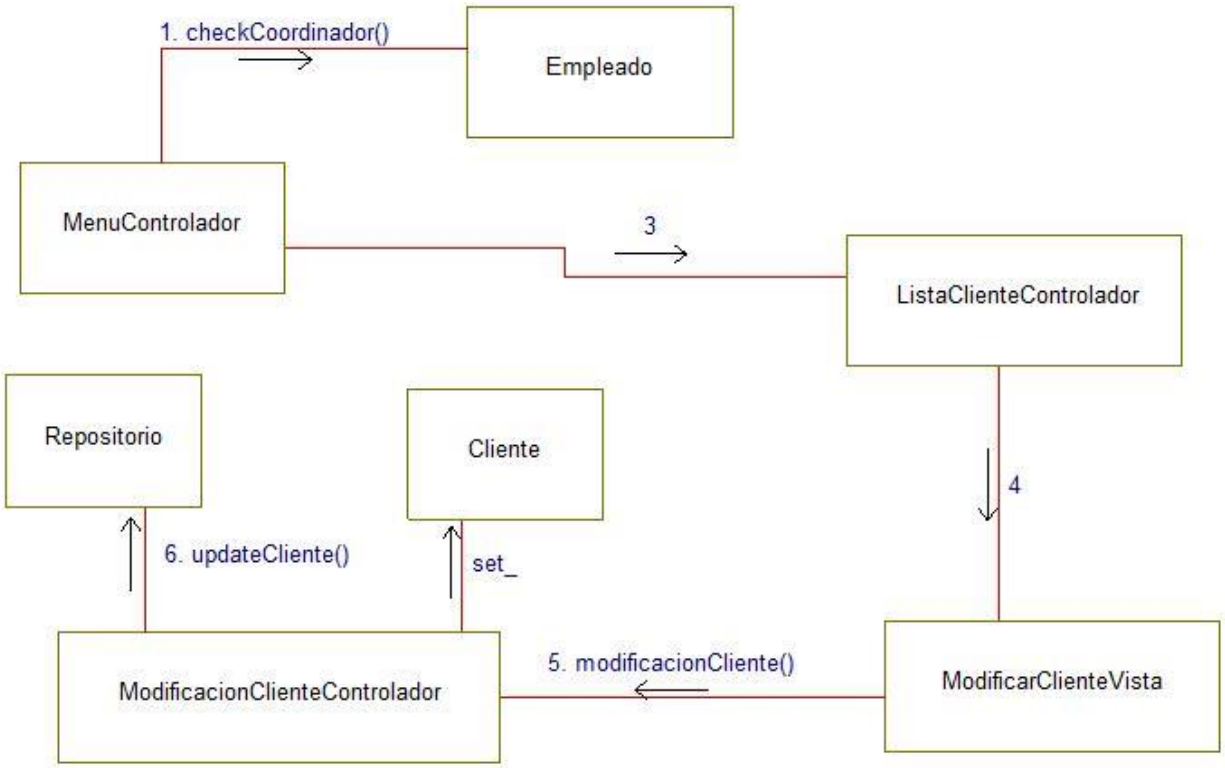


DIAGRAMA DE SECUENCIA - MODIFICAR CLIENTE (CU07)

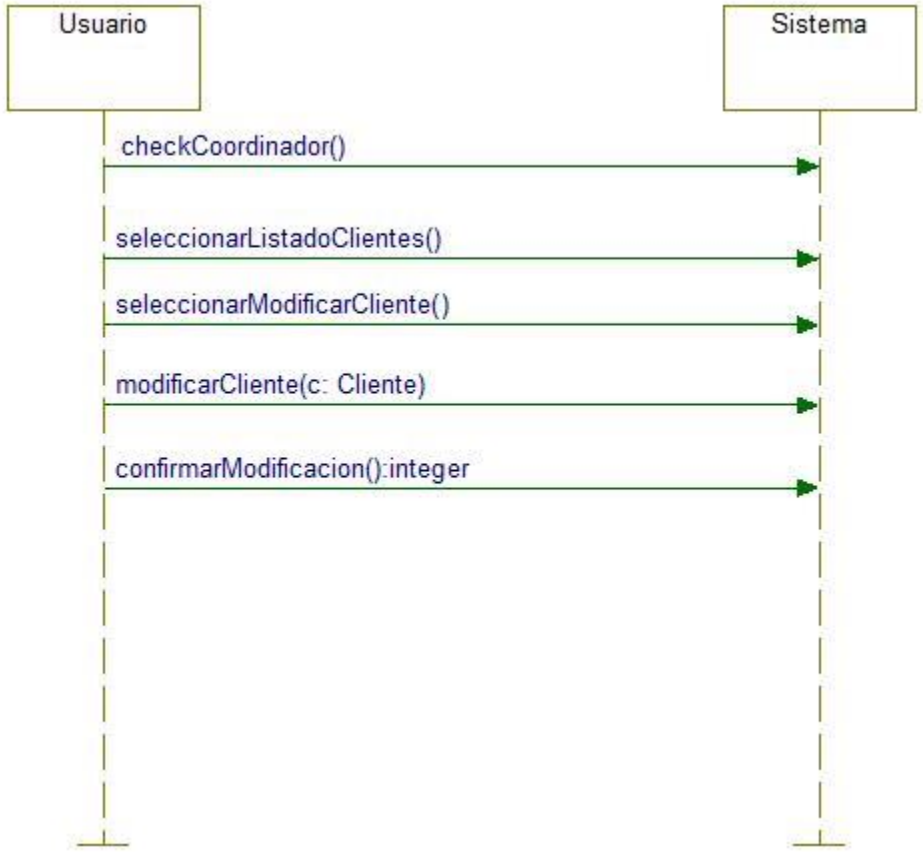


DIAGRAMA DE COLABORACIÓN - PEDIDO BÁSICO AUTOMÁTICO (CU25)

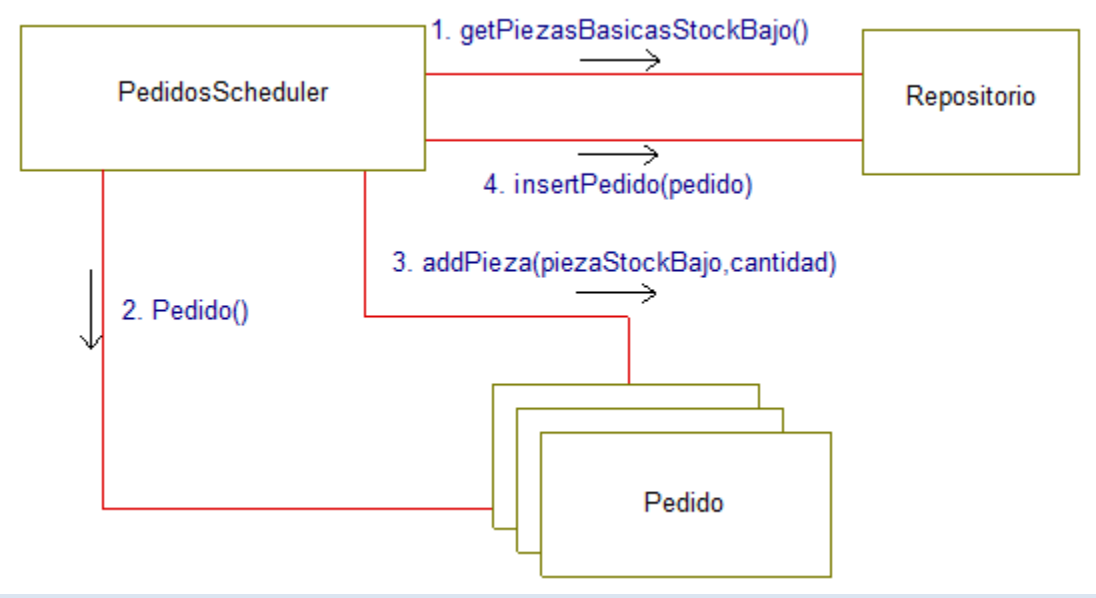


DIAGRAMA DE SECUENCIA - PEDIDO BÁSICO AUTOMÁTICO (CU25)

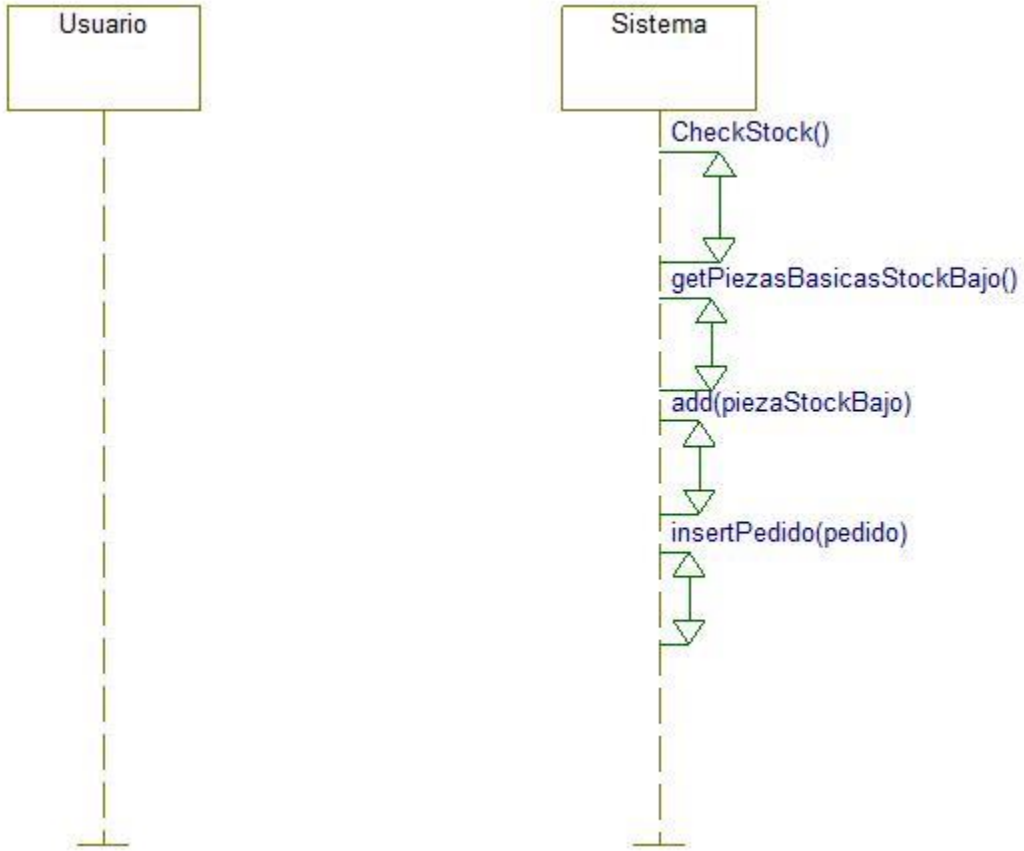


DIAGRAMA DE COLABORACIÓN - LISTAR PETICIONES DE TRABAJO (CU45)

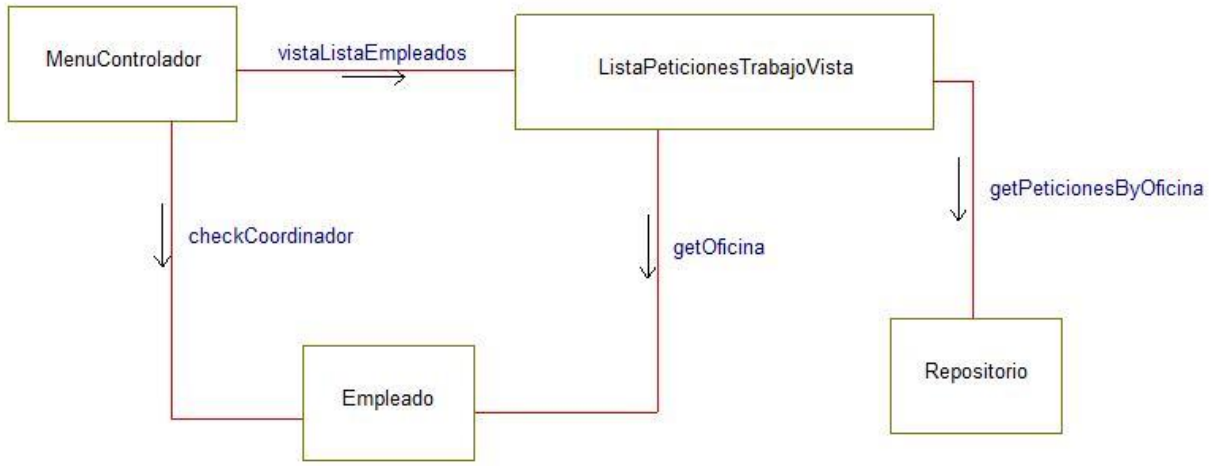


DIAGRAMA DE SECUENCIA - LISTAR PETICIONES DE TRABAJO (CU45)

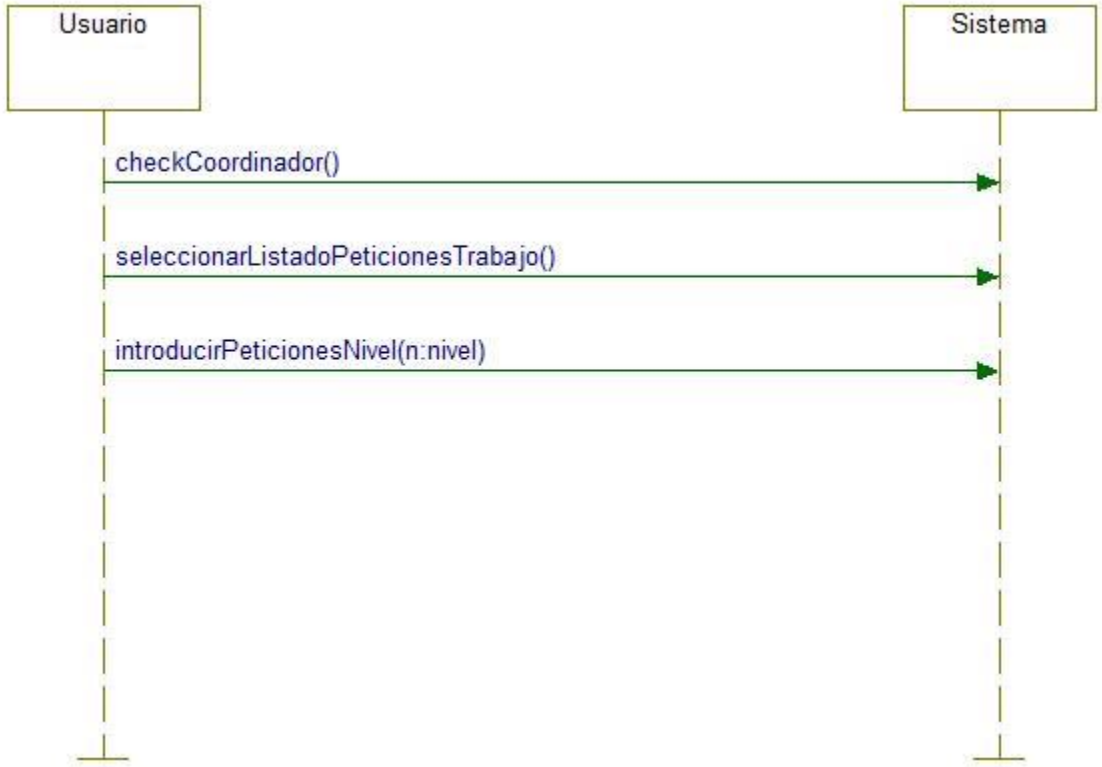


DIAGRAMA DE COLABORACIÓN - CAMBIAR TAMAÑO FUENTE (CU75)

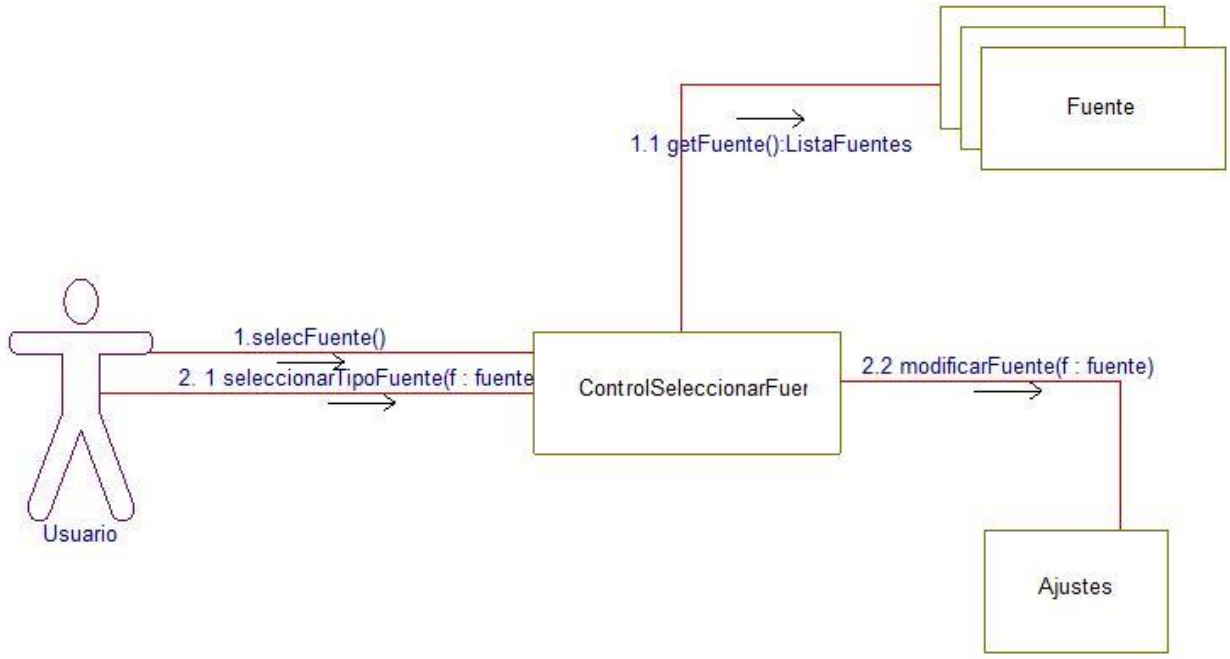


DIAGRAMA DE SECUENCIA - CAMBIAR TAMAÑO FUENTE (CU75)

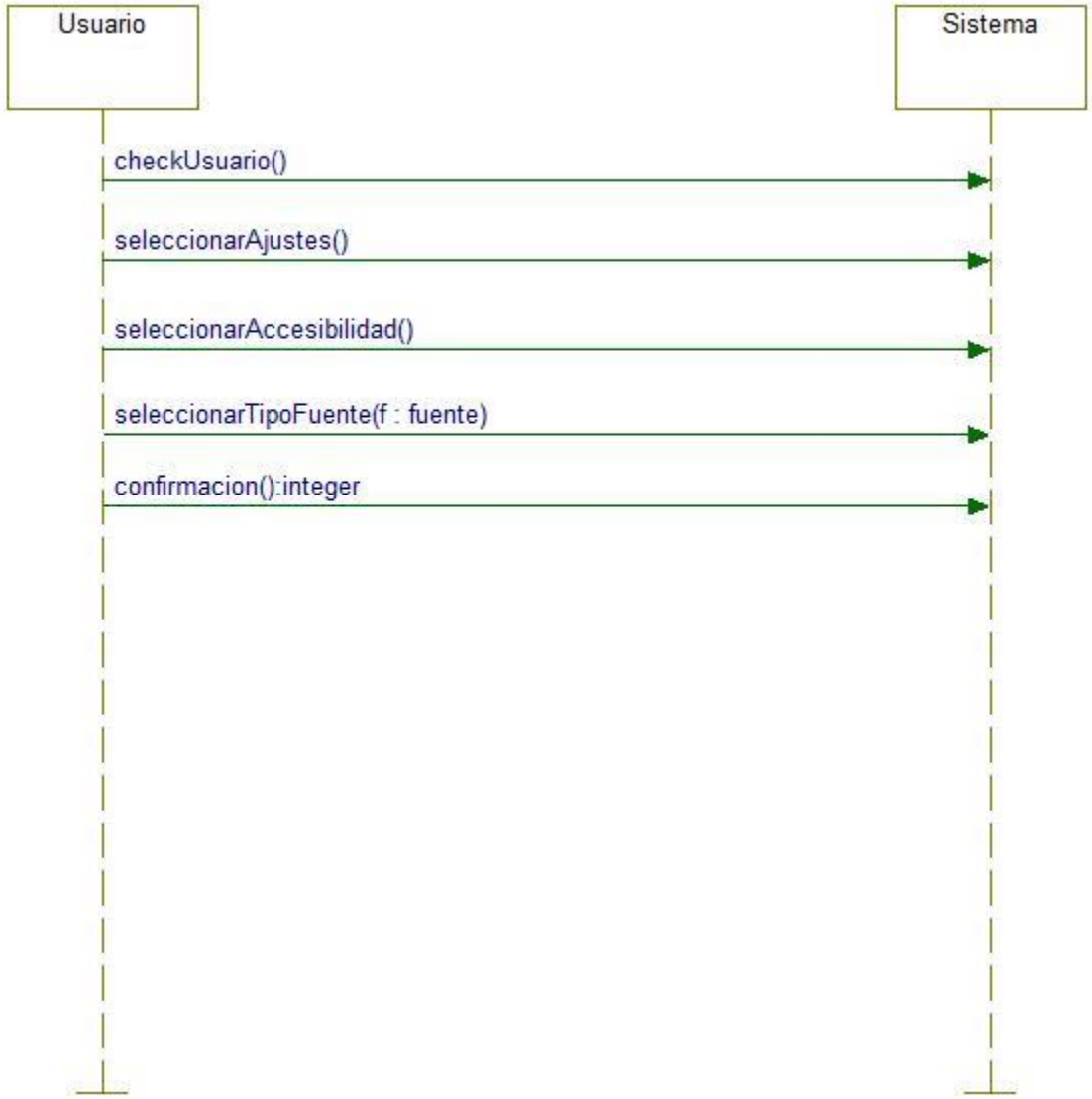


DIAGRAMA DE COLABORACIÓN - ALTA PIEZA BÁSICA (CU29)

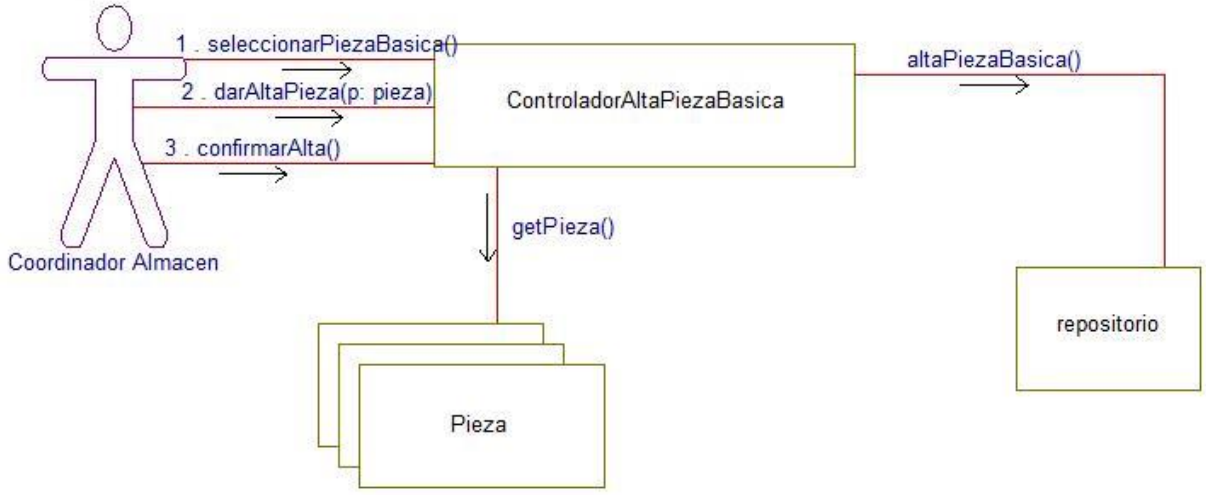


DIAGRAMA DE SECUENCIA - ALTA PIEZA BÁSICA (CU29)

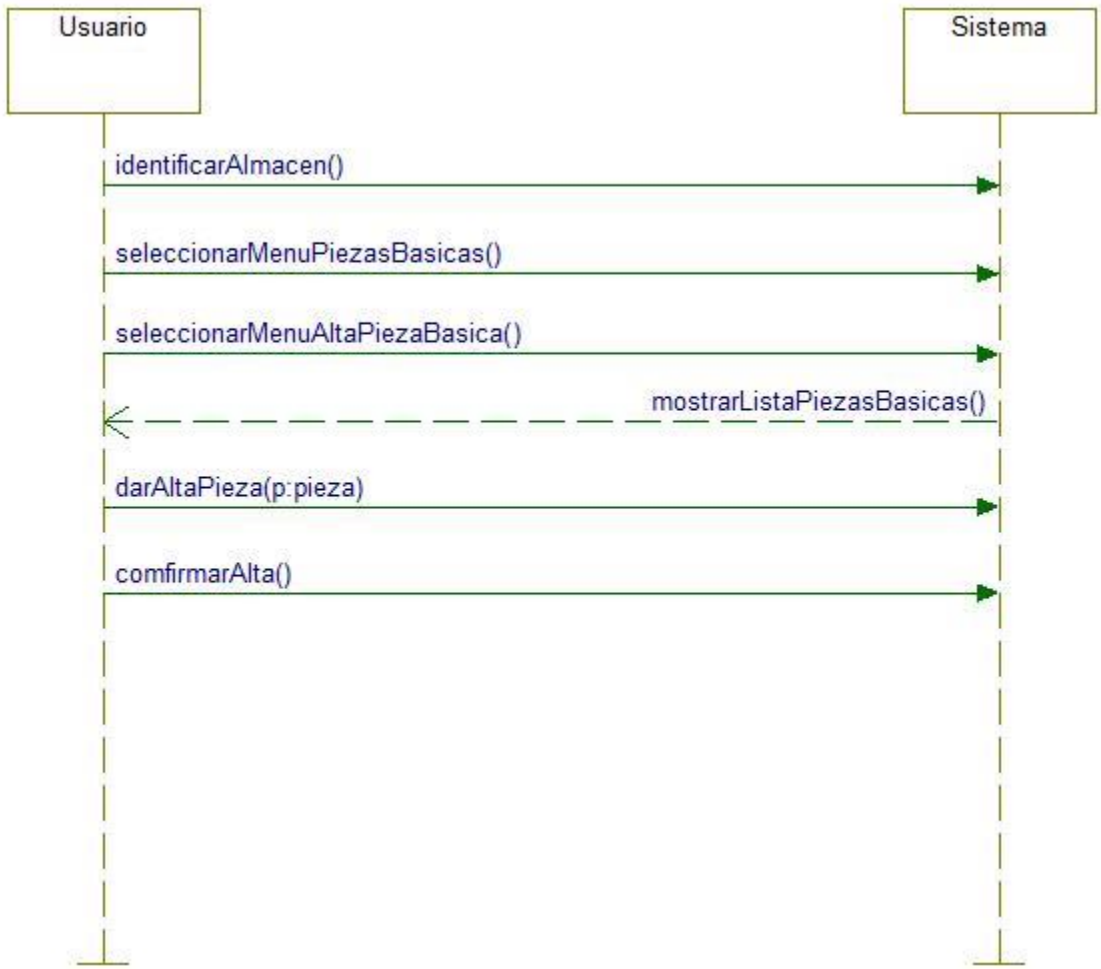


DIAGRAMA DE COLABORACIÓN - MODIFICAR STOCK PIEZA (CU34)

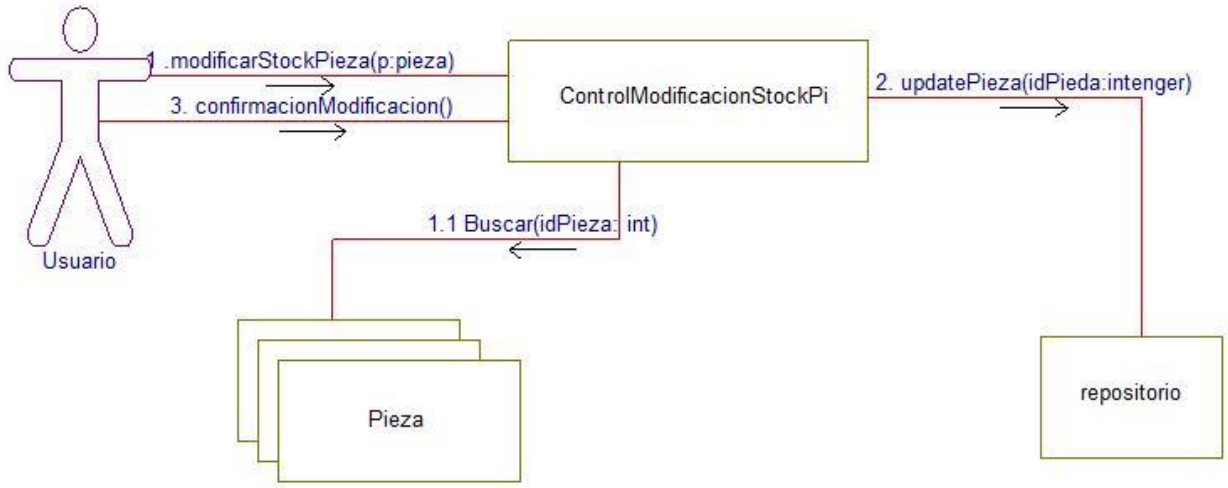


DIAGRAMA DE SECUENCIA - MODIFICAR STOCK PIEZA (CU34)

DIAGRAMA DE COLABORACIÓN - ALTA FACTURA (CU45)

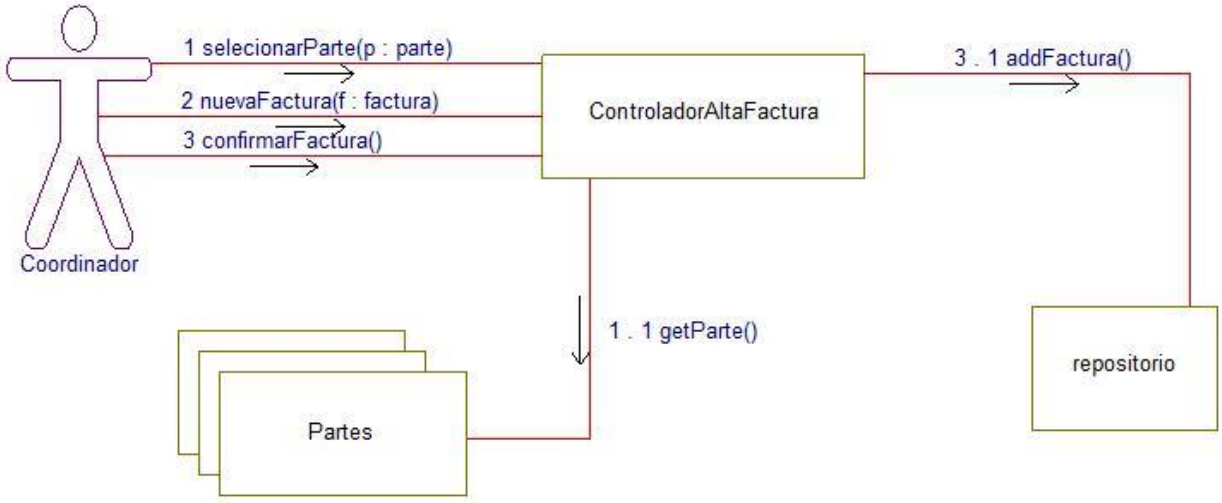


DIAGRAMA DE SECUENCIA - ALTA FACTURA (CU45)