

Universidad de las Ciencias Informáticas
Facultad 6



**Título: Sistema de gestión integral de la planificación
y control del servicio de comedores en la UCI:
análisis y diseño del módulo “Gestión del Menú”**

Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas

Autoras: Kenia Galvez Rodríguez

Yamilka Tamayo Baez

Tutora: Ing. Ailime Conrado Barthelemy

Junio, 2008

“No hay más que asomarse a las puertas de la tecnología y la ciencia contemporánea para preguntarnos si es posible vivir y conocer ese mundo futuro sin un enorme caudal de preparación y conocimientos”.

Fidel Castro

Declaración de autoría

Declaramos ser las únicas autoras de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Kenia Galvez Rodríguez

Yamilka Tamayo Baez

Ing. Ailime Conrado Barthelemy

Firma del autor

Firma del autor

Firma del tutor

Datos de Contacto

Tutora:

Ing. Ailime Conrado Barthelemy.

Universidad de las Ciencias Informáticas, Habana, Cuba.

Email: ailime@uci.cu

Agradecimientos

Le agradecemos a Fidel por haber creado este proyecto y por dejarnos formar parte del mismo.

A todo el equipo del proyecto Alimentos UCI por su ayuda en todo momento.

A nuestra tutora por su apoyo incondicional.

Al profesor Aliesky Sarmiento por habernos atendido siempre que lo necesitamos.

A todos nuestros compañeros de grupo, a los que han permanecido junto a nosotros durante estos 5 años de esfuerzos y a los que por una razón u otra ya no se encuentran en nuestras aulas.

A nuestras madres, abuelos, padrastros, vecinos y amigos por el apoyo brindado y la ayuda incondicional.

Dedicatoria

Todos los resultados que obtuvimos en esta investigación y durante los años de estudio por los que transitamos, se lo dedicamos a nuestra familia, por haber estado junto a nosotras en todo momento, por haber sido nuestra guía, nuestro apoyo y nuestra fuerza.

A nuestras amigas, que lo serán para siempre, Martha, Anabel y Ana Isabel.

En especial:

Kenia: A mi mamá por ser mi guía y el mejor ejemplo a seguir.

A mi padrastro, que más que eso ha sido mi padre.

A Silvia, Mayito, Néstor y Silvita por creer.

A Reynaldo, Zulema y Mirta.

A Marlene y a todos los García por su ayuda.

A Nani porque juntas empezamos y demostramos que podíamos.

A Harley por el apoyo y las fuerzas en esta última etapa tan difícil.

Yamilka: A mi madre Teresita, a mi abuelo Rafael y a mi padrastro Omedis, por estar a mi lado.

A Delia (yeya) y a Manuel, por sus consejos y su cariño.

Especialmente a mi abuela Teresa, a pesar de que ya no la tengo a mi lado, porque a ella le debo lo que soy.

Resumen

La Vicerrectoría de Logística (VRL) de la Universidad de las Ciencias Informáticas (UCI), cuenta con una Dirección General de Alimentos (DGA), en la cual se llevan a cabo una serie de procesos que son de vital importancia para lograr un buen funcionamiento logístico en la UCI.

En la DGA está definido el proceso de planificación y control del servicio de comedores el cual tiene implícitas diferentes actividades: Planificación de un menú a partir de las existencias, Cálculo de la cantidad de comensales, Reajuste de la planificación diaria del menú y el Cálculo de la cantidad de alimento a utilizar, además se encuentra definido el Análisis Económico como proceso de apoyo para alcanzar la eficiencia en esas actividades.

En la actualidad se realiza una planificación deficiente de muchas de las actividades que se llevan a cabo en este proceso, sobre todo las relacionadas con la información que se gestiona referente al menú, lo que trae consigo que este esté sujeto a constantes cambios y que generalmente no cumpla con los indicadores requeridos: Balance Nutricional, Norma de Consumo y Frecuencia Establecida. Debido a esto la DGA necesita una herramienta informática que soporte dicho proceso.

Es por ello que el objetivo de esta investigación es realizar el análisis y diseño de un sistema para la gestión integral de los procesos de planificación y control del servicio de comedores UCI específicamente de la Gestión del Menú.

Palabras Claves: Producto, Plato, Menú, Comensales, Asignación de alimentos, Listado de productos disponibles, Plan de Carga.

Índice

AGRADECIMIENTOS.....	I
DEDICATORIA	II
RESUMEN.....	III
INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	4
Introducción	4
1.1 Aplicaciones informáticas relacionadas con el campo de acción	4
1.1.1 Sistemas de Gestión de la Información	4
1.1.2 Sistemas similares a nivel internacional	4
1.1.3 Sistema de gestión de la información a nivel nacional	6
1.2 Análisis de la solución propuesta.....	6
1.3 Proceso de Desarrollo de Software	7
1.3.1 Proceso Unificado de Desarrollo (Rational Unified Process).....	7
1.3.2 Extreme Programing (XP).....	8
1.4 Roles y artefactos desarrollados en el trabajo	9
1.5 UML como lenguaje de Modelado	10
1.6 Herramientas Case	11
1.7 Herramienta para la gestión de requisitos	12
1.9 Patrones.....	16
1.9.1 Patrones de Casos de Uso	16
1.9.2 Patrones de arquitectura.....	17
1.9.3 Patrones de diseño	17
1.10 Conclusiones	19
CAPITULO 2: CARACTERÍSTICAS DEL SISTEMA.....	21
Introducción	21
2.1 Objeto de Estudio.....	21
2.1.1 Objetivos estratégicos de la organización.....	21
2.1.2 Descripción de los procesos de negocio actuales	22
2.1.3 Análisis crítico de los procesos	23
2.2 Modelo del negocio actual	24
2.2.1 Reglas del Negocio	24
2.2.2 Actores y trabajadores del negocio	25
2.2.3 Modelo de Casos de Uso del Negocio.....	26
2.2.4 Descripciones de los CU del negocio	27
2.2.5 Modelo de Objetos del Negocio	29
2.3 Modelado del Sistema.....	31
2.3.1 Requerimientos Funcionales y no Funcionales	31
Requerimientos funcionales.....	31
2.3.2 CU del Sistema	36
2.3.3 Actores del Sistema	37

2.3.4 Matriz de Trazabilidad.....	38
2.3.5 Diagrama de CU del Sistema.....	39
2.3.6 Descripciones de los Casos de Uso del Sistema.....	42
2.3.7 Vista de Casos de Uso.....	50
2.4 Conclusiones	50
CAPITULO 3: ANÁLISIS Y DISEÑO.....	52
Introducción	52
3.1 Arquitectura del sistema.....	52
3.1.1 Vista Lógica.....	52
3.1.2 Vista de despliegue.....	53
3.2 Prototipos de interfaz no funcionales	54
3.3 Mapa de navegación.....	54
3.4 Aplicación de los Patrones de diseño	55
3.5 Modelo del diseño	56
3.5.1 Realización de Casos de Uso	57
3.5.2 Descripciones de las Clases del Diseño	59
3.5.3 Diagramas de Clases del diseño	63
3.5.4 Diagramas de Secuencia	80
3.6 Validación del diseño realizado	87
3.7 Conclusiones	88
CONCLUSIONES GENERALES	89
RECOMENDACIONES.....	90
REFERENCIAS BIBLIOGRÁFICAS	91
BIBLIOGRAFÍA.....	93
ANEXOS	94
GLOSARIO DE TÉRMINOS.....	102

Índice de Figuras

FIG. 1 DIAGRAMA DE CU DEL NEGOCIO.....	26
FIG. 2 MODELO DE OBJETOS DEL CU CONFECCIONAR MENÚ	29
FIG. 3 MODELO DE OBJETOS DEL CU CONFECCIONAR MENÚ SEMANAL.....	30
FIG. 4 MODELO DE OBJETOS DEL CU PLANIFICAR DIARIAMENTE EL MENÚ Y AJUSTARLO	30
FIG. 5 MODELO DE OBJETOS DEL CU ELABORAR ASIGNACIÓN DE ALIMENTOS PARA EL PRÓXIMO PERÍODO.....	31
FIG. 6 DIAGRAMA DE GENERALIZACIÓN DE ACTORES DEL SISTEMA.....	38
FIG. 7 DIAGRAMA DE PAQUETES DEL SISTEMA	39
FIG. 8 DIAGRAMA DE CU DEL PAQUETE CLIENTE	40
FIG. 9 DIAGRAMA DE CU DEL PAQUETE GESTOR_PRODUCTOS.....	40
FIG. 10 DIAGRAMA DE CU DEL PAQUETE GESTOR_EXISTENCIAS.....	40
FIG. 11 DIAGRAMA DE CU DEL PAQUETE JEFE DE COMPRAS.....	40
FIG. 12 DIAGRAMA DE CU DEL PAQUETE DIRECTOR GENERAL DE ALIMENTOS	41
FIG. 13 DIAGRAMA DE CU DEL PAQUETE PLANIFICADOR DEL MENÚ	41
FIG. 14 VISTA DE CU DEL SISTEMA	50
FIG. 15 DIAGRAMA DE LA VISTA LÓGICA.....	53
FIG. 16 VISTA DE DESPLIEGUE	54
FIG. 17 REALIZACIÓN DEL CU GESTIONAR PRODUCTOS	57
FIG. 18 REALIZACIÓN DEL CU GESTIONAR PLATO	57
FIG. 19 REALIZACIÓN DEL CASO DE USO HALLAR CARACTERÍSTICAS DEL PLATO	57
FIG. 20 REALIZACIÓN DEL CASO DE USO APROBAR PLATO	57
FIG. 21 REALIZACIÓN DEL CASO DE USO GESTIONAR ASIGNACIÓN DE ALIMENTOS PARA EL PRÓXIMO PERÍODO.....	57
FIG. 22 REALIZACIÓN DEL CU CREAR SOLICITUD DE MENÚ	57
FIG. 23 REALIZACIÓN DEL CASO DE USO APROBAR SOLICITUD DE MENÚ	58
FIG. 24 REALIZACIÓN DEL CASO DE USO MOSTRAR EXISTENCIAS	58
FIG. 25 REALIZACIÓN DEL CASO DE USO GESTIONAR CARGA	58
FIG. 26 REALIZACIÓN DEL CASO DE USO GESTIONAR PLAN DE CARGA.....	58
FIG. 27 REALIZACIÓN DEL CASO DE USO GESTIONAR MENÚ	58
FIG. 28 REALIZACIÓN DEL CASO DE USO CONFIRMAR MENÚ.....	58
FIG. 29 REALIZACIÓN DEL CASO DE USO GESTIONAR NORMAS NUTRICIONALES	58
FIG. 30 REALIZACIÓN DEL CASO DE USO GESTIONAR VALOR NUTRICIONAL	59
FIG. 31 DIAGRAMA DE CLASES DEL DISEÑO WEB DEL CU APROBAR PLATO	65
FIG. 32 DIAGRAMA DE CLASES DEL DISEÑO (WEB) DEL CU GESTIONAR PRODUCTOS.....	66
FIG. 33 DIAGRAMA DE CLASES DEL DISEÑO (WEB) DEL CU GESTIONAR PLATO.....	67
FIG. 34 DIAGRAMA DE CLASES DEL DISEÑO (WEB) DEL CU HALLAR CARACTERÍSTICAS DEL PLATO.....	68
FIG. 35 DIAGRAMA DE CLASES DEL DISEÑO (WEB) DEL CU GESTIONAR ASIGNACIÓN DE ALIMENTOS PARA EL PRÓXIMO PERÍODO	69
FIG. 36 DIAGRAMA DE CLASES DEL DISEÑO (WEB) DEL CU CREAR SOLICITUD DE MENÚ	70
FIG. 37 DIAGRAMA DE CLASES DEL DISEÑO (WEB) DEL CU APROBAR SOLICITUD MENÚ	71
FIG. 38 DIAGRAMA DE CLASES DEL DISEÑO (WEB) DEL CU MOSTRAR EXISTENCIAS	72
FIG. 39 DIAGRAMA DE CLASES DEL DISEÑO (WEB) DEL CU GESTIONAR CARGA	73
FIG. 40 DIAGRAMA DE CLASES DEL DISEÑO (WEB) DEL CU GESTIONAR PLAN DE CARGA.....	74
FIG. 41 DIAGRAMA DE CLASES DEL DISEÑO (WEB) DEL CU GESTIONAR MENÚ.....	75
FIG. 42 DIAGRAMA DE CLASES DEL DISEÑO (WEB) DEL CU CONFIRMAR MENÚ	76
FIG. 43 DIAGRAMA DE CLASES DEL DISEÑO (WEB) DEL CU GESTIONAR NORMAS NUTRICIONALES	77
FIG. 44 DIAGRAMA DE CLASES DEL DISEÑO (WEB) DEL CU GESTIONAR VALOR NUTRICIONAL	78
FIG. 45 FRAGMENTO DEL PAQUETE MODELO	79
FIG. 46 DIAGRAMA DE SECUENCIA DEL CU APROBAR PLATO	81

FIG. 47 DIAGRAMA DE SECUENCIA DEL CU GESTIONAR PLATO: ESC. BUSCAR.....	82
FIG. 48 DIAGRAMA DE SECUENCIA DEL CU GESTIONAR PLATO: ESC. CREAR	83
FIG. 49 DIAGRAMA DE SECUENCIA DEL CU GESTIONAR PLATO: ESC. ELIMINAR.....	84
FIG. 50 DIAGRAMA DE SECUENCIA DEL CU GESTIONAR PLATO: ESC. MODIFICAR	85
FIG. 51 DIAGRAMA DE SECUENCIA DEL CU GESTIONAR PLATO: ESC. MOSTRAR	86
FIG. 54 MAPA DE PROCESOS LOGÍSTICOS (21).....	94
FIG. 55 ORGANIGRAMA DE LA VICERRECTORÍA DE LOGÍSTICA.....	94
FIG. 56 GRÁFICA DE RUP	95
FIG. 57 REPRESENTACIÓN DEL PATRÓN MODELO VISTA CONTROLADOR	95
FIG. 58 PROTOTIPO NO FUNCIONAL DEL CU APROBAR PLATO	96
FIG. 59 PROTOTIPO NO FUNCIONAL DEL CU GESTIONAR PLATO. SECCIÓN "CREAR PLATO"	96
FIG. 60 PROTOTIPO NO FUNCIONAL DEL CU GESTIONAR PLATO. SECCIÓN "EDITAR PLATO"	97
FIG. 61 PROTOTIPO NO FUNCIONAL DEL CU GESTIONAR PLATO. SECCIÓN "BUSCAR PLATO".....	97
FIG. 62 PROTOTIPO NO FUNCIONAL DEL CU GESTIONAR PLATO. SECCIÓN "MOSTRAR PLATO"	98
FIG. 63 PROTOTIPO NO FUNCIONAL DEL CU GESTIONAR PLATO. SECCIÓN "ELIMINAR PLATO".....	98
FIG. 64 MAPA DE NAVEGACIÓN 1 (PARA EL JEFE DE ALMACÉN, EL JEFE DE COMPRAS Y EL SOLICITUD CLIENTE).....	99
FIG. 65 MAPA DE NAVEGACIÓN 2 (PARA EL PLANIFICADOR DEL MENÚ)	100
FIG. 66 MAPA DE NAVEGACIÓN 3 (PARA EL DIRECTOR GENERAL DE ALIMENTOS)	101

Índice de Tablas

Tabla 1 Actores del Negocio.....	25
Tabla 2 Trabajadores del Negocio.....	26
Tabla 3 Descripción textual del CU Confeccionar Menú	27
Tabla 4 Descripción textual del CU Planificar diariamente el menú y ajustarlo.....	28
Tabla 5 Descripción textual del CU Elaborar Asignación de Alimentos para el próximo período.....	28
Tabla 6 Descripción textual del CU Confeccionar el Menú Semanal	29
Tabla 7 Actores del sistema.....	37
Tabla 8 Descripción del CU del Sistema “Gestionar Productos”	42
Tabla 9 Descripción del CU del Sistema “Gestionar Plato”	43
Tabla 10 Descripción del CU del Sistema “Hallar las características del plato”	43
Tabla 11 Descripción del CU del Sistema “Aprobar Plato”	44
Tabla 12 Descripción del CU del Sistema “Gestionar Asignación de Alimentos para el próximo período”	44
Tabla 13 Descripción del CU del Sistema “Crear solicitud de menú”	45
Tabla 14 Descripción del CU del Sistema “Aprobar solicitud de menú”	45
Tabla 15 Descripción del CU del Sistema “Mostrar Existencias”	46
Tabla 16 Descripción del CU del Sistema “Gestionar Carga”	46
Tabla 17 Descripción del CU del Sistema “Gestionar Plan de Carga”	47
Tabla 18 Descripción del CU del Sistema “Gestionar menú”	48
Tabla 19 Descripción del CU del Sistema “Confirmar menú”	48
Tabla 20 Descripción del CU del Sistema “Gestionar Normas Nutricionales”	49
Tabla 21 Descripción del CU del Sistema “Gestionar Valor Nutricional”	50
Tabla 22 Descripción de la Clase del Diseño de Aplicaciones Web: Fr_Aprobarplato.....	59
Tabla 23 Descripción de la Clase del Diseño de Aplicaciones Web: PlatoPeer	59
Tabla 24 Descripción de la Clase del Diseño de Aplicaciones Web: aprobar_plato_Actions.....	60
Tabla 25 Descripción de la Clase del Diseño de aplicaciones Web: IntermediaPlato	61
Tabla 26 Descripción de la Clase del Diseño de Aplicaciones Web: BasePlato	63

Introducción

La Universidad de las Ciencias Informáticas (UCI) fue creada por el Comandante en Jefe Fidel Castro Ruz en marzo del 2002, fundamentando la idea de que la informática en el país podría convertirse en una potente fuerza científica y económica para el desarrollo del mismo. La UCI comenzó su primer curso, en septiembre de ese mismo año, con la misión de formar especialistas en ciencias informáticas de primer nivel, además de generar productos y servicios informáticos, así como brindar soluciones tecnológicas integrales para la economía nacional a través de un amplio grupo de programas.

La UCI cuenta con un Departamento de VRL en el cual se llevan a cabo diferentes procesos clasificados en: Estratégicos, Claves y de Apoyo (Ver anexo 1). Esta investigación está enfocada en los procesos Claves, los cuales expresan el objeto social y la razón de existencia de este departamento, dentro de este proceso en el servicio de comedores.

Este contiene una serie de subprocesos dentro de los que se destaca: planificación y control del servicio de comedores y análisis económico, este último enfocado al anterior por ser clave para lograr la eficiencia de todas las actividades que tiene implícitas dicho proceso, estas actividades son:

- Confección del menú a partir de las existencias.
- Cálculo de la cantidad de alimento a utilizar.
- Planificación de la cantidad de comensales.
- Reajuste de la planificación diaria.

El proceso de planificación y control del servicio de comedores se lleva a cabo en la VRL, aunque tiene su mayor expresión en la DGA y ATM fundamentalmente, estando estas dos áreas muy interrelacionadas pues para la ejecución de las actividades anteriormente mencionadas en la DGA, es necesario consultar datos y documentación existentes en ATM, lo que convierte a la DGA en un cliente de esta área. (Ver anexo 2)

Los problemas que se presentan actualmente están relacionados con el trabajo manual que realizan los involucrados en este proceso, pues deben manejar gran cantidad de información para elaborar los diferentes documentos que se realizan como consecuencia del desarrollo del mismo, los cuales incluyen numerosos cálculos, la entrega de estos debe efectuarse en el menor tiempo posible y con la mayor precisión.

En estos momentos las decisiones que se toman están basadas en la experiencia y apreciación de los directivos del sistema logístico lo que trae consigo en muchas ocasiones errores en la planificación de los servicios.

Por otra parte no está definido un flujo adecuado de información, lo cual provoca que los datos estén dispersos y sean difíciles de encontrar y en la mayoría de los casos ni siquiera se registran.

Actualmente existe inconformidad con los procesos de planificación y control que se llevan a cabo, específicamente lo que respecta al menú. Debido a que no existe integración, entre la Planificadora del menú y ATM se invierte mucho tiempo en la confección del menú, además este está sujeto a constantes variaciones y la mayoría de las veces no cumple con los indicadores requeridos (Balance Nutricional, Normas de Consumo, Frecuencias Establecidas). La mala planificación de estos servicios trae como consecuencia la pérdida de gran cantidad de alimentos, afectando económicamente a la universidad y provocando inconformidad en los comensales.

Producto de la importancia que tiene la gestión de la información del menú varias empresas (General Soft SA, Cadre Distribución S.L) con grandes experiencias en el proceso de desarrollo de software se han dedicado a crear herramientas que contribuyan con la gestión del menú, pero para comercializar sus productos establecen cotas de pago muy elevadas, por lo que comprarlos y mantenerlos crearía dependencias con el proveedor, con costos adicionales, por lo que el país no puede adquirirlos ya que provocaría grandes pérdidas económicas.

En Cuba algunas instituciones (Instituto Superior Politécnico José Antonio Echeverría (ISPJAE)) se han proyectado en este sentido, como resultado se han desarrollado algunos software pero su uso en la universidad no sería factible, pues no resolverían todos los problemas a los que se enfrenta la DGA.

Se puede concluir que el proceso de planificación y control del menú no fluye de manera eficiente, por lo que se identifica como **problema científico**: ¿Cómo contribuir a la gestión de la información en la planificación y control del menú en la UCI?

El problema planteado se enmarca en el **objeto de estudio**: Los procesos de gestión de la información en el servicio de comedores.

El objeto delimita el **campo de acción**: Los procesos de gestión de la información en la planificación y control del menú en la UCI.

Para dar solución al problema se define como **objetivo general**: Desarrollar el análisis y el diseño del módulo "Gestión del menú" del sistema de gestión integral de la planificación y control del servicio de comedores en la UCI.

Objetivos Específicos:

- Analizar los procesos que se llevan a cabo en la gestión del menú del sistema de gestión integral de la planificación y control del servicio de comedores en la UCI.

- Identificar las funcionalidades que debe cumplir el Módulo: Gestión del Menú.
- Diseñar el Módulo: Gestión del Menú.

Para lograr los objetivos se desarrollarán las siguientes **tareas**:

1. Realización de entrevistas al cliente para lograr la familiarización con el proceso de gestión del menú del sistema de gestión integral de la planificación y control del servicio de comedores en la UCI.
2. Estudio del Informe del Proyecto de Gestión Integral del servicio de comedores.
3. Estudio de los Sistemas de Gestión de la Información para la planificación y control del servicio de comedores.
4. Estudio de las tecnologías que se emplearán para el desarrollo de la investigación.
5. Selección de las tecnologías que se emplearán para el desarrollo de la investigación.
6. Modelación del proceso de gestión del menú.
7. Obtención, descripción y validación de los requerimientos.
8. Realización del diseño de las clases candidatas a la implementación.

El documento está estructurado de la siguiente manera:

Capítulo 1. Fundamentación teórica

Este capítulo comprende el estado del arte de la Gestión de la Información en la planificación y control del menú, y se realiza una descripción del problema. Se describe la justificación de las tecnologías y metodologías utilizadas para la solución de la problemática planteada.

Capítulo 2. Características del Sistema

En este capítulo se hace una definición del objeto de estudio del problema, se plantean los objetivos estratégicos de la organización y procesos de negocio que los soportan. Descripción de los procesos que serán objeto de automatización. Descripción general de la propuesta de sistema, como debe funcionar. Definición del Modelo de Negocio. Especificación de los requisitos y Casos de uso (CU) del sistema.

Capítulo 3. Análisis y diseño del sistema

En este capítulo se obtienen los artefactos correspondientes al flujo de trabajo de análisis y diseño. Se modela el diseño, donde se representan las realizaciones de los CU del sistema, a través de las clases del diseño con extensiones Web además de las descripciones de las mismas.

Capítulo 1: Fundamentación Teórica

Introducción

Este capítulo comprende el estado del arte de la Gestión de la Información en la planificación y control del menú, y se realiza una descripción del problema. Se describe la justificación de las tecnologías y metodologías utilizadas para la solución de la problemática planteada.

1.1 Aplicaciones informáticas relacionadas con el campo de acción

1.1.1 Sistemas de Gestión de la Información

La información, es un elemento fundamental para el desarrollo, con el transcurso de los años, la gestión de esta ocupa un espacio mayor en la economía de los países a escala mundial. Para poder utilizarla en la toma de decisiones, la misma debe gestionarse (recabar, registrar, analizar) de manera eficiente para lograr que las decisiones que se tomen sean cada vez más acertadas.

1.1.2 Sistemas similares a nivel internacional

La gestión de la información de los servicios que se prestan en los comedores que tienen grandes cantidades de comensales se hace cada vez más imprescindible, es por esto que en el mundo existen empresas que se dedican a desarrollar aplicaciones que contribuyen en la optimización de los procesos de planificación y control del menú, en tal caso se encuentran:

Cadre Distribución S.L con su software **SATN2007**, el cual es una nueva versión del **SATN2001**, está enfocado fundamentalmente al área de la salud. El mismo cuenta con una comprensiva base de información y conocimientos sobre los alimentos, brindando la posibilidad de realizar análisis completos de dietas, recetas y menús. Permite hallar los objetivos nutricionales que conducen a una salud óptima, utiliza diferentes tablas dietéticas confeccionadas, en algunos casos por organismos oficiales y en otros elaborados por entidades o asociaciones científicas. Realiza también el cálculo del consumo basal, la termogénesis de la dieta, el consumo físico y las cantidades de todos y cada uno de los nutrientes que se necesitan individualmente. Establece los requerimientos nutricionales para distintas situaciones fisiológicas (lactantes, niños, adolescentes, hombres, mujeres adultas) y para los distintos nutrientes (energía, proteínas, vitaminas, minerales). Además posee una Base de Datos de alimentos con información nutritiva de más de 800 productos.

En países como EE UU y Chile se utiliza **Food Processor** (Versión Demo - Inglés) el cual es un excelente programa desarrollado específicamente para profesionales de la nutrición. Este permite la creación de menús, que pueden ser incluidos en una base de datos para seguir utilizándolos

posteriormente, lo cual permite calcular el requerimiento y gasto calórico de una forma más precisa, por ello, es ideal para armar planes de alimentación que incluyen actividad física. Otras funciones especiales incluyen: creación de etiquetas, extrapolación de datos a la pirámide alimentaria y valor biológico de las proteínas consumidas.

La empresa **GeneralSoft SA** ubicada en Quito, Ecuador, es la productora del software **EquiLibra Professional**, el cual proporciona más de 1500 platos nacionales e internacionales con sus respectivos componentes nutricionales y su preparación culinaria. Dispone de un mecanismo de intercambios nutricionales utilizando todos los platos del sistema lo que permite personalizar dietas al gusto y la disponibilidad de los alimentos, permite realizar búsquedas automáticas de alimentos y platos preparados desde la base de datos, para equilibrar en una forma fácil y rápida el desbalance nutricional a través de los alimentos más adecuados. Además maneja y controla en forma histórica un número ilimitado de usuarios o grupos de usuarios, entre otras funcionalidades. Permite también elaborar dietas profesionalmente equilibradas, en algunos casos estas pueden ser generadas para instituciones con control de costos y canasta de compras.

ASSETS NS es un sistema comercializado por la firma panameña D' Marco S.A. Este es un sistema de gestión integral estándar y parametrizado que permite el control de los procesos de compras, ventas, producción, taller, inventario, finanzas, contabilidad, presupuesto, activos fijos, útiles, herramientas y recursos humanos. Dispone, además, de métodos novedosos para administración y planificación de inventarios, así como una amplia gama de análisis y consultas que permiten no sólo conocer exactamente la situación actual, sino proyectar decisiones futuras. El ASSETS NS facilita adaptarse a las exigencias de cada entidad en particular, garantizando que sus reportes tengan la forma y el contenido que el usuario les defina. Este sistema está diseñado para Multi Compañía, con una estructura organizativa a varios niveles, en la que podrán existir: Grupo Corporativo, Grupo de Agrupaciones, Almacenes y Centros de Costos. Para entidades con esta estructura se brinda un módulo de comunicaciones que facilita poder intercambiar información entre ellas, con el fin de consolidar la información sobre la Gestión Comercial y Contable, pudiéndose obtener los estados financieros, resúmenes de compras, ventas, entre otros, a distintos niveles. En Cuba este sistema es utilizado fundamentalmente en grandes almacenes. En la UCI se usa en ATM debido a que es el almacén central de la universidad en el cual están concentrados la mayor parte de los productos con los que cuenta la misma.

Estas aplicaciones informáticas no pueden ser utilizadas en la universidad, pues estas solo contemplan para sus respectivas empresas una parte de las actividades del proceso de planificación y control del

servicio de comedores, en las cuales, por ejemplo: no contemplan la relación entre los comensales y el menú, no tiene implementada una funcionalidad que permita reajustar el menú diariamente evidenciándose la relación con la existencia de productos. Debido a esto la DGA continuaría sin resolver los problemas a los que se enfrenta.

1.1.3 Sistema de gestión de la información a nivel nacional

En el país algunas instituciones se han proyectado hacia el desarrollo de software que contribuyan a lograr una buena planificación y control del servicio de comedores. En la ISPJAE, por ejemplo, se utiliza el Sistema de Control de Comedores (SISCOMED), este usa un sistema de tarjetas magnéticas, permite obtener la cantidad de comensales registrados que reciben cada servicio de manera diaria o por mes en los comedores; incluyendo los comensales que pasan dos veces por el comedor en un mismo evento (desayuno, almuerzo, comida). El sistema no puede ser utilizado en el centro debido a que el mismo no brinda la posibilidad de gestionar la información referente al Plan de Carga, no permite obtener el Listado de productos disponibles, de la misma forma que no está integrado al sistema ASSETS, además no contempla la integración entre la asignación de alimentos y el menú.

En la UCI se han obtenido algunas aplicaciones informáticas con el fin de erradicar las dificultades que afronta la DGA, pero ninguna ha tenido un resultado totalmente eficiente para los directivos y personal involucrado en el proceso de planificación y control del servicio de comedores, pues estas aplicaciones han estado enfocadas a resolver solamente los problemas relacionados con los comensales, ellas son: el Sistema de Control de Acceso, el Sistema de Reservación, el Sistema de Asignación de Comensales y el Sistema de Reportes.

1.2 Análisis de la solución propuesta

Los trabajadores de la VRL, específicamente los de la DGA, necesitan una herramienta informática que les facilite la gestión de la información en el proceso de planificación y control del menú. Esta herramienta debe contribuir a tener un rápido acceso a los documentos necesarios para la planificación del menú por ejemplo: Modelo de Existencia en ATM, Plan de Carga, Asignación de Alimentos del período anterior, entre otros. Entre este sistema y el ASSETS utilizado en el área de ATM debe existir una conexión la cual facilitará la obtención de las existencias reales en dicho almacén.

Para darle solución a las necesidades del personal que labora en estas áreas se propone como objetivo general desarrollar el análisis y el diseño del módulo “Gestión del menú” del sistema de gestión integral de la planificación y control del servicio de comedores en la UCI, siendo este sistema

propuesto una aplicación web debido a que permite interactuar con las bases de datos y distribuir la información. Su uso es más fácil, pues no necesitan ser instaladas, solamente se requiere de un servidor Web al cual se accederá desde cualquier parte de la universidad, a cualquier hora y por múltiples usuarios al mismo tiempo, además no es necesario que los mismos sean especialistas en la navegación y utilización del sistema.

1.3 Proceso de Desarrollo de Software

En el proceso de ingeniería de software se definen un conjunto de etapas parcialmente ordenadas para lograr un objetivo, en este caso, la obtención de un producto de software. El **proceso de desarrollo de software** “es aquel en que las necesidades del usuario son traducidas en requerimientos de software, estos requerimientos transformados en diseño y el diseño implementado en código, el código es probado, documentado y certificado para su uso operativo. Concretamente define quién está haciendo qué, cuándo y cómo alcanzar un cierto objetivo” (1). El proceso de desarrollo de software requiere un conjunto de conceptos, una metodología y un lenguaje propio. Este proceso es también conocido como ciclo de vida del software.

1.3.1 Proceso Unificado de Desarrollo (Rational Unified Process)

El Proceso Unificado de Desarrollo Software (RUP) se caracteriza por estar dirigido por casos de uso, centrado en la arquitectura y ser iterativo e incremental. RUP es un marco de desarrollo compuesto de cuatro fases:

- Inicio
- Elaboración
- Construcción
- Transición

Cada una de estas fases está dividida en una serie de iteraciones que ofrecen como resultado un incremento del producto desarrollado, que añade o mejora las funcionalidades del sistema en desarrollo. En estas iteraciones se realizan las actividades definidas en el ciclo de vida clásico, aunque todas las iteraciones suelen incluir trabajo en la mayoría de estas actividades (Ver anexo 3), el grado de esfuerzo dentro de cada una de ellas varía a lo largo del proyecto y una actividad se puede realizar completamente en más de una iteración en dependencia de la complejidad del proyecto. RUP tiene

definido cuatro elementos: los roles, que responden al ¿Quién?, los artefactos, que responden al ¿Cómo?, las actividades que responden al ¿Qué? y los flujos de trabajo, que responden al ¿Cuándo?

El **rol** define las responsabilidades de los individuos del equipo de trabajo, cada individuo puede desempeñar varios roles y un rol puede ser representado por varios individuos. La **actividad** es una unidad de trabajo que un individuo que desempeñe un rol puede realizar. Las actividades tienen un objetivo concreto, crear o actualizar algún producto. El **artefacto** es un fragmento de información que es producido y usado durante el proceso de desarrollo de software. Los artefactos son los resultados tangibles del proyecto que se van creando y usando hasta obtener el producto final. El **flujo de trabajo** es una relación de actividades que producen resultados observables dado por una secuencia de actividades realizadas por los diferentes roles. RUP tiene definidos los siguientes flujos de trabajo:

- Modelado del negocio
- Requerimientos
- Análisis y Diseño
- Implementación
- Pruebas
- Despliegue
- Gestión del proyecto
- Configuración y control de cambios
- Entorno

RUP utiliza el Lenguaje Unificado de Modelado (Unified Modeling Language, UML) como lenguaje de notación.

1.3.2 Extreme Programming (XP)

Extreme Programming es una de las metodologías de desarrollo de software más exitosas en la actualidad utilizadas para proyectos de corto plazo, corto equipo y cuyo plazo de entrega era ayer. La metodología consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto (2). La metodología se basa en:

Pruebas Unitarias: se basa en las pruebas realizadas a los principales procesos, de tal manera que adelantándonos en algo hacia el futuro, podamos hacer pruebas de las fallas que pudieran ocurrir.

Refabricación: se basa en la reutilización de código, para lo cual se crean patrones o modelos estándares, siendo más flexible al cambio.

Programación en pares: una particularidad de esta metodología es que propone la programación en pares, la cual consiste en que dos desarrolladores participen en un proyecto en una misma estación de trabajo. Cada miembro lleva a cabo la acción que el otro no está haciendo en ese momento.

Debido a los beneficios que ofrece y a las características que posee la investigación a realizar, además de la opinión de los especialistas y personal involucrado se decidió utilizar RUP como metodología para guiar el proceso de desarrollo de software, por estar concebida para proyectos complejos, de larga duración, además durante este proceso se exige generar una serie de documentos necesarios tanto para el cliente como para el desarrollador.

1.4 Roles y artefactos desarrollados en el trabajo

En el desarrollo del proyecto en el cual está enmarcada esta investigación, se han tenido en cuenta una serie de roles y artefactos que están dentro de los flujos de trabajo definidos por RUP y que se implementarán en el mismo:

1- Modelo de Negocio

Roles:

- Analista de los procesos del negocio
- Diseñador del negocio

Artefactos:

Actor del negocio
Trabajadores del negocio
Casos de uso del negocio
Descripción de los casos de uso del negocio
Diagrama de casos de uso del negocio
Diagrama de Actividades
Modelo de objetos del negocio
Reglas del negocio

2- Requerimientos

Roles:

- Analista del sistema

- Especificador de Requerimientos

Artefactos:

Actores del sistema
Casos de uso del sistema
Diagrama de casos de uso del sistema
Descripciones textuales de los casos de uso del sistema
Especificación de requerimientos (Requisitos funcionales y no funcionales)
Vista de casos de uso

3- Análisis y Diseño

Roles:

- Diseñador de Interfaz de Usuario
- Diseñador

Artefactos:

Mapa de navegación
Prototipos de interfaz de usuario
Paquetes de diseño
Clases de diseño
Realización de casos de uso del sistema
Diagrama de despliegue
Vista lógica

1.5 UML como lenguaje de Modelado

El Lenguaje de Modelado Unificado (UML) es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra gran cantidad de software (3), es además estándar, consolidado y Orientado a Objetos. Los elementos de UML se muestran mediante diagramas que presentan múltiples vistas del sistema, ese conjunto de vistas son conocidas como modelos. Este lenguaje recomienda utilizar los procesos que otras metodologías tienen definidos. Se puede decir que una parte de UML define una abstracción con significado de un lenguaje para expresar otros modelos, es decir, otras abstracciones de un sistema o un conjunto de unidades conectadas que se organizan para conseguir un propósito. Esto en principio puede parecer complicado pero no es la realidad si logramos comprender que uno de los objetivos de este lenguaje es definir modelos, no solo establecerlos.

Para el desarrollo del software propuesto se utilizará UML como lenguaje de modelado por todas las ventajas que posee y principalmente por ser utilizado por la metodología seleccionada.

1.6 Herramientas Case

Las herramientas CASE (Computer Aided Software Engineering) permiten organizar y mejorar la información de un proyecto informático, haciendo que se torne más flexible, permitiendo una mejor comunicación entre los participantes. En el mundo existen diferentes herramientas CASE para el modelado de artefactos, entre ellas se encuentran: Rational Rose, Visual Paradigm, Umbrello.

Rational Rose: es una herramienta con plataforma independiente. Una de las grandes ventajas del Rational es que utiliza la notación estándar en la arquitectura de Software (UML), la cual permite visualizar el sistema completo utilizando un lenguaje común. Otra ventaja de Rose es que se pueden modelar sus componentes e interfaces en forma individual y luego unirlos con otros componentes del proyecto. Además proporciona un lenguaje común de modelado para el equipo que facilita la creación de software de calidad más rápidamente. Esta herramienta es líder en el mercado para el análisis, modelamiento, diseño y construcción orientado a objetos (4)

Umbrello: es una herramienta libre para crear y editar diagramas UML, que ayuda en el proceso del desarrollo de software, y está diseñado principalmente para KDE, aunque funciona en otros entornos de escritorio. Maneja gran parte de los diagramas estándar UML pudiendo crearlos, incluso manualmente, importándolos a partir de código en C++, Java, Python, Pascal/Delphi, Ada, o también Perl (haciendo uso de una aplicación externa). Así mismo, permite crear un diagrama y generar el código automáticamente en los lenguajes antes citados. El formato de fichero que utiliza está basado en XMI.

Visual Paradigm: es una herramienta fácil de utilizar que apoya las últimas notaciones de UML, generación de código a partir del modelo de este lenguaje, importa diagramas realizados en Rational Rose, exportación/importación XMI, generador de impresos. Sus nuevas características marcan con etiqueta VPTS, subversión y CVS, realces del diagrama de secuencia, recordando estados del nodo del árbol, generan el rubí, el pitón reverso, la terminación. Tiene integración con herramientas como: Java, NetBeans IDE, Oracle JDeveloper. Permite la generación de código en lenguajes muy usados como: C++, PHP, Java, C #, Perl. A diferencia de otras herramientas CASE tiene disponibilidad en múltiples plataformas entre las que se encuentran: Windows, Linux, Mac OSX, entre otras. Puede estar en varios idiomas, es compatible entre ediciones.

Para la realización de esta investigación se decidió utilizar Visual Paradigm Versión 6.0 pues además de todas las características anteriores presenta licencia comercial, es fácil de instalar y actualizar, soporta ingeniería inversa y aplicaciones Web, además presenta un ambiente gráfico agradable al usuario.

1.7 Herramienta para la gestión de requisitos

Los requisitos de software son las necesidades de los clientes para resolver un determinado problema por lo cual, la gestión de requisitos, es una de las partes más importantes del proceso de desarrollo de software. El proceso de gestión de requisitos, no se basa únicamente en la deducción de los mismos, si no que implica más tareas, entre las cuales se destacan:

- Recogida de requisitos
- Formalización de requisitos
- Revisión
- Gestión

Las herramientas que dan soporte a estas tareas son las conocidas como Herramientas de Gestión de Requisitos (5). Entre estas se encuentran:

Open Source Requirement Management Tool (OSRMT), es una herramienta de software libre, presenta licencia GPL y permite la descripción avanzada de diversos tipos de requisitos garantizando la trazabilidad entre todos los documentos relacionados con la ingeniería de requisitos (funcionalidades, requisitos, casos de uso, casos de prueba).

Características y funcionalidad básica de la herramienta

La herramienta integra diversos módulos como son:

- Gestión de documentos de ingeniería de requisitos (funcionalidades, requisitos, casos de uso, casos de prueba)
- Trazabilidad entre documentos de trabajo
- Informes y estadísticas

Algunas de las funcionalidades que ofrece la herramienta son:

Gestión de requisitos, diferenciando entre Requisitos:

- Funcionalidades
- Requisitos técnicos
- Casos de prueba

Trazabilidad entre todos los documentos de trabajo:

- Requisito-Requisito (control de versiones)
- Requisito-Requisito (dependencia entre requisitos)
- Requisito-Funcionalidad
- Requisito-Caso de Prueba
- Visualización de la matriz de trazabilidad
- Árbol de trazabilidad para facilitar las auditorías
- Gráfico de dependencias entre documentos de trabajo para poder determinar el impacto de un cambio

Personalización y configuración:

- Definición de los atributos de una funcionalidad
- Definición de los atributos de un requisito
- Definición de los atributos de un caso de prueba
- Personalización de vistas

Esta herramienta nos permite obtener una visualización de los requisitos en forma jerárquica, la cual es intuitiva y fácil de manejar. Es multiplataforma. Como herramienta open source de gestión de requisitos no tiene mucha competencia en cuanto a la funcionalidad ofrecida.

Tiene una buena documentación pese a tratarse de una herramienta muy reciente. Existen muchas opciones para configurar y personalizar la herramienta a las necesidades concretas de una organización, además lleva incorporado un sistema de gestión de la configuración que permite definir líneas base. Existe un gran soporte para mantener la trazabilidad entre los documentos y mecanismos que facilitan la importación y exportación de la información en XML. (5)

Requisite Management (REM), esta es una herramienta experimental gratuita de Gestión de Requisitos diseñada para soportar la fase de Ingeniería de Requisitos de un proyecto de desarrollo software de acuerdo con la metodología definida en la Tesis Doctoral "Un Entorno Metodológico de Ingeniería de Requisitos para Sistemas de Información", realizada por el profesor Amador Durán del Departamento de Lenguajes y Sistemas Informáticos de la Universidad de Sevilla.

Características y funcionalidad básica

Los diversos módulos integrados en la herramienta son:

- Gestión de documentos de ingeniería de requisitos
- Trazabilidad entre documentos de trabajo
- Informes y estadísticas
- Generación de documento HTML

Algunas de las funcionalidades ofrecidas por la herramienta son:

Gestión de requisitos, diferenciando entre:

- Requisitos
- Funcionalidades
- Requisitos técnicos
- Casos de prueba

Trazabilidad entre todos los documentos de trabajo:

- Requisito-Requisito (control de versiones)
- Requisito-Requisito (dependencia entre requisitos)
- Requisito-Funcionalidad
- Requisito-Caso de Prueba
- Visualización de la matriz de trazabilidad
- Árbol de trazabilidad para facilitar las auditorías
- Gráfico de dependencias entre documentos de trabajo para poder determinar el impacto de un cambio

Esta herramienta tiene una interfaz muy intuitiva, por lo que la curva de aprendizaje es mínima debido a lo cual se recomienda como una muy buena herramienta para comenzar con el hábito de realizar la gestión de requisitos, ya que es muy sencilla, permite incluir la trazabilidad entre los requisitos y soporta dependencias entre estos.

Enterprise Architect (EA): es una herramienta flexible, completa y potente de modelado UML bajo la plataforma Windows. Provee lo más nuevo en desarrollo de sistemas, administración de proyectos y análisis de negocio. Esta herramienta abarca integralmente el ciclo de vida, cubriendo el desarrollo de software desde el levantamiento de los requisitos, a través de las etapas de análisis, modelos de diseño, testeo y finalmente el mantenimiento y re-uso. Es una herramienta utilizada para el desarrollo de varios tipos de software para un amplio rango de industrias, incluyendo: bancos, ingeniería, finanzas, medicina, investigación y muchas más. Enterprise Architect combina el poder de la última especificación UML 2.1 con alto rendimiento, interfaz intuitiva, para traer modelado avanzado al escritorio, y para el equipo completo de desarrollo e implementación. Con un gran conjunto de características y un valor sin igual para el dinero, EA puede equipar a su equipo entero, incluyendo analistas, evaluadores, administradores de proyectos, personal del control de calidad, equipo de desarrollo y más, por una fracción del costo de algunos productos competitivos. (6)

EA es una herramienta multi-usuario, diseñada para ayudar a construir software robustos y fáciles de mantener. Ofrece salida de documentación flexible y de alta calidad. El manual de usuario está disponible en línea.

Se escoge como herramienta para la gestión de requisitos OSRMT debido a que es open source y multiplataforma, características que la diferencian de EA y de REM.

1.8 Herramientas para el diseño de los prototipos de interfaz no funcionales

Las herramientas de diseño web brindan gran ayuda a los desarrolladores, haciéndoles más óptimo y fácil el trabajo, es por esto que para el desarrollo de un sitio web con calidad se debe realizar una correcta selección de la herramienta a utilizar. Entre estas: FrontPage, Macromedia Dreamweaver con todas sus versiones, entre otras.

Macromedia Dreamweaver 8 incluye las versiones más recientes de Dreamweaver, Flash, Fireworks y FreeHand (7). Permite el diseño de aplicaciones, presenta la capacidad de programar bajo lenguajes como: ASP, CSS, PHP, JSP y XML; lo que la convierte en una herramienta altamente compatible con las principales tecnologías de servidor, esto y el amplio desarrollo sobre sitios web, hace que mantenga en la sombra a herramientas como FrontPage. Soporta la mayoría de las tecnologías servidor y HTML

- Realiza el trabajo con capas
- Presenta una página de inicio para lograr un mejor acceso a documentos, recursos y archivos
- Brinda la posibilidad de crear botones Flash, formularios, JavaScripts y más, es de gran ayuda. Además de poder insertar elementos web, encontramos una gran precisión en la importación de información de Word y Excel, con las funciones de copiar y pegar. (8)
- JavaScript para la creación de efectos e interactividades
- Inserción de archivos multimedia

Es ampliable y abierto, cuenta con las tecnologías y productos que el usuario desea para tener una mayor libertad a la hora de elegir las tecnologías más convenientes. Permite crear el trabajo en corto tiempo al agilizar el flujo del mismo. Se puede actualizar tanto con componentes fabricados por Macromedia como con componentes desarrollados por otras compañías, lo cual permite la realización de acciones más avanzadas.

FrontPage es una herramienta para la edición de páginas web de Microsoft. Creado hace ya muchos años, ha tenido multitud de versiones que han ido mejorando su funcionamiento. Está orientado a personas inexpertas y sin conocimientos de HTML. Sus capacidades son semejantes a las de otros editores, como el crear mapas de imágenes, gestionar la arborescencia de las páginas del sitio, entre otras. Lamentablemente, al ser un producto Microsoft, está orientado a construir páginas optimizadas para Internet Explorer. Por esta misma razón, al insertar algún elemento activo en una página web, como es el caso de los controles ActiveX, o los scripts de cliente, sólo suele funcionar en Internet Explorer. Conseguir páginas que se vean bien en Netscape Navigator puede ser complicado con este programa, lo que es un serio inconveniente.

Por las características de las herramientas y basados en la experiencia que se ha tenido con el trabajo de las mismas, se decidió utilizar como herramienta para el diseño de los prototipos de interfaz no funcionales Macromedia Dreamweaver 8 por todas las facilidades que brinda para la creación de presentaciones web.

1.9 Patrones

Un patrón captura la esencia de una familia exitosa de soluciones probadas a un problema recurrente dentro de un cierto contexto. (9)

Un patrón es una especie de modelo que puede o no seguirse para solucionar algún problema o para realizar alguna actividad.

1.9.1 Patrones de Casos de Uso

Para el desarrollo de esta investigación se usarán patrones de casos de usos, los cuales son utilizados como herramientas o técnicas, obtenidas de la experiencia que poseen los desarrolladores del proyecto, facilitando la solución a los problemas que se presentan en la modelación de sistemas, lo que hace que se puedan obtener modelos con mayor calidad y de forma más rápida. Entre estos tipos de patrones de casos de uso se pueden encontrar:

- **Patrón CRUD** (*Creating, Reading, Updating and Deleting*): propone identificar un CU, llamado “Información CRUD” o “Administrar Información”, que modela todas las operaciones que se pueden realizar sobre una parte de la información de cierto tipo (o sea en una misma entidad), tal como crearla, leerla, actualizarla y eliminarla.
- **Múltiples actores:** Roles comunes (*Multiple Actors: Common Role*): consiste en que dos actores juegan el mismo papel hacia el caso de uso. Este rol es representado por otro actor, heredado por los actores que comparten este rol.

1.9.2 Patrones de arquitectura

Un patrón de arquitectura de software describe un problema particular y recurrente del diseño, que surge en un contexto específico, y presenta un esquema genérico y probado de su solución. (9)

Para el desarrollo de la aplicación se utilizará el framework Symfony el cual está basado en el patrón conocido como arquitectura Modelo-Vista-Controlador (MVC). Este patrón está formado por 3 niveles (Ver anexo 5):

- El modelo representa la información con la que trabaja la aplicación, es decir, su lógica de negocio.
- La vista transforma el modelo en una página web que permite al usuario interactuar con ella.
- El controlador se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista. (10)

1.9.3 Patrones de diseño

Un patrón de diseño es una descripción de clases y objetos comunicándose entre sí, adaptada para resolver un problema de diseño general en un contexto particular, identifica: clases, instancias, roles, colaboraciones y la distribución de responsabilidades.

La arquitectura MVC implementa internamente los patrones de asignación de responsabilidades GRASP.

”Los patrones GRASP describen los principios fundamentales de diseño de objetos para la asignación de responsabilidades. Constituyen un apoyo para la enseñanza que ayuda a entender el diseño de objeto esencial y aplica el razonamiento para el diseño de una forma sistemática, racional y explicable.” (11)

Los patrones GRASP son nueve, cinco principales y cuatro adicionales, de estos patrones se usan:
Principales:

- Bajo acoplamiento, se basa en la idea de tener las clases lo menos ligadas entre sí posible. De forma tal que en caso de producirse una modificación en alguna de ellas, la repercusión en el resto de clases sea mínima, potenciando así la reutilización, y disminuyendo la dependencia entre estas.
- Alta Cohesión, este patrón permite asignar una responsabilidad para mantener alta la cohesión, es decir, que cada elemento debe realizar una labor única en el sistema, la cual no debe ser desempeñada por ningún otro elemento del mismo y debe estar relacionada con la clase que la realiza.

- Experto, es el patrón más usado pues el asignar responsabilidades; es un principio básico que suele utilizarse en el diseño orientado a objetos. Con él no se pretende designar una idea oscura ni extraña; expresa simplemente la “intuición” de que los objetos hacen cosas relacionadas con la información que poseen.
- Creador, propone asignar la responsabilidad de crear nuevos objetos de una clase a otra si esta última: contiene la información necesaria para realizar la creación del objeto, o si usa directamente las instancias creadas del objeto o si almacena o maneja varias instancias de la clase.
- Controlador, este patrón sirve de intermediario entre una interfaz y el algoritmo que la implementa, siendo el que recibe los datos del usuario y los envía a las distintas clases según el método llamado.

Adicionales:

- Fabricación pura, este patrón se basa en la problemática de ¿cómo proceder cuando las soluciones encontradas comprometen la cohesión y el acoplamiento? es decir, ¿Qué objetos deberían tener la responsabilidad cuando no se quiere violar los objetivos de los patrones de alta cohesión y bajo acoplamiento, u otros, o cuando las soluciones que ofrece el experto, por ejemplo, no son adecuadas? Propone como solución asignar un grupo de responsabilidades altamente cohesivas a una clase artificial o de conveniencia, sin que esta represente ningún concepto del dominio del problema. Tal clase es una *fabricación* de la imaginación e idealmente las responsabilidades asignadas a esta fabricación soportan alta cohesión y bajo acoplamiento, de manera que el diseño de la fabricación es muy limpio o *puro*.

Los patrones del grupo GoF se clasifican en: creacionales, estructurales y de comportamiento se hará uso de:

Creacionales:

Singleton, el cual provee una única instancia global gracias a que:

- La propia clase es responsable de crear la única instancia.
- Permite el acceso global a dicha instancia mediante un método de clase.
- Declara el constructor de clase como privado para que no sea instanciable directamente.

Estructurales:

Decorator, añade objetos individuales de forma dinámica y transparente siendo aplicable cuando:

- La responsabilidades de un objeto pueden ser retiradas.
- La extensión mediante la herencia no es viable.
- Hay una necesidad de extender la funcionalidad de una clase, pero sin poder hacerlo mediante la herencia.
- Hay la necesidad de extender dinámicamente la funcionalidad de un objeto.

Facade, sirve para proveer de una interfaz única y sencilla que haga de intermediaria entre un cliente y una interfaz o grupo de interfaces más complejas, además puede:

- Hacer una biblioteca de software más fácil de usar y entender, pues implementa métodos convenientes para tareas comunes.
- Hacer el código que usa la librería más legible.
- Reducir la dependencia de código externo en los trabajos internos de una librería.

Composite, permite la construcción de objetos complejos a partir de otros más simples y similares entre sí, debido a la composición recursiva y a la estructura en forma de árbol, lo que hace más simple el tratamiento de los objetos creados, ya que al poseer todos ellos una interfaz común, se tratan de la misma manera.

Comportamiento:

Chain of Responsibility, permite establecer una cadena de objetos receptores a través de los cuales se pasa una petición formulada por un objeto emisor. Cualquiera de los objetos receptores puede responder a la petición en función de un criterio establecido.

1.10 Conclusiones

En este capítulo se realiza un análisis de las tendencias actuales, proponiéndose las soluciones que más se ajustan al caso de estudio. Se explican los objetivos generales y específicos del problema. Además se muestran algunas de las metodologías y tecnologías utilizadas para la elaboración de los artefactos a realizar por el rol del analista y el diseñador. Luego de la realización del mismo se arribó a las siguientes conclusiones:

- Analizar y diseñar una aplicación Web que facilite la gestión de la información en la planificación y control del menú en la UCI.
- Utilizar RUP como metodología de desarrollo para guiar el proceso de modelación, análisis y diseño de la aplicación que se propone.

- Utilizar Visual Paradigm como herramienta CASE.
- Utilizar Macromedia Dreamweaver 8 como herramienta para el diseño de los prototipos de interfaz no funcionales.

Capítulo 2: Características del sistema.

Introducción

En este capítulo se hace una definición del objeto de estudio del problema, se plantean los objetivos estratégicos de la organización y procesos de negocio que los soportan. Descripción de los procesos que serán objeto de automatización. Descripción general de la propuesta de sistema, como debe funcionar. Definición del Modelo de Negocio. Especificación de los requisitos y CU del sistema.

2.1 Objeto de Estudio

2.1.1 Objetivos estratégicos de la organización

La VRL atiende varias áreas dentro de la universidad, dentro de ellas está la DGA la cual a su vez controla otras áreas: Complejos de Comedores, Pantris, Panadería, Dulcería, Cocina Comedor de IP (CCIP) y CCCLU (Ver anexo 3).

Centralmente se lleva a cabo la planificación y control de una serie de actividades asociadas al servicio de comedores. Actualmente estas actividades no se ejecutan de forma correcta lo que trae como consecuencia la pérdida de alimentos, proporcionando grandes pérdidas a la universidad tanto económicas como de recursos. La actividad que más se ve afectada en este sentido es la confección del menú debido a que el mismo se planifica para cada servicio (desayuno, almuerzo, comida y merienda, los cuales además pueden estar incluidos en un menú para la playa, un menú de protocolo, un menú para una dieta, un menú semanal, para un mes, es decir para cualquier período que se solicite) muchas veces no cumple con los indicadores requeridos, además de estar sujeto a constantes cambios pues no se tiene un control de lo que realmente hay en el almacén y se invierte mucho tiempo en su confección, por no estar la información necesaria centralizada y no existir una integración entre el Planificador del menú y ATM.

Precisamente el objeto de estudio de esta investigación está enmarcado en los procesos de gestión de la información del servicio de comedores y el campo de acción se corresponde con los procesos gestión de la información en el proceso de planificación y control del menú en la UCI.

La razón por la cual se propone realizar el análisis y diseño de un sistema que permita la gestión de la información en el proceso de planificación y control del menú en la UCI es precisamente para lograr que la DGA pueda “garantizar la oferta de platos que conforman la alimentación de los estudiantes y trabajadores del centro, becados y externos, en regímenes especiales de trabajo y estudio teniendo en

cuenta un óptimo balance nutricional sin pérdidas económicas y mejorando continuamente la variedad, la atención a los comensales y la calidad del servicio en general".(12)

2.1.2 Descripción de los procesos de negocio actuales

En la VRL, específicamente en la DGA y en ATM, se lleva a cabo el proceso de planificación y control del servicio de comedores, y dentro de este una serie de actividades como:

- Confeccionar el menú a partir de las existencias en ATM.
- Calcular la cantidad de productos a utilizar.
- Realizar el ajuste de la planificación diaria.
- Planificar comensales.

El desarrollo de dichas actividades está a cargo de un personal capacitado dentro del área con el objetivo de optimizar la ejecución de las mismas así como reducir el gasto innecesario de materiales, debido a esto es importante controlar eficientemente los recursos y aprovecharlos al máximo. Se necesita que se haga una buena planificación para eliminar la inconformidad que actualmente existe entre los clientes de estas áreas y que por supuesto, contribuya a reducir la pérdida de alimentos.

El proceso de planificación y control del menú comienza con la elaboración del documento Asignación de alimentos para el próximo período, este proceso se desarrolla tres veces al año y los productos que estén contenidos en este documento son los que se asignarán por el Consejo de Estado (CE) a los almacenes de la universidad.

Luego se realiza la confección del menú, es importante aclarar que existen diferentes tipos de menú, como son: menú para la playa, menú de protocolo, menú para una dieta, menú semanal, en caso de que el menú que se esté planificando es para más de un día se confecciona la Demanda de materiales. En esta actividad se tienen en cuenta las existencias en ATM y los productos de próximo arribo, siendo estos los productos que deben arribar a los almacenes de ATM antes de que se produzca el menú planificado, es decir, se consulta el Modelo de Existencias en ATM y el Plan de Carga para conformar el Listado de Productos Disponibles. Luego de haber elaborado este documento el cual es necesario para la confección del menú se planifica una propuesta del mismo, teniendo en cuenta que esta debe cumplir con los indicadores requeridos (Balance Nutricional, Normas de Consumo y Frecuencia Establecida) tratando de lograr además que haya variedad en los platos que se ofertan en los servicios (desayuno, almuerzo, comida y merienda) y posteriormente se procede a planificar los comensales para ese menú.

El menú diariamente antes de ser elaborado se revisa y se ajusta en caso de ser necesario, pues puede ocurrir que alguno de los productos que forman parte de algún plato planificado para ese menú no esté disponible en los almacenes y es necesario sustituirlo por otro, este proceso trae consigo la

creación de tres documentos que se confeccionan diariamente luego de haberse realizado el reajuste, estos son: Orden de producción, Solicitud de materiales y Vale de salida.

Luego de haber sido ofertado el menú del día se realiza un reporte del comportamiento de los servicios que fueron ofertados, este reporte representa el proceso de control de los comedores, el cual permite verificar la cantidad de alimentos que se elaboraron y la cantidad de comensales reales que pasaron, archivándose además el menú ofertado como menú real. Una vez realizado este reporte se realiza el balance nutricional real de los servicios ofertados en ese día y se verifica el cumplimiento de las normas. Todos los indicadores se archivan como documentos reales, ejemplo: menú real, balance nutricional real, frecuencia establecida real.

2.1.3 Análisis crítico de los procesos

La UCI cuenta con tres complejos de comedores, los que están compuestos por tres o cuatro comedores a su vez, pues el número de personas que disfruta de los servicios que estos brindan es muy elevado. Hoy en día estos servicios no se realizan de forma eficiente debido a que el proceso de planificación y control que aquí se lleva a cabo se realiza manualmente por lo que la ocurrencia de errores es mayor; además, la información no está centralizada y en ocasiones ni siquiera se registra, lo que trae como consecuencia que no exista la posibilidad de obtener indicadores estadísticos de los procesos. Estos problemas planteados originan una situación en el área que atenta contra el correcto funcionamiento económico de la universidad.

El proceso de planificación del menú es la actividad más crítica que actualmente se desarrolla en la DGA, pues es el resultado de una larga jornada de trabajo consultando documentos: Modelo de Existencia en ATM y Plan de Carga, para posteriormente elaborar el Listado de Productos disponibles, el menú que se planifica en escasas ocasiones se llega a convertir en el menú real que se oferta, esto es producto a la falta de integración que existe entre el Planificador del menú y ATM por lo cual no se tiene dominio de lo que realmente está en existencia en el almacén. Es por ello que cuando se hace el ajuste de la planificación diaria del menú este está sujeto a una serie de modificaciones.

A pesar de que en la actualidad existen herramientas que contribuyen con los procesos de planificación y control la universidad no puede adquirirlas, debido a que muchas de las empresas desarrolladoras de estos software imponen altos costos de licencia por su utilización y en algunos casos se prohíbe a entidades nacionales adquirirlos debido al injusto bloqueo que el gobierno de Estados Unidos le impone al país; por otra parte también existen limitantes en cuanto al almacenamiento y gestión de los datos, por no contar con un sistema capaz de tales prestaciones,

debido al gran flujo de información que exigen los comedores de la UCI (tal es el caso de la herramienta empleada en la ISPJAE).

Estos problemas planteados atentan contra el correcto funcionamiento de la DGA pues la probabilidad de controlar los recursos y la pérdida de alimentos es cada vez más difícil y sin posibilidad de hacerlo de una forma rápida y eficiente.

2.2 Modelo del negocio actual

Para comprender claramente todo negocio es necesario modelar los procesos que se llevan a cabo en el mismo. Los requerimientos más importantes que puede tener el sistema se obtienen luego de haber elaborado el modelo del negocio, con este modelo se pueden organizar y presentar los procesos más detallados, los cuales se desarrollan en la planificación y control del menú en la UCI.

2.2.1 Reglas del Negocio

Las reglas del negocio describen políticas que deben cumplirse o condiciones que deben satisfacerse, por lo que regulan algún aspecto del negocio. A continuación se mencionan las identificadas para el presente negocio:

- No se puede planificar un plato si no está aprobado por el Director General de Alimentos.
- No se puede planificar un plato si no tiene existencias, o plan de carga que cubra los comensales planificados.
- No se puede confirmar un menú en el reajuste diario si alguno de sus platos no tiene existencias para los comensales planificados.
- No se puede planificar un plato que exista en una planificación, el mismo día, en el mismo servicio, en el mismo complejo de producción y servicios.
- La confirmación de las planificaciones (comensales y menú) y los reportes diarios son realizados por el Director General de Alimentos.
- No se puede realizar el reporte diario, si no se ha confirmado la planificación del menú y comensales para ese complejo de producción y servicios en el mismo día. Además, los platos que se ofertaron, tienen que existir en la planificación de ese complejo, en el servicio y día especificado.
- No se debe realizar el Balance Nutricional de un complejo de producción y servicios sino existen los cuatro indicadores nutricionales para el complejo. Además, las planificaciones y los reportes existentes para ese complejo en el período seleccionado, tienen que estar confirmados.

- Si se desea crear un plato cuyo nombre coincide con uno existente, el nuevo plato a entrar se introduce con el nombre y entre paréntesis la diferencia con el plato anterior.
- Las unidades de medidas de los productos en las normas de consumo deben ser las mismas utilizadas para la confección de los platos. (12)

2.2.2 Actores y trabajadores del negocio

Actores del Negocio	Descripción
Cliente	Son las personas dentro de la universidad que solicitan un servicio al Director General de Alimentos, estas pueden ser: directivos (FEU, UJC, trabajadores).
Reloj	Representa un ente que inicia las actividades que se desarrollan con cierta periodicidad dentro de la DGA de la UCI.
Grupo de Apoyo del Consejo de Estado	Representa un grupo de personas que trabajan en el CE y que son las encargadas de aprobar o no la Asignación de alimentos para el próximo período elaborado por el Planificador del menú, el Director General de Alimentos, el Representante de Compras y el Vicerrector de Logística.

Tabla 1 Actores del Negocio

Trabajadores del Negocio	Descripción
Director General de Alimentos	Es una persona que trabaja en la UCI encargada de aprobar la solicitud de un servicio presentada por el cliente, de orientar al Planificador del menú la confección de un menú y luego realiza la aprobación de este.
Planificador del menú	Es el responsable de confeccionar el menú, de planificar y ajustar el menú diariamente y de elaborar junto al el Director de General de Alimentos, el Representante de Compras y el

	Vicerrector de Logística el documento de Asignación de alimentos para el próximo período.
Representante de Compras	Representa a un grupo de personas encargadas de las compras por productos ellos son: Grupo de Compras de viandas, Grupo de Compras de productos cárnicos, Grupo de Compras de productos secos y Grupo de Compras de vegetales.

Tabla 2 Trabajadores del Negocio

2.2.3 Modelo de Casos de Uso del Negocio

El diagrama de casos de uso (CU) del negocio permite representar gráficamente a los procesos del negocio y la interacción de estos con los actores del mismo.

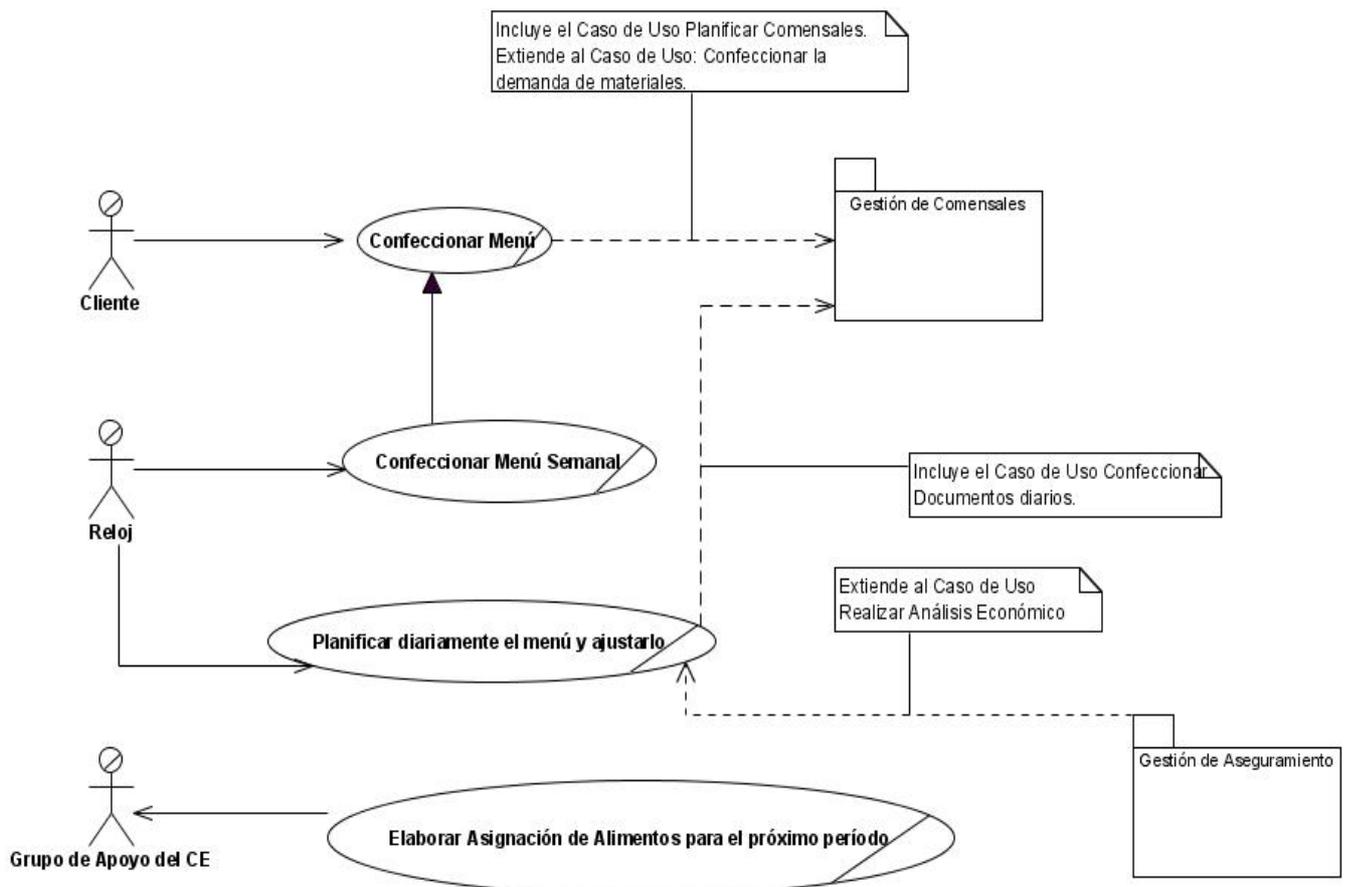


Fig. 1 Diagrama de CU del Negocio

2.2.4 Descripciones de los CU del negocio

Para poder realizar el modelo de los procesos del negocio que están relacionados con el campo de acción es necesario centrar la atención en los procesos de Confección del menú, en la elaboración del documento Asignación de alimentos para el próximo período, en la Planificación diaria del menú y ajuste. El primer paso para la modelación del negocio actual propuesto consiste en la identificación de los procesos que se desarrollan.

Caso de Uso:	Confeccionar menú
Actores:	Cliente
Trabajadores:	Planificador del Menú, Jefe de Almacén, Jefe de Compras, Director General de Alimentos.
Resumen:	Este proceso se puede realizar en cualquier momento, es decir, el Cliente puede solicitar un servicio (ya sea a un menú para actividades de la FEU, un menú de protocolo, etc.) al Director General de Alimentos cuando lo necesite, luego este, si aprueba la solicitud, le informa al Planificador del Menú que debe confeccionar el menú para dicha solicitud. Este proceso tiene una estrecha relación con el proceso de Planificar Comensales puesto que siempre que se confecciona un menú se Planifican los comensales para el mismo, con el objetivo de verificar que dicho menú cubra los comensales existentes. Una vez que este esté confeccionado se necesita la aprobación del Director de Alimentos para poder archivar el menú.
Precondiciones:	

Tabla 3 Descripción textual del CU Confeccionar Menú

Caso de Uso:	Planificar diariamente el menú y ajustarlo.
Actores:	Reloj
Trabajadores:	Planificador del menú, Jefe de Almacén, Director General de alimento.
Resumen:	Este proceso se realiza todos los días, bien temprano en la mañana en cuanto llegan los trabajadores a la Universidad. El

	proceso comienza a partir de que el Jefe de Almacén le envía al Planificador del menú el Modelo de Existencia en ATM. Luego que el planificador del menú recibe el modelo, revisa que el menú para ese día no deba sufrir cambios debido a que los almacenes no tengan existencias de productos para cubrir el menú planificado para ese día, en caso contrario lo modifica y se lo envía al Director General de Alimentos para su aprobación.
Precondiciones:	Que existan los documentos Menú, Modelo de Existencia en ATM y la Planificación de Comensales.

Tabla 4 Descripción textual del CU Planificar diariamente el menú y ajustarlo

Caso de Uso:	Elaborar la Asignación de alimentos para el próximo período.
Actores:	Grupo de Apoyo del Consejo de Estado (CE).
Trabajadores:	Encargado de Elaborar Documento (Planificador del menú, Jefe del departamento de Compras, Director de Alimentos, Vicerrector de Logística), Rector.
Resumen:	Este proceso se realiza tres veces en el año, donde el Encargado de Elaborar Documento tomando como referencia las asignaciones en el período anterior y el Listado de Productos que da el Consejo de Estado elabora una propuesta para la Asignación de Alimentos para el próximo período, la cual será aprobada o no por el Rector.
Precondiciones:	

Tabla 5 Descripción textual del CU Elaborar Asignación de Alimentos para el próximo período

Caso de Uso:	Confeccionar el Menú Semanal.
Actores:	Reloj.
Trabajadores:	Planificador del menú, Jefe de Almacén, Jefe de Compra, Director General de Alimentos.
Resumen:	Este proceso se realiza un día determinado de la semana. El Planificador del menú elabora una propuesta para el menú de la próxima semana y se lo envía al Director General de Alimentos para ser aprobada, en caso que este no aprueba la propuesta

	se la envía nuevamente al Planificador del menú para que la modifique.
Precondiciones:	

Tabla 6 Descripción textual del CU Confeccionar el Menú Semanal

Para ver los diagramas de actividades y el flujo normal de los eventos de cada proceso, ver el expediente de proyecto, el cual se encuentra ubicado en: \\10.34.18.228\Expediente de Proyecto\Expediente del proyecto\Alimentos UCI\Gestión del Menú.

2.2.5 Modelo de Objetos del Negocio

El modelo de objetos del negocio representa los trabajadores del negocio y su interacción con las entidades del mismo en el flujo de trabajo Modelado de Negocio.

Debido a la cantidad de documentos que son necesarios consultar en el proceso de gestión del menú, se decidió para lograr una mayor claridad del modelo de objetos del negocio confeccionarlo para cada CU.

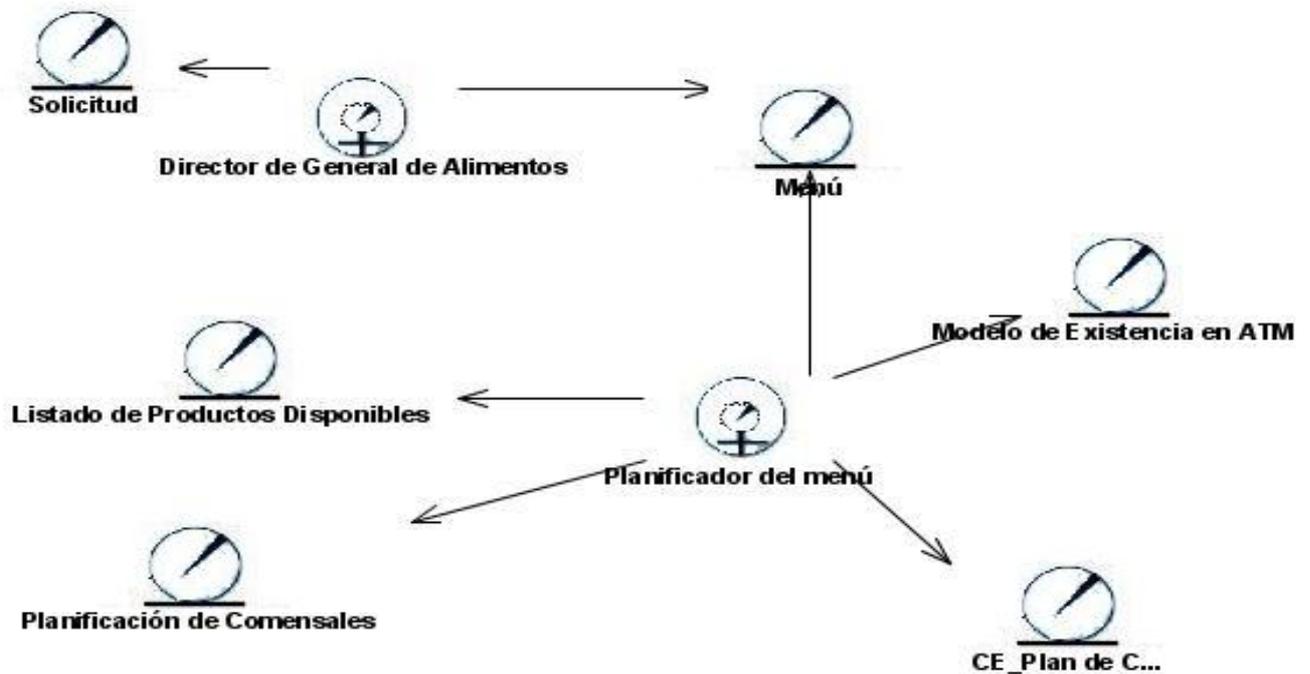


Fig. 2 Modelo de Objetos del CU Confeccionar Menú

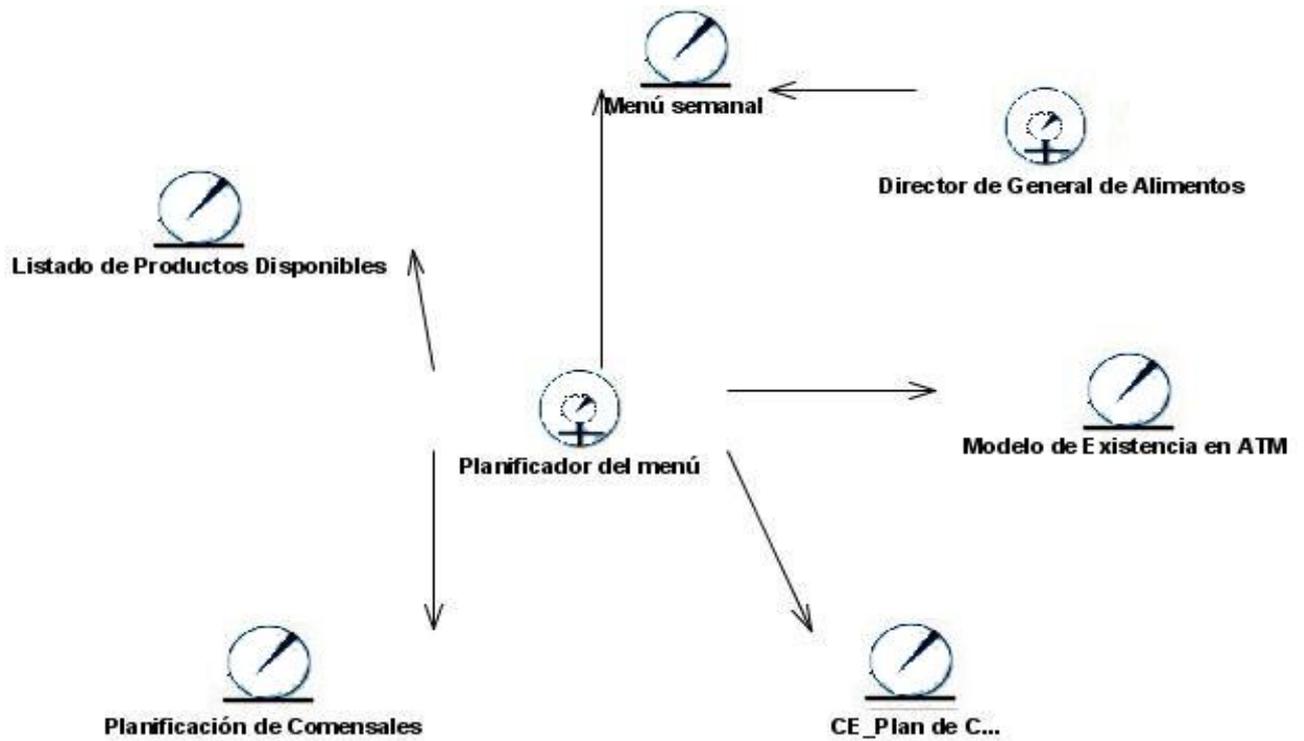


Fig. 3 Modelo de Objetos del CU Confeccionar Menú semanal

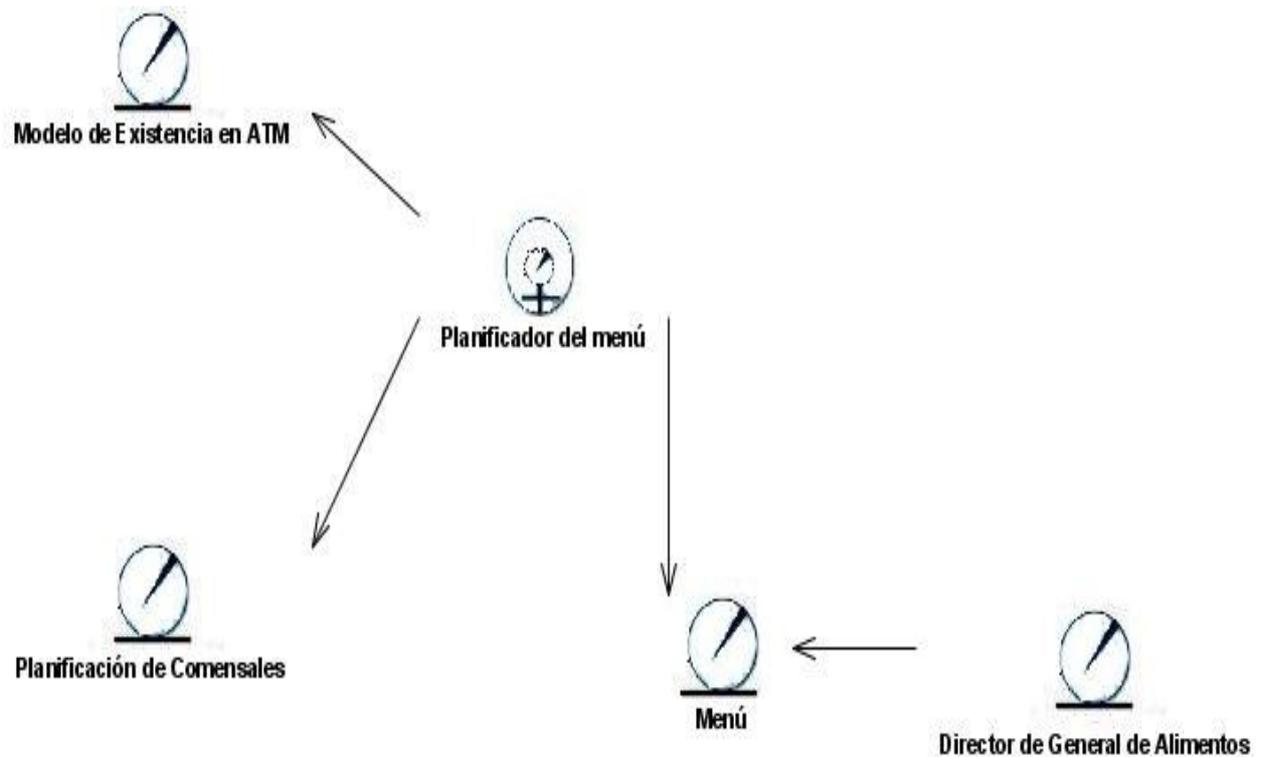


Fig. 4 Modelo de Objetos del CU Planificar diariamente el menú y ajustarlo

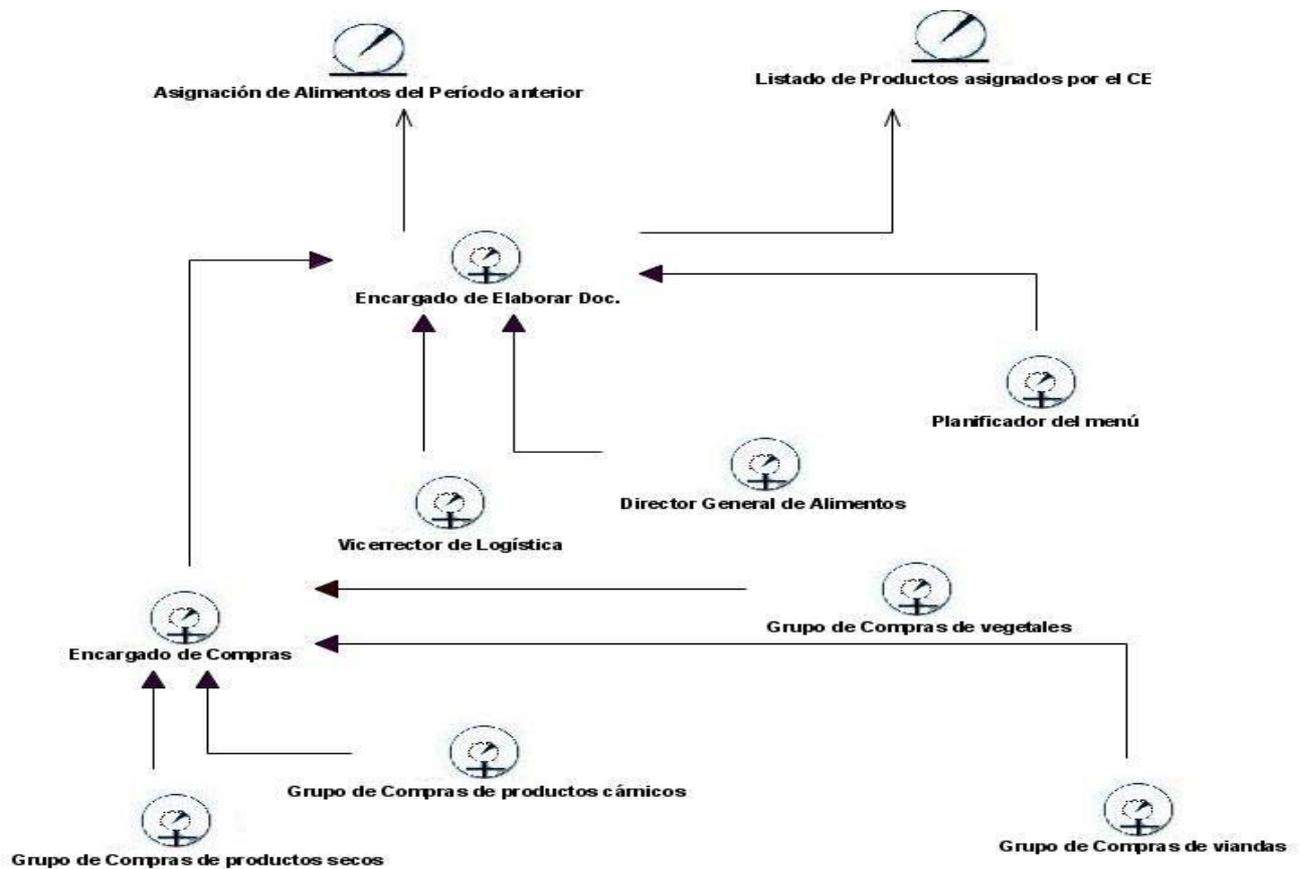


Fig. 5 Modelo de Objetos del CU Elaborar Asignación de alimentos para el próximo período

Luego de haber modelado el proceso del negocio actual se procede al Modelado del Sistema con el objetivo de definir las funcionalidades, características y los procesos fundamentales del mismo.

2.3 Modelado del Sistema

2.3.1 Requerimientos Funcionales y no Funcionales

Los requisitos constituyen la base del modelado de un sistema, durante el proceso de desarrollo de software. De hecho, es la aparición de unas determinadas necesidades lo que motiva el desarrollo de un sistema o aplicación, el cual tiene como fin solucionar esas necesidades o deficiencias detectadas.

Requerimientos funcionales

Los requerimientos funcionales representan funcionalidades que el sistema debe cumplir.

RF1- Mostrar listado de productos

RF2- Imprimir listado de productos

- RF 3- Buscar un producto
- RF 4- Crear un nuevo plato
- RF 5- Modificar plato
- RF 6- Mostrar listado de platos
- RF 7- Imprimir listado de platos
- RF 8- Eliminar plato
- RF 9- Buscar un plato
- RF 10- Calcular aporte en cuanto a carbohidratos
- RF 11- Calcular aporte en cuanto a grasas
- RF 12- Calcular aporte en cuanto a proteínas
- RF 13- Calcular aporte en cuanto a energía
- RF 14- Aprobar plato
- RF 15- Crear una Asignación de alimentos para el próximo período
- RF 16- Calcular la cantidad de cada producto por evento
- RF 17- Calcular la cantidad de cada producto por período
- RF 18- Calcular el Balance Nutricional de cada producto
- RF 19- Calcular la frecuencia establecida para cada producto por semana
- RF 20- Mostrar Asignación de alimentos para el próximo período
- RF 21- Imprimir Asignación de alimentos para el próximo período
- RF 22- Modificar Asignación de alimentos para el próximo período
- RF 23- Aprobar Asignación de alimentos para el próximo período
- RF 24- Crear una nueva solicitud de menú
- RF 25- Aprobar solicitud de menú
- RF 26- Mostrar existencias a partir del Modelo de Existencia en ATM
- RF 27- Mostrar Cargas
- RF 28- Imprimir listado de cargas
- RF 29- Crear Carga
- RF 30- Modificar Carga
- RF 31- Buscar Carga
- RF 32- Confirmar carga
- RF 33- Mostrar Plan de Carga
- RF 34- Imprimir Plan de Carga
- RF 35- Crear Plan de Carga
- RF 36- Modificar Plan de Carga

- RF 37- Confirmar Plan de Carga
- RF 38- Crear un nuevo menú
- RF 39- Verificar Balance Nutricional
- RF 40- Mostrar menú
- RF 41- Imprimir Menú
- RF 42- Modificar menú
- RF 43- Buscar menú
- RF 44- Reajustar menú diario
- RF 45- Confirmar menú
- RF 46- Mostrar normas nutricionales
- RF 47- Imprimir normas nutricionales
- RF 48- Modificar normas nutricionales
- RF 49- Adicionar normas nutricionales
- RF 50- Mostrar valor nutricional
- RF 51- Imprimir valor nutricional
- RF 52- Modificar valor nutricional
- RF 53- Adicionar valor nutricional

Requerimientos no funcionales del sistema

Los requerimientos no funcionales son propiedades o cualidades que el sistema debe tener.

1. Requerimiento de Software

Para la instalación del sistema se debe disponer del Sistema Operativo Windows 98 o superior, o cualquier distribución de Linux. Con el objetivo de lograr que las interfaces Web puedan visualizarse, las computadoras clientes usarán uno de los siguientes navegadores: Internet Explorer 5.5 o superior, Netscape, Mozilla 1.7 o superior o FireFox 0.9.3 o superior.

Se debe instalar un servidor Web Apache 1.3 o superior.

2. Requerimiento de Hardware

Para el correcto funcionamiento de la aplicación es necesario que haya conectividad.

La máquina que se usará como servidor Web debe tener alta disponibilidad además de un rendimiento adecuado. Este servidor debe presentar las siguientes características: Procesador Pentium IV superior

a 2.0 GHz, 512 MB de memoria RAM (incluye la utilizada por el Sistema Operativo) y como mínimo 80.0 GB de capacidad en disco duro.

Las computadoras clientes deben contar con al menos una impresora para garantizar la impresión de los documentos necesarios sin ninguna dificultad y deben tener como mínimo un procesador Pentium IV superior a 1.0 GHz, 256 MB de memoria RAM. Estas máquinas deben de estar conectadas en red con el servidor.

3. Requerimiento de Apariencia o Interfaz externa

Las páginas no se cargarán con mucha información, poseerán las imágenes necesarias además de colores claros, correspondiéndose a los estándares establecidos para lograr un buen diseño. Se contará en la página principal con la información necesaria para guiar al usuario y con una interfaz diferente para cada tipo de usuario que se registre.

4. Requerimiento de Usabilidad

La aplicación será utilizada por el personal de la VRL de la UCI específicamente por los que laboren en la DGA, ATM y las diferentes unidades de producción y servicios.

Se impartirá una preparación a los usuarios con la explicación de cómo se realizará el trabajo con la aplicación y la misma contendrá un manual de usuario que será usado como ayuda para su desempeño.

5. Requerimiento de Soporte

Una vez que la aplicación esté terminada se instalará en la UCI para realizar las pruebas necesarias con el objetivo de comprobar el correcto funcionamiento de la misma.

Cada cierto tiempo se realizará por los administradores del sistema un mantenimiento y en caso de ser necesario se realizarán otras versiones para mejorar la aplicación diseñada.

6. Requerimiento de Seguridad

Para acceder a cualquiera de las páginas de la aplicación los usuarios deben ser sometidos a la autenticación, especificando usuario y contraseña de los mismos.

Cada usuario tendrá permisos de acceso a las páginas de la aplicación de acuerdo con la responsabilidad de los mismos.

Cada usuario tendrá definido un solo rol para el manejo de la aplicación, únicamente en caso extremo se le asignará más de un rol a un usuario.

La contraseña de los usuarios debe de tener un mínimo de 7 caracteres y debe tener una fortaleza media.

Los usuarios deberán cambiar su contraseña cada 45 días como máximo, debido a lo cual 5 días antes de que se cumpla el plazo máximo serán avisados.

Las comunicaciones entre los clientes y el servidor serán seguras.

7. Restricciones en el Diseño y la Implementación

El análisis y el diseño de la aplicación se harán bajo las condiciones que impone la Metodología RUP utilizando como lenguaje de modelado a UML. Como herramienta CASE se utilizará Visual Paradigm 6.0 para realizar el modelado de los artefactos que se obtienen como consecuencia de los flujos de trabajo que define RUP y como herramienta para la confección de los prototipos no funcionales del sistema se utilizará Macromedia Dreamweaver 8. Además se utilizará a Symfony como framework y debido a que este utiliza como lenguaje de programación PHP 5 será este el lenguaje que se utilice para la implementación Web de la aplicación, JavaScript para la programación de escritorio y PostgreSQL como gestor de Base de Datos.

8. Requerimiento de Rendimiento

A pesar de que no es necesario que las solicitudes de los clientes tengan una velocidad de respuesta como los sistemas de tiempo real, sí es necesario garantizar la rapidez de respuesta del sistema. Debido a que la aplicación está concebida para un ambiente Cliente/Servidor, debe garantizarse que los tiempos de respuestas sean generalmente rápidos al igual que la velocidad de procesamiento de la información.

9. Requerimiento de Portabilidad

La aplicación podrá ser usada sobre los sistemas operativos Windows y Linux.

10. Requerimiento Políticos-Culturales

En la aplicación se usará el idioma español. Las imágenes que se usarán estarán relacionadas con el tema de estudio. En caso de ser necesario realizar algún cambio en la aplicación este será tramitado por la VRL o la DGA.

11. Requerimientos de extensibilidad

Se debe lograr un diseño adaptable, con la capacidad de poder soportar funcionalidades adicionales o modificar las funcionalidades existentes sin afectar a los restantes requerimientos contemplados en el sistema.

2.3.2 CU del Sistema

1. Gestionar productos
2. Gestionar plato
3. Hallar características del plato
4. Aprobar plato
5. Gestionar Asignación de alimentos para el próximo período
6. Crear solicitud de menú
7. Aprobar solicitud de menú
8. Mostrar Existencias
9. Gestionar Plan de Carga
10. Gestionar Carga
11. Gestionar menú
12. Confirmar menú
13. Gestionar Normas Nutricionales
14. Gestionar Valor Nutricional

2.3.3 Actores del Sistema

Actores	Descripción
Planificador del menú	Es el que elabora la Asignación de alimentos para el próximo período. Es el encargado, además, de la confección del menú y es el que gestiona toda la información referente a los platos.
Gestor_Productos	Es una generalización del rol de Jefe de Almacén y Jefe de Compras, es el encargado de gestionar la información referente a los productos.
Director General de Alimentos	Es el encargado de aprobar el plato que se cree. También se encarga de confirmar el menú planificado. Es el que gestiona la información referente a las Normas y Valores Nutricionales establecidos por el CE.
Jefe de Compras	Es el encargado de gestionar la información referente a las Cargas y al Plan de Carga.
Jefe de Almacén	Es el encargado de mostrar las existencias en ATM y de gestionar la información referente a los productos.
Gestor_Existencias	Es una generalización del rol de Planificador del menú y del Jefe de Almacén, es el encargado de gestionar las existencias en Almacén.
Cliente	Es el encargado de hacer una solicitud para un menú, este cliente puede ser (un presidente de la FEU de las facultades, algún decano, etc.).

Tabla 7 Actores del sistema

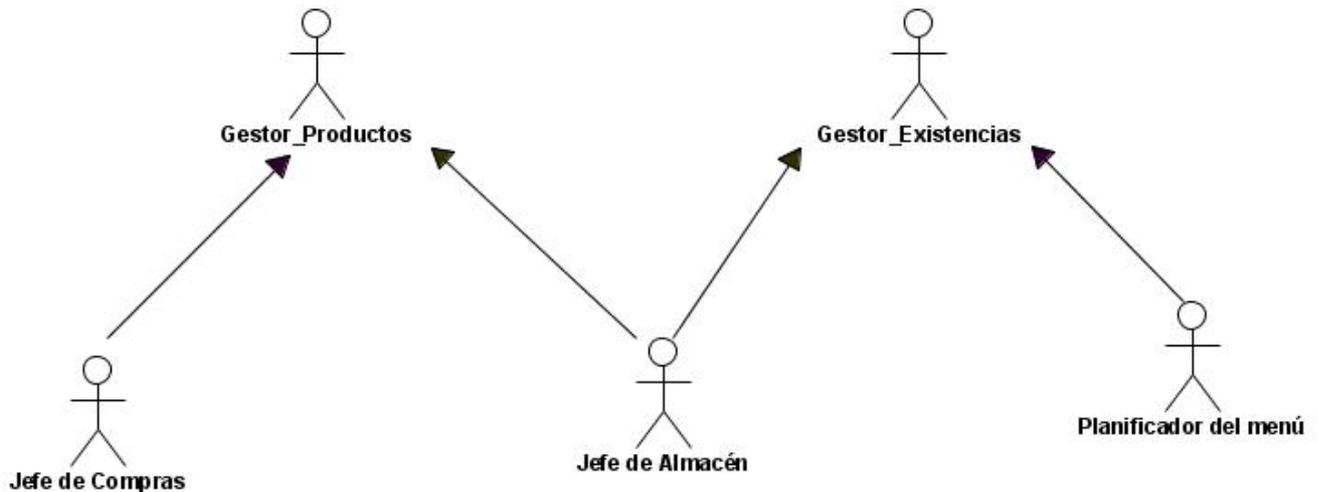


Fig. 6 Diagrama de Generalización de actores del sistema

2.3.4 Matriz de Trazabilidad

Para poder lograr una exitosa gestión de los requisitos de un software es necesario hacer una buena trazabilidad de los mismos, pues esta permite conocer los elementos que se afectan al ocurrir algún cambio en algún otro elemento que tenga relación con el primero. Este trabajo se hace mucho más sencillo con el uso de la matriz de trazabilidad.

La matriz de trazabilidad es una herramienta que ayuda a seguir, ordenar y mostrar como realizamos la validación de un sistema partiendo de los requerimientos de usuarios y mostrando en cada etapa de calificación como verificamos los mismos (14).

La Matriz de trazabilidad nos brinda numerosos beneficios, dentro de los que se destacan:

- Facilita la administración y seguimiento de los ítems críticos del análisis de riesgo.
- Permite una mejor visualización del alcance de la calificación y de los test.
- Ayuda a demostrar que la validación se encuentra completa.
- Ayuda a la administración de cambios y la visualización del impacto de los mismos en la calificación.
- Posee un gran valor agregado en el momento de las auditorias y las inspecciones mostrando a través de la misma un mapa general de la validación. (14)

Para ver la Matriz de Trazabilidad remitirse al expediente de proyecto.

2.3.5 Diagrama de CU del Sistema

Los diagramas de CU del sistema muestran la interacción o comunicación que existe entre los CU de un sistema y los actores del mismo.

Los CU pueden ser asignados a un paquete en concreto. Estos paquetes permiten localizar los cambios en un proceso del negocio, en el comportamiento de un actor y en un conjunto de CU estrechamente relacionados (13). Existen tres criterios para el empaquetamiento de los CU, de los cuales se escogió para el desarrollo de la aplicación:

- El criterio de los CU para dar soporte a un determinado actor del sistema.

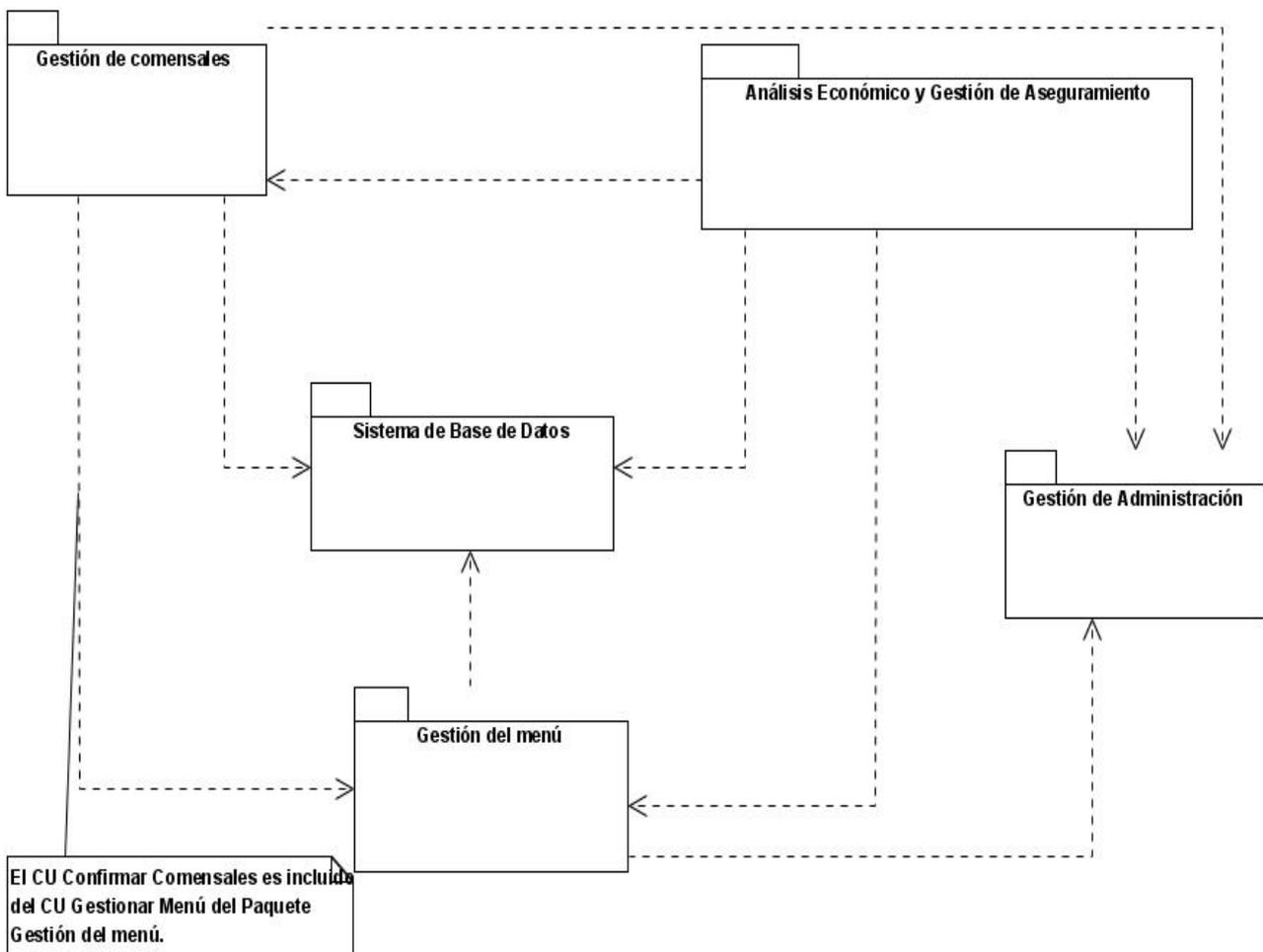


Fig. 7 Diagrama de Paquetes del sistema



Fig. 8 Diagrama de CU del Paquete Cliente

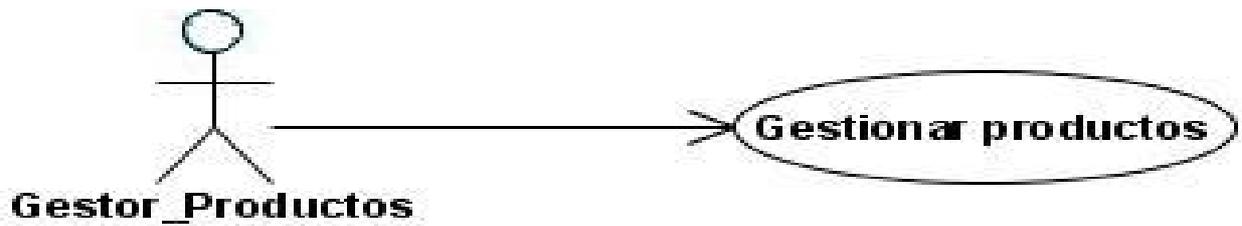


Fig. 9 Diagrama de CU del Paquete Gestor_Productos

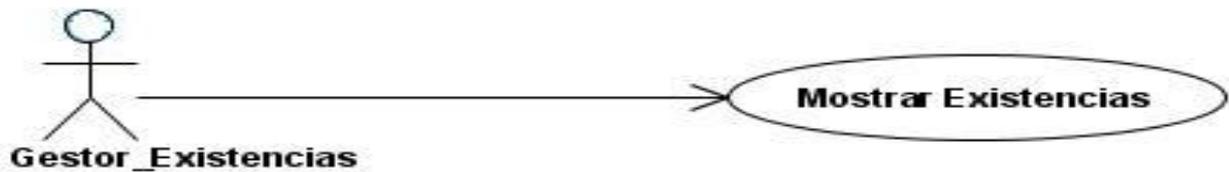


Fig. 10 Diagrama de CU del Paquete Gestor_Existencias

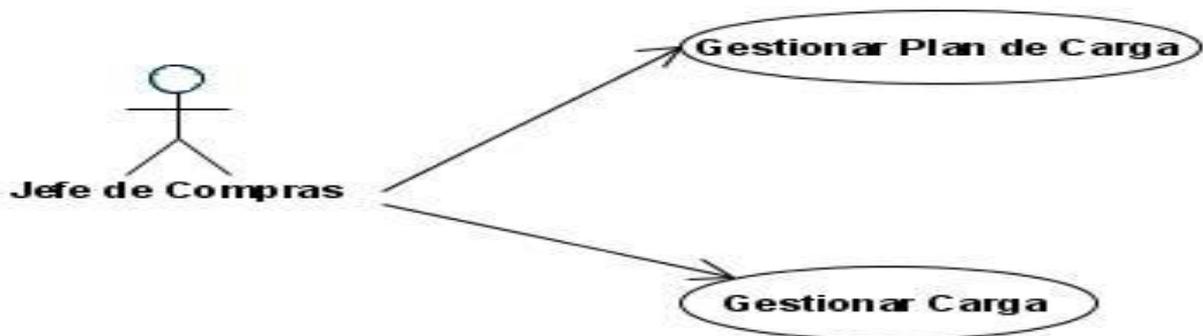


Fig. 11 Diagrama de CU del Paquete Jefe de Compras

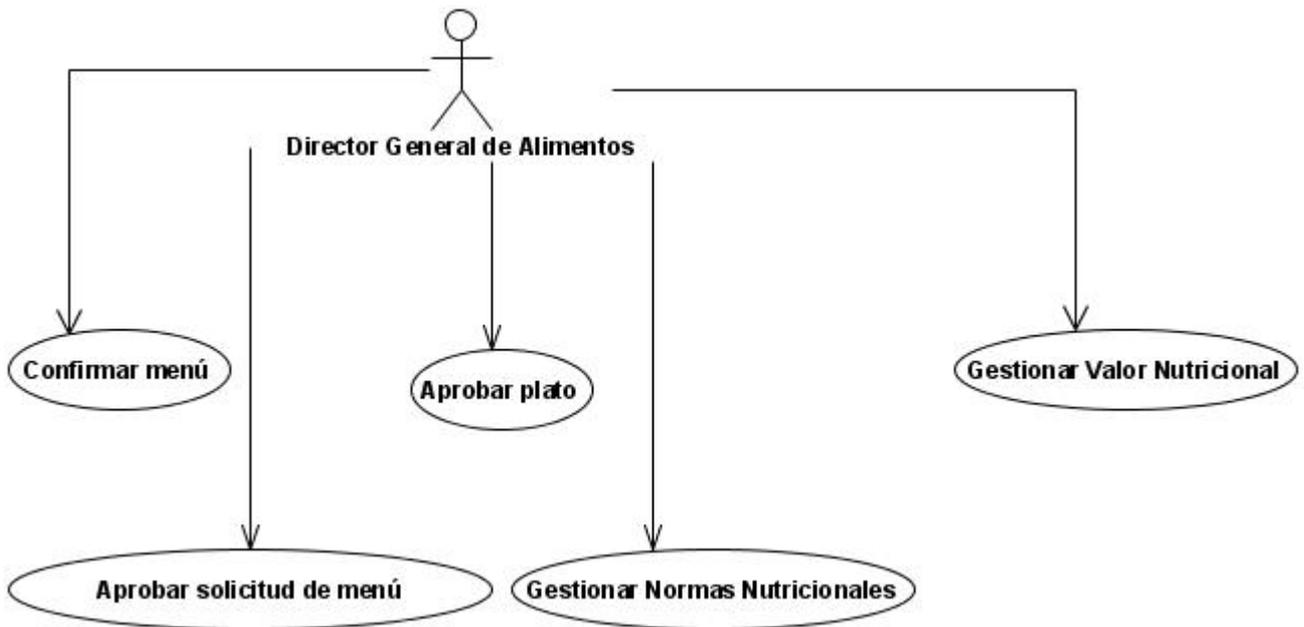


Fig. 12 Diagrama de CU del Paquete Director General de Alimentos

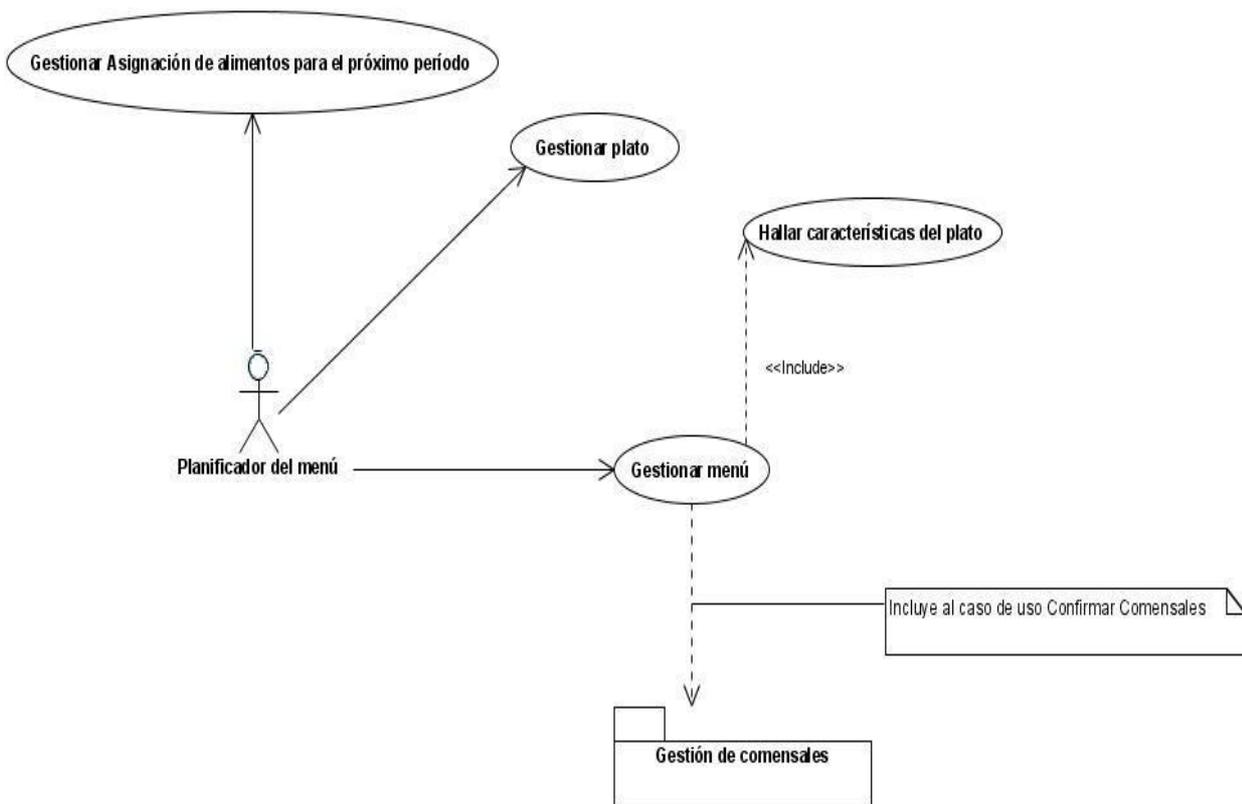


Fig. 13 Diagrama de CU del Paquete Planificador del menú

2.3.6 Descripciones de los Casos de Uso del Sistema

Nombre del Caso de Uso	Gestionar productos
Actores	Gestor_Productos (Jefe de Almacén, Jefe de Compras)
Propósito	Gestionar la información referente a los productos.
Resumen	El CU comienza cuando el Gestor_Productos decide gestionar los productos, luego de esto el sistema ejecuta la acción solicitada, esta puede ser: mostrar productos, buscar producto, eliminar producto, adicionar producto o modificar producto; en caso de ser la primera el sistema muestra un listado con todos los productos, en caso de ser la segunda el sistema solicita la entrada del nombre del producto a buscar y muestra los datos del mismo, en caso de ser la tercera el sistema muestra un listado de los productos existentes y da la posibilidad de eliminarlo, en caso de ser la cuarta el sistema solicita la entrada de una serie de datos entre los que se encuentra el nombre verificando que no exista otro producto con este mismo nombre y en caso de ser la quinta el sistema muestra un listado con los productos que pueden ser modificados.
Referencias	RF1.1, RF1.1.1, RF1.2
Precondiciones	-
Poscondiciones	-
Requerimientos especiales	

Tabla 8 Descripción del CU del Sistema “Gestionar Productos”

Nombre del Caso de Uso	Gestionar plato
Actores	Planificador del menú
Propósito	Gestionar la información referente a los platos.
Resumen	El CU comienza cuando el Planificador del menú decide gestionar los platos, luego de esto el sistema ejecuta la acción solicitada, esta puede ser: crear un plato nuevo, modificar un plato ya existente, buscar un plato o eliminar un

	plato. Si la opción seleccionada es la de crear un plato el sistema muestra un listado con los productos que puede usar para la confección del plato. Si decide modificar un plato el sistema solicita la entrada del nombre del plato, lo muestra para ser modificado. Si decide mostrar los platos existentes el sistema le muestra un listado con los platos que hay creados hasta el momento, y si decide eliminar el sistema le pide el nombre del plato que quiere eliminar y lo elimina.
Referencias	RF1.1, RF2.1, RF2.2, RF2.3, RF2.3.1, RF2.4, RF2.5
Precondiciones	-
Poscondiciones	-
Requerimientos especiales	

Tabla 9 Descripción del CU del Sistema “Gestionar Plato”

Nombre del Caso de Uso	Hallar características del plato
Actores	
Propósito	Hallar la cantidad de proteínas, grasas, carbohidratos y energías que aporta un plato.
Resumen	El CU comienza cuando el Planificador del menú decide hallar la cantidad de proteínas, grasas, energías y carbohidratos que aporta un plato determinado, el sistema calcula estas cantidades teniendo en cuenta lo que aporta cada uno de los productos que forman el plato.
Referencias	RF 1.1, RF3.1, RF3.2, RF3.3, RF3.4
Precondiciones	-
Poscondiciones	-
Requerimientos especiales	

Tabla 10 Descripción del CU del Sistema “Hallar las características del plato”

Nombre del Caso de Uso	Aprobar plato
Actores	Director General de Alimentos
Propósito	Aprobar el plato nuevo creado para que pueda formar parte

	de un menú.
Resumen	El CU comienza cuando el Director General de Alimentos decide aprobar los platos creados, el sistema le muestra un listado con los platos que no se han aprobado y el aprueba los que estime que están correctos.
Referencias	RF 2.3, RF4
Precondiciones	Debe existir al menos un plato creado
Poscondiciones	Se guarda el plato para formar parte del listado de platos y que pueda formar parte de un menú.
Requerimientos especiales	

Tabla 11 Descripción del CU del Sistema “Aprobar Plato”

Nombre del Caso de Uso	Gestionar Asignación de alimentos para el próximo período
Actores	Planificador del Menú
Propósito	Confeccionar la Asignación de alimentos para el próximo período.
Resumen	El CU comienza cuando el Planificador del Menú decide gestionar información referente al documento Asignación de alimentos para el próximo período. El sistema le da la posibilidad de llenar, mostrar y buscar modificar este documento, en caso de que se escoja la última opción el sistema da la posibilidad de modificarlo.
Referencias	RF1.1, RF5.1, RF5.1.1, RF5.1.2, RF5.1.3, RF5.1.4, RF5.2, RF5.2.1, RF5.3, RF5.4.
Precondiciones	-
Poscondiciones	Se genera un nuevo documento con los productos que se le solicitan al Consejo de Estado.
Requerimientos especiales	

Tabla 12 Descripción del CU del Sistema “Gestionar Asignación de Alimentos para el próximo período”

Nombre del Caso de Uso	Crear solicitud de menú
-------------------------------	--------------------------------

Actores	Cliente
Propósito	Hacer una solicitud de un menú la cual será aprobada o no por el Director General de Alimentos.
Resumen	El CU comienza cuando el cliente (persona autorizada) decide hacer la solicitud de un menú elaborada con anterioridad por varias personas, este menú puede ser para una playa, para una excursión, en fin para cualquier actividad planificada ya sea por una facultad, por la UJC o por cualquier área.
Referencias	RF2.3, RF6.
Precondiciones	-
Poscondiciones	-
Requerimientos especiales	

Tabla 13 Descripción del CU del Sistema “Crear solicitud de menú”

Nombre del Caso de Uso	Aprobar solicitud de menú
Actores	Director General de Alimentos
Propósito	Ordenar la confección de un menú de acuerdo con las solicitudes de los diferentes clientes de la Dirección General de Alimentos.
Resumen	El CU comienza cuando el Director General de Alimentos decide ver las solicitudes de menús elaboradas por los clientes (directivos de la FEU, UJC, trabajadores). A partir de ello el sistema le muestra un listado con las solicitudes hechas hasta el momento.
Referencias	RF2.3, RF7.
Precondiciones	Debe haberse hecho al menos una solicitud de un menú.
Poscondiciones	Se genera un menú por planificar.
Requerimientos especiales	

Tabla 14 Descripción del CU del Sistema “Aprobar solicitud de menú”

Nombre del Caso de Uso	Mostrar Existencias
-------------------------------	----------------------------

Actores	Gestor_Existencias (Jefe de Almacén y Planificador del menú)
Propósito	Mostrar las existencias reales que hay en ATM.
Resumen	El CU comienza cuando el Gestor_Existencias decide ver los productos que están en existencia en ATM, luego de esto el sistema muestra un listado con las existencias reales por una conexión que se establecerá con el ASSETS.
Referencias	RF1.1, RF8
Precondiciones	-
Poscondiciones	-
Requerimientos especiales	

Tabla 15 Descripción del CU del Sistema “Mostrar Existencias”

Nombre del Caso de Uso	Gestionar Carga
Actores	Jefe de Compras
Propósito	Determinar los productos que van a formar parte de una carga.
Resumen	El CU comienza cuando el Jefe de Compras decide gestionar la información referente a las cargas que formarán parte de uno o varios planes de carga. El sistema le dará la posibilidad de crear una carga, de mostrar las cargas existentes, de buscar una carga, de modificar alguna ya existente o de confirmar una carga
Referencias	RF1.1, RF9.1, RF 9.1.1, RF9.2, RF9.3, RF9.4, RF 9.5
Precondiciones	-
Poscondiciones	-
Requerimientos especiales	

Tabla 16 Descripción del CU del Sistema “Gestionar Carga”

Nombre del Caso de Uso	Gestionar Plan de Carga
Actores	Jefe de Compras
Propósito	Gestionar la información que tiene que ver con el documento

	Plan de Carga.
Resumen	El CU comienza cuando el Jefe de Compras decide gestionar la información que requiere el documento Plan de Carga, luego de esto el sistema ejecuta la acción solicitada, esta puede ser: mostrar el Plan de Carga, llenar el Plan de Carga, modificar Plan de Carga o confirmar Plan de Carga. Si se decide mostrar el plan el sistema muestra todos los datos de todos los planes de carga que han sido creados, si la opción solicitada es llenar el sistema pide el período para el cual se creará el plan, si se decide modificar un Plan de Carga existente el sistema solicita el período del mismo y muestra un listado de cargas las cuales pueden ser incluidas en el nuevo plan y si se decide confirmar el sistema muestra un listado con los planes de carga sin modificar.
Referencias	RF9.1, RF10.1, RF10.1.1, RF10.2, RF10.3, RF10.4
Precondiciones	Tener al menos una carga creada.
Poscondiciones	Este documento será archivado para ser consultado siempre que sea necesario.
Requerimientos especiales	

Tabla 17 Descripción del CU del Sistema “Gestionar Plan de Carga”

Nombre	Gestionar menú
Actores	Planificador del menú
Propósito	Gestionar toda la información referente a cualquier menú asociado a las unidades de producción y servicio.
Resumen	El CU comienza cuando el Planificador del menú decide gestionar el menú, luego de esto el sistema ejecuta la acción solicitada, esta puede ser: crear un menú, modificar un menú existente, mostrar un menú, buscar un menú o realizar el ajuste del menú. Si la opción seleccionada es la de crear un menú el sistema brinda la opción de crearlo. Si decide modificar uno ya existente el sistema muestra los menús que ya han sido creados. Si decide mostrar los menús existentes

	el sistema le muestra un listado con los menús que hay creados hasta el momento. En caso de que la opción solicitada sea buscar el sistema solicita la entrada del identificador del menú y lo muestra y si se decide reajustar el menú el sistema muestra el menú para ese día informándole al Planificador si se debe modificar o no el mismo.
Referencias	RF2.3, RF11.1, RF11.1.1, RF11.2, RF11.2.1, RF11.3, RF11.4, RF11.5, RF11.6, CU Hallar características del plato (incluido), CU Verificar Comensales (incluido)
Precondiciones	Debe existir una Asignación de alimentos aprobada.
Poscondiciones	Luego de cada acción el menú será guardado como menú plan.
Requerimientos especiales	

Tabla 18 Descripción del CU del Sistema “Gestionar menú”

Nombre del Caso de Uso	Confirmar menú
Actores	Director General de Alimentos
Propósito	Aprobar el menú planificado
Resumen	El CU comienza cuando el Director General de Alimentos decide confirmar los menús planificados el sistema le muestra un listado con los menús planificados y el confirma los que desee.
Referencias	RF 11.2, RF12
Precondiciones	Debe existir al menos un menú planificado.
Poscondiciones	Se guarda este documento como menú real.
Requerimientos especiales	

Tabla 19 Descripción del CU del Sistema “Confirmar menú”

Nombre del Caso de Uso	Gestionar Normas Nutricionales
Actores	Director General de Alimentos
Propósito	Gestionar las normas nutricionales establecidas por el Consejo de Estado asociadas a un evento.

Resumen	El CU comienza cuando el Director General de Alimentos decide gestionar la información referente a las normas nutricionales asociadas a un evento. El sistema le da la posibilidad de: mostrar las normas establecidas hasta el momento y de modificar esas normas además de adicionar nuevas normas.
Referencias	RF13.1, RF13.2, RF13.3
Precondiciones	-
Poscondiciones	Las normas quedan guardadas para poder saber si los menús que se planifican cumplen con las normas nutricionales establecidas.
Requerimientos especiales	

Tabla 20 Descripción del CU del Sistema “Gestionar Normas Nutricionales”

Nombre del Caso de Uso	Gestionar Valor Nutricional
Actores	Director General de Alimentos
Propósito	Gestionar cuanto se debe ofertar en un evento de: <ul style="list-style-type: none"> • Grasas • Proteínas • Carbohidratos • Energías
Resumen	El CU comienza cuando el Director General de Alimentos decide gestionar la información referente al balance nutricional asociado a un evento. El sistema le da la posibilidad de: mostrar el valor nutricional establecido hasta el momento y de modificarlo, además de adicionar un nuevo valor.
Referencias	RF14.1, RF14.2, RF14.3
Precondiciones	
Poscondiciones	El valor nutricional queda guardado para poder saber si los

	menús que se planifican cumplen con el balance nutricional establecido.
Requerimientos especiales	

Tabla 21 Descripción del CU del Sistema “Gestionar Valor Nutricional”

Para ver las descripciones de cada uno de los flujos de eventos de cada CU del sistema ver el expediente de proyecto.

2.3.7 Vista de Casos de Uso

La vista de la arquitectura del modelo de CU debería incluir los CU que describan alguna funcionalidad importante y crítica, o que impliquen algún requisito importante que deba desarrollarse pronto dentro del ciclo de vida del software. (15)

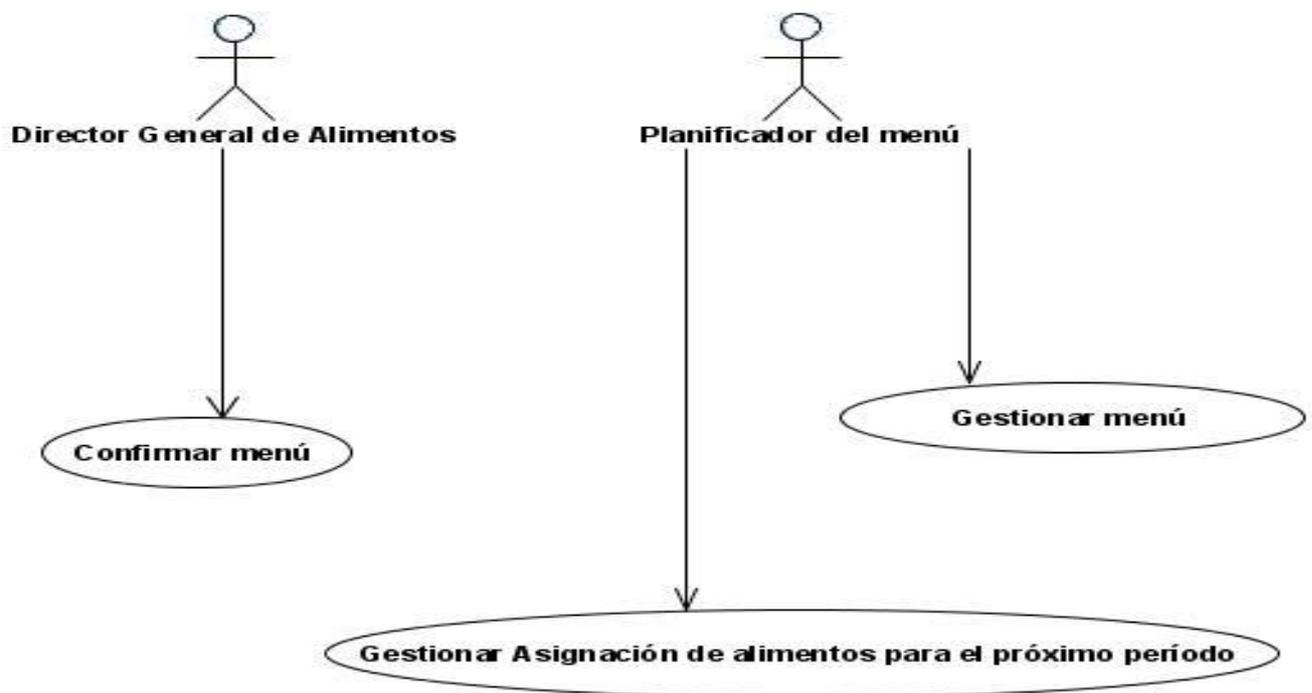


Fig. 14 Vista de CU del sistema

2.4 Conclusiones

A lo largo de este capítulo se definieron los actores, trabajadores, entidades y reglas del negocio, además de mostrarse cómo fluye actualmente el proceso de gestión del menú. Se definieron también

los requerimientos tanto los funcionales como los no funcionales de la aplicación y los CU del sistema con sus respectivas descripciones

Capítulo 3: Análisis y Diseño

Introducción

En este capítulo se obtienen los artefactos correspondientes al flujo de trabajo de análisis y diseño. Se modela el diseño, donde se representan las realizaciones de los CU del sistema, a través de las clases del diseño con extensiones Web además de las descripciones de las mismas.

3.1 Arquitectura del sistema

La arquitectura de software es una especificación de las principales ideas del diseño, que proporciona una descripción entendible de la arquitectura del sistema software. Contiene varias vistas entre la Vista Lógica y la Vista de Despliegue.

3.1.1 Vista Lógica

La Vista Lógica muestra los principales componentes de diseño y sus relaciones de manera independiente. Representa un subconjunto del Modelo de diseño, mostrando los elementos del mismo que son más importantes para la arquitectura del sistema. Describe las clases fundamentales, su organización en paquetes y subsistemas y estos en capas.

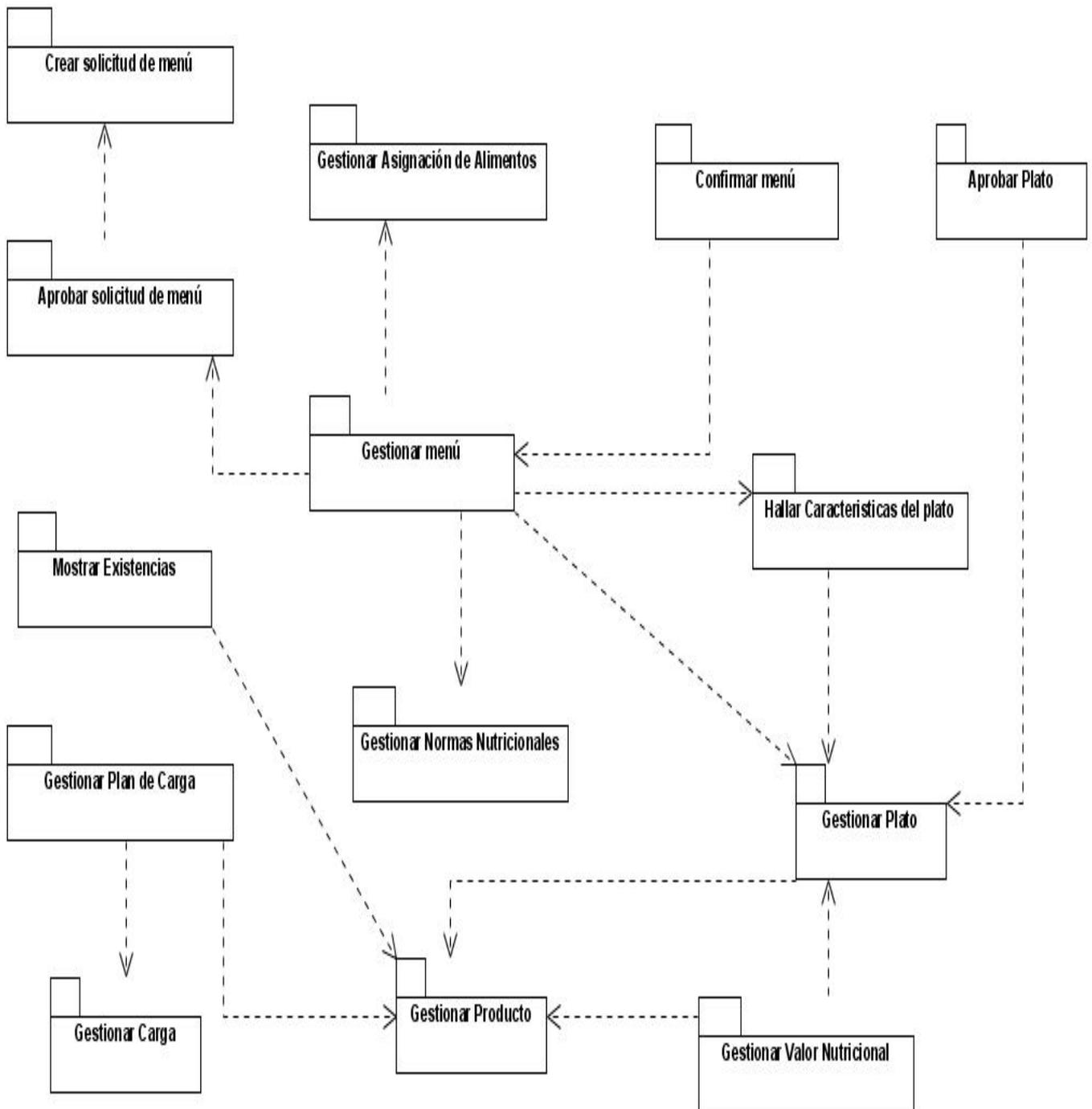


Fig. 15 Diagrama de la Vista Lógica

3.1.2 Vista de despliegue

La Vista de despliegue suministra una base para la comprensión de la distribución física de un sistema a través de nodos. Se ocupa de los requerimientos no funcionales.

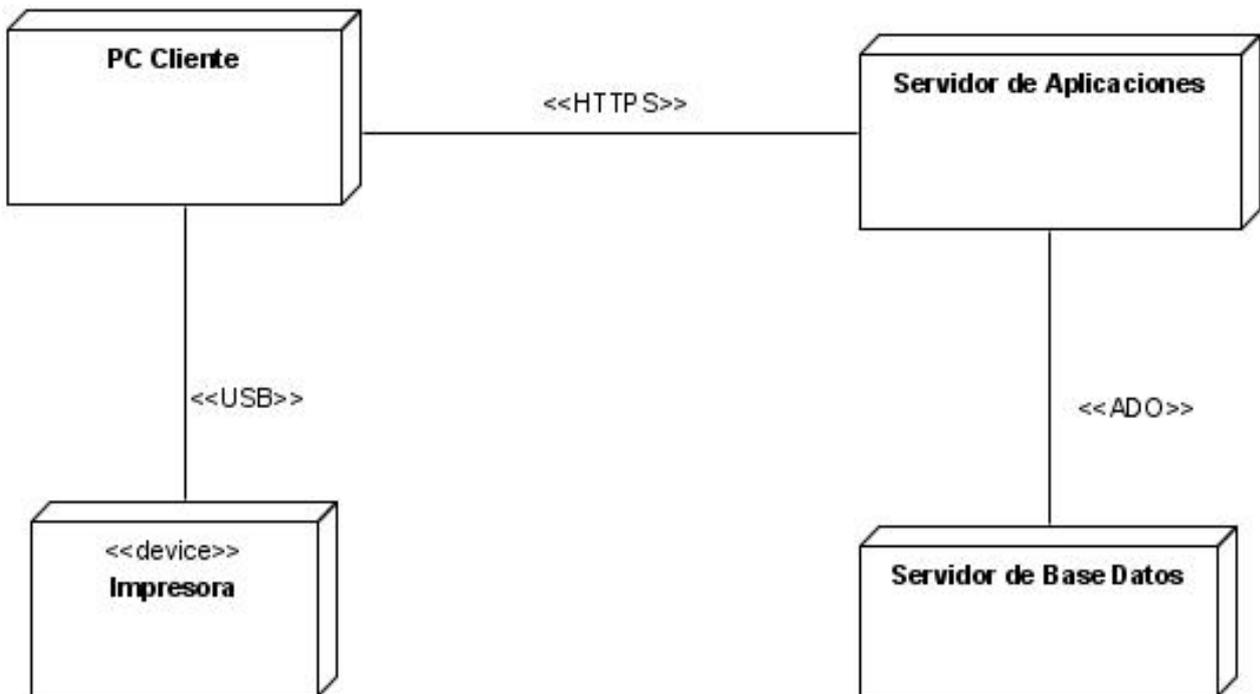


Fig. 16 Vista de Despliegue

3.2 Prototipos de interfaz no funcionales

Los Prototipos de interfaz no funcionales ayudan a adquirir una visión más acertada de lo que será el producto, quedando claro lo que se requiere y lo que es factible; son construidos además con el objetivo de validar y explorar el diseño de la aplicación, esto asegura que se haga un diseño correcto del sistema y no se pierda tiempo en el desarrollo.

En los anexos 5, 6, 7, 8, 9 y 10 aparecen algunos de los prototipos no funcionales realizados, el resto se encuentra en el expediente de proyecto.

3.3 Mapa de navegación

Un mapa de navegación es la representación gráfica de la organización de la información de una estructura web. Expresa todas las relaciones de jerarquía y secuencia y permite elaborar escenarios de comportamiento de los usuarios. También grafica, de modo que todos los profesionales participantes en un proyecto lo tengan claro, diferencias entre páginas dinámicas, administrables o estáticas.

El principal valor de un mapa de navegación es que permite anticipar errores de organización de la información, de modo de corregirlos cuando aún no se ha invertido tiempo y dinero en la construcción del producto. (16)

- Seleccionar la pantalla de entrada al sitio web presentación (página web: index.html).

- Ordenar de manera jerarquizada las pantallas con los contenidos (por niveles o categorías).
- Establecer los vínculos entre pantallas (páginas web) permitiendo una navegación hipertextual.(17)

Para hacer más entendible el mapa de navegación de la aplicación Web que se desea diseñar, se decidió dividir mismo en tres.(Ver anexos 11, 12 y 13)

3.4 Aplicación de los Patrones de diseño

Durante el flujo de trabajo de análisis y diseño propuesto por RUP es casi inevitable el uso de los patrones de diseño. En el primer capítulo de este documento se mencionan los patrones que van a ser utilizados en dicho flujo de trabajo, en el presente epígrafe se explicará el uso de los mismos, es decir, donde se aplica cada uno.

El Bajo acoplamiento se usa para desacoplar las vistas de los modelos y los modelos de la forma en que se muestran e ingresan los datos, además se pone en evidencia el uso de este patrón por ejemplo, en la clase Actions quien hereda solamente de sfActions logrando un bajo acoplamiento de clases.

El patrón de Alta cohesión se hace evidente en cada elemento diseñado, pues los mismos están altamente especializados en sus tareas por ejemplo, la vista en mostrar los datos al usuario, el controlador en las entradas y el modelo en su objetivo del negocio. Symfony admite la asignación de responsabilidades con una alta cohesión, como ejemplo de ellos se tiene la clase Actions la cual es responsable de definir las acciones para las plantillas y colaborar con otras para realizar diferentes operaciones (instanciar objetos y acceder a las properties), es decir, que está formada por diferentes funcionalidades las que se encuentran estrechamente relacionadas lo que hace posible que el software sea flexible frente a grandes cambios.

El Experto es uno de los más utilizados, por ejemplo: en Propel, el cual es la librería externa que utiliza Symfony para realizar su capa de abstracción en el modelo, se encapsula toda la lógica de los datos y son generadas las clases con todas las funcionalidades comunes de las entidades.

El Controlador se pone de manifiesto en el controlador frontal pues este maneja todas las peticiones Web que se realizan al sistema, ya que es el punto de entrada único de toda la aplicación en un entorno determinado. Cuando este recibe una petición, utiliza el sistema de enrutamiento para asociar el nombre de una acción y el nombre de un módulo con la URL entrada por el usuario.

El Creador, se evidencia en la clase Intermedia se encuentran las acciones definidas para dar respuesta a cada funcionalidad del sistema. En la misma se crean los objetos de las clases que

representan las entidades, evidenciando de este modo que la clase Intermedia es "creador" de dichas entidades.

La Fabricación pura se pone de manifiesto en la cla Intermedia ya que el objetivo de la misma es manejar el acceso a los datos de las clases del Modelo para lo cual tiene definido un grupo de funcionalidades que responden a una acción que debe ejecutar el sistema y que son llamadas a través de la clase Actions.

El patrón *Singleton* se utiliza en la clase Intermedia, diseñada en el Modelo y es la encargada de la lógica del negocio, pues la misma tiene un atributo de tipo Singleton (instancia) a través del cual la clase Actions accede a todas las funcionalidades de esta clase. Además se evidencia en el controlador frontal en el cual se puede hacer llamada a `sfContext::getInstance()`, en una acción, el método `getContext()`, un objeto muy útil que guarda una referencia a todos los objetos del núcleo de Symfony.

Debido a la arquitectura MVC y a la composición que tienen cada una de estas partes en Symfony se usa implícitamente el patrón *Decorator*, específicamente en la Vista ya que esta está compuesta por diversas partes dentro de las que está la plantilla y el *layout*. Este último es el que almacena el código común para todas las páginas de la aplicación, por lo que el contenido de la plantilla se integra a él, es decir, es el que decora la plantilla.

El *Composite* es usado fundamentalmente en las vistas, pues estas están compuestas por diferentes elementos como son: el layout, la plantilla, los elementos parciales.

El patrón *Facade* se utiliza en la clase Intermedia con el objetivo de simplificar el acceso a un conjunto de clases, las del modelo, para ello en esta se define un grupo de métodos, uno para cada operación permitida de modo que sean estos métodos los que internamente hagan las operaciones con el fin de llevar a cabo la correcta lógica de la aplicación.

El patrón *Chain of Responsibility* es usado en todas las clases diseñadas puesto que cada una de ellas tendrá una responsabilidad en el proceso solicitud – respuesta de una petición realizada por un usuario.

3.5 Modelo del diseño

En el flujo de trabajo de Análisis y Diseño, definido por RUP, se describe cómo se debe implementar el sistema, esto como resultado de un análisis realizado a los requerimientos no funcionales del mismo. El diseño no debe tener ambigüedades para que el modelo final que se obtenga sea suficiente para la implementación. El modelo de diseño es el resultado más importante de este flujo de trabajo, el cual

consiste en colaboraciones de clases, las que pueden ser agrupadas en paquetes y subsistemas, además describe la realización de los CU.

3.5.1 Realización de Casos de Uso

Describe como un caso de uso particular es realizado dentro del modelo de diseño, en términos de colaboración de paquetes (18).

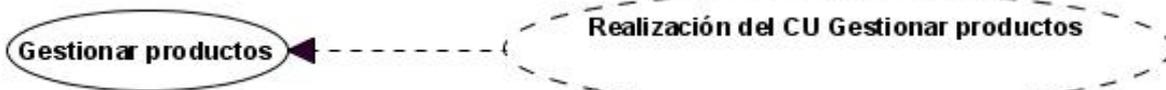


Fig. 17 Realización del CU Gestionar productos

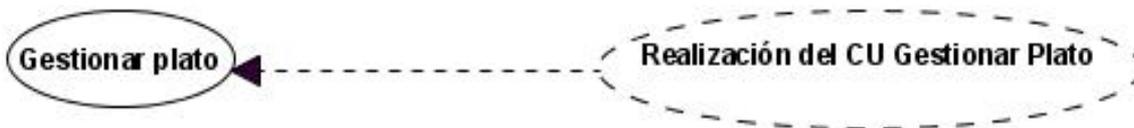


Fig. 18 Realización del CU Gestionar plato

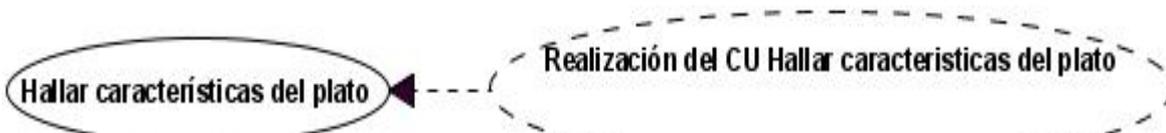


Fig. 19 Realización del Caso de Uso Hallar características del plato

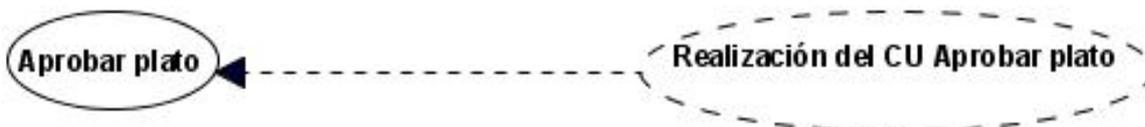


Fig. 20 Realización del Caso de Uso Aprobar plato

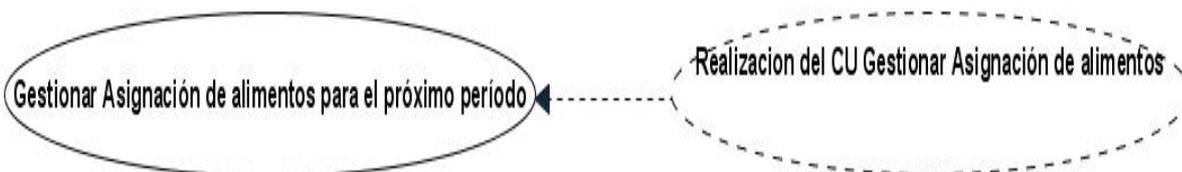


Fig. 21 Realización del Caso de Uso Gestionar Asignación de alimentos para el próximo período

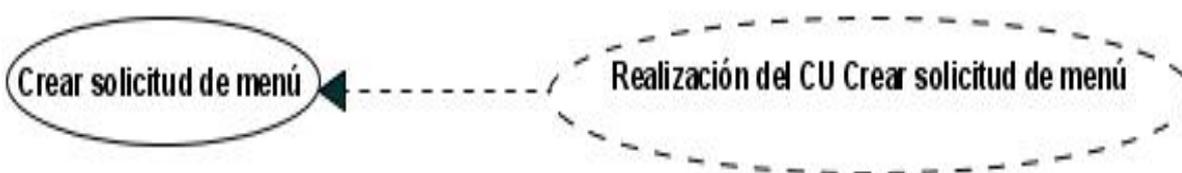


Fig. 22 Realización del CU Crear solicitud de menú

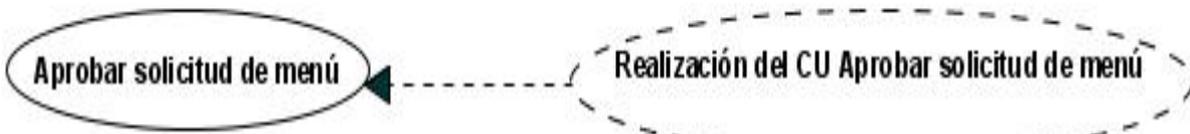


Fig. 23 Realización del Caso de Uso Aprobar Solicitud de menú

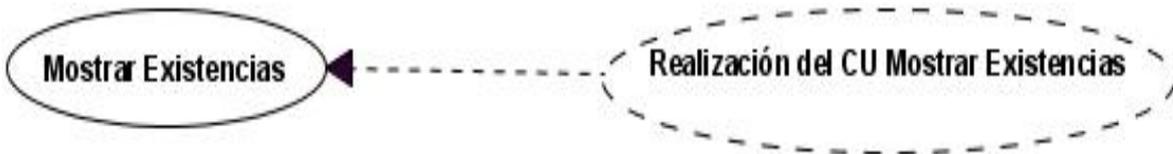


Fig. 24 Realización del Caso de Uso Mostrar Existencias

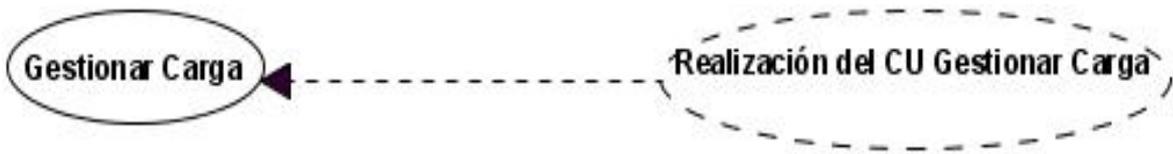


Fig. 25 Realización del Caso de Uso Gestionar Carga

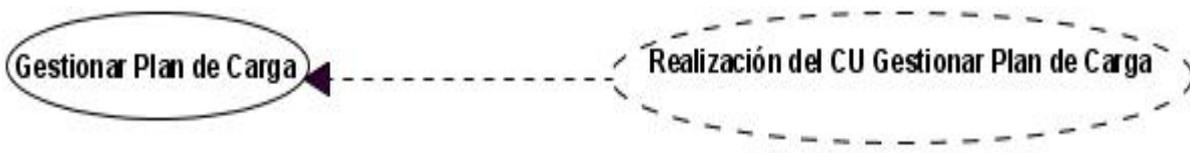


Fig. 26 Realización del Caso de Uso gestionar Plan de Carga

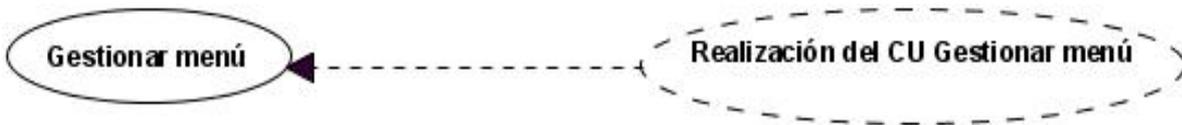


Fig. 27 Realización del Caso de Uso Gestionar menú

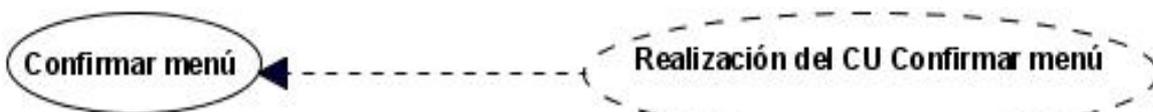


Fig. 28 Realización del Caso de Uso Confirmar menú



Fig. 29 Realización del Caso de Uso Gestionar Normas Nutricionales



Fig. 30 Realización del Caso de Uso Gestionar Valor Nutricional

3.5.2 Descripciones de las Clases del Diseño

Nombre:	Fr_Aprobarplato	
Tipo de clase	Formulario (Formulario de la página cliente Aprobar Plato)	
Atributo	Tipo	
\$platos	Select	
\$cancelar	Button	
\$aceptar	Button	

Tabla 22 Descripción de la Clase del Diseño de Aplicaciones Web: Fr_Aprobarplato

Nombre:	PlatoPeer	
Tipo de clase	Clase Peer (Permite la comunicación con la Base de Datos, es decir proporciona los medios necesarios para obtener los registros de las tablas)	
Atributo	Tipo	
Para cada responsabilidad:		
Nombre:	doSelect()	
Descripción:	Permite obtener un listado de objetos con todos los datos del Plato que se encuentran en las tablas (en dependencia del objeto tipo Criteria que se le pase por parámetro)	
Nombre:	doDelete()	
Descripción:	Permite eliminar pasándole como parámetro un valor, de la entidad Plato	

Tabla 23 Descripción de la Clase del Diseño de Aplicaciones Web: PlatoPeer

Nombre:	aprobar_plato_Actions	
Tipo de clase	Clase Actions (Clase utiliza el modelo y crea variables para la vista)	
Atributo		Tipo
Nombre:	executeAprobarplato()	
Descripción:	Permite aprobar un plato	
Nombre:	executeListarplatos()	
Descripción:	Permite listar los platos	
Nombre:	executeBuscaringredientes()	
Descripción:	Permite buscar los ingredientes de un plato	
Nombre:	executeMostrarpcpaldga()	
Descripción:	Indica la creación de la página principal que se muestra con las opciones que se pueden realizar, en dependencia de los permisos que tenga el actor.	

Tabla 24 Descripción de la Clase del Diseño de Aplicaciones Web: aprobar_plato_Actions

Nombre:	IntermediaPlato	
Tipo de clase	Clase Actions (Clase utiliza el modelo y crea variables para la vista)	
Atributo		Tipo
\$intancia		Singleton
Nombre:	aprobarplato()	
Descripción:	Permite aprobar un plato	
Nombre:	editarPlato()	
Descripción:	Permite modificar un plato	
Nombre:	crearPlato()	
Descripción:	Permite crear un plato	
Nombre:	buscarPlato()	
Descripción:	Permite buscar un plato	
Nombre:	eliminarPlato()	
Descripción:	Permite eliminar un plato	
Nombre:	mostrarPlatos()	
Descripción:	Permite mostrar un plato	
Nombre:	hallarCaract()	
Descripción:	Permite hallar las características de un plato	
Nombre:	getInstancia()	
Descripción:	Devuelve una instancia	
Nombre:	listarPlatos()	
Descripción:	Permite listar los platos	
Nombre:	buscarIngredientes()	
Descripción:	Permite buscar los ingredientes de un plato	
Nombre:	executeMostrarpcpaldga()	
Descripción:	Indica la creación de la página principal que se muestra con las opciones que se pueden realizar, en dependencia de los permisos que tenga el actor.	

Tabla 25 Descripciones de la Clase del Diseño de aplicaciones Web: **IntermediaPlato**

Nombre:	BasePlato	
Tipo de clase	Clase Base (Clase que se genera directamente a partir del esquema)	
Atributo		Tipo
\$nombre_plato		
\$coccion		
\$tiempo		
\$temperatura		
\$estado		
\$id_plato		
\$unidad_medida		
Para cada responsabilidad:		
Nombre:	getNombre()	
Descripción:	Devuelve el nombre de un menú	
Nombre:	setNombre()	
Descripción:	Cambia el nombre de un menú	
Nombre:	getCoccion ()	
Descripción:	Devuelve la cocción de un plato	
Nombre:	setCoccion ()	
Descripción:	Cambia la cocción de un plato	
Nombre:	getTiempo()	
Descripción:	Devuelve el tiempo de cocción de un plato	
Nombre:	setTiempo()	
Descripción:	Cambia el tiempo de cocción de un plato	
Nombre:	getTemperatura()	
Descripción:	Devuelve la temperatura a la que debe cocinarse un plato	
Nombre:	setTemperatura()	
Descripción:	Cambia la temperatura a la que debe cocinarse un plato	
Nombre:	getEstado()	
Descripción:	Devuelve el estado de un plato	
Nombre:	setEstado()	
Descripción:	Cambia el estado de un plato	

Nombre:	getUM()
Descripción:	Devuelve la unidad de medida de un plato
Nombre:	setUM()
Descripción:	Cambia la unidad de medida de un plato
Nombre:	getID()
Descripción:	Devuelve el identificador de un plato
Nombre:	setID()
Descripción:	Cambia el identificador de un plato
Nombre:	save()
Descripción:	Función propia de la Base para salvar en la entidad los nuevos datos que se entran
Nombre:	dosave()
Descripción:	Función llamada por la función save()

Tabla 26 Descripción de la Clase del Diseño de Aplicaciones Web: BasePlato

3.5.3 Diagramas de Clases del diseño

Las clases del diseño son una construcción similar en la implementación del sistema, pues el lenguaje usado en estas es el mismo que el utilizado en la implementación, teniendo sus métodos una correspondencia directa con los métodos en la implementación de las clases.

Los diagramas de clases muestran las clases del sistema y sus interrelaciones, así como un conjunto de interfaces, colaboraciones y sus relaciones.

UML posee una extensión para aplicaciones Web, la cual fue desarrollada por Jim Conallen desde 1998 denominada WAE "Web Application Extensión", esta permite beneficiar toda la gramática interna de UML para modelar aplicaciones con elementos específicos de la arquitectura de un entorno WEB.

(19), esta extensión, utilizada para el diseño de las clases, usa los siguientes estereotipos:



nombre_clase

<<Server Page>> Representa la página Web que tiene el código que se ejecuta del lado del servidor, el cual interactúa con recursos en el servidor. Las operaciones representan las funciones del código y los atributos las variables visibles dentro del alcance de la página. (20)



Formulario

<<Form>> Posee un grupo de elementos de entrada que son parte de una página cliente. Se relaciona directamente con la etiqueta de igual nombre del HTML. Sus atributos son los elementos de entrada del formulario (Text Field, Text Area, Button,

Label, Radio Button, Radio Group, Select, Check Box y Hidden Fields). (20)



<<Client Page>> Una instancia de Página Cliente es una página Web, con formato HTML; mezcla de datos, presentación y lógica. Son interpretadas por el browser. Cada página cliente solo puede ser construida por una página servidor. (20)

Debido a la utilización del framework Symfony se introduce el uso del patrón de arquitectura Modelo-Vista-Controlador, por lo que los diagramas de clases del diseño correspondientes reflejarán cómo tienen lugar las interacciones entre los diferentes elementos y clases del sistema que los componen. Cada una de las peticiones realizadas por los usuarios es atendida por un controlador frontal (Index), el cual, pasándole el módulo y la acción a realizar, le encomienda a un componente de Symfony procesar la solicitud del usuario. Este componente además de verificar la seguridad del sistema, es decir, aparte de verificar que el usuario que realiza la petición tenga los privilegios necesarios para acceder a esa funcionalidad del sistema le informa a la clase Actions la acción que debe realizar, la misma es la encargada de recuperar la información necesaria de las clases contenidas en el modelo y de notificar a la plantilla para que construya, con los datos requeridos, las páginas clientes.

A continuación se presentan los diagramas de clases del diseño, en el primero de estos diagramas se muestra cómo quedaría el Modelo con todas las clases correspondientes al CU Aprobar plato. En los restantes diagramas en el paquete del modelo se representa solo una de las clases que contiene este paquete para cada CU, especificándose mediante una nota las que se utilizan en cada uno de los procesos, por la particularidad del framework utilizado pues este genera por cada tabla de la Base de Datos cuatro clases. Debido a esto se tiene un paquete Modelo general en que se encuentran todas las clases necesarias para la ejecución de los procesos relacionados con la gestión del menú.

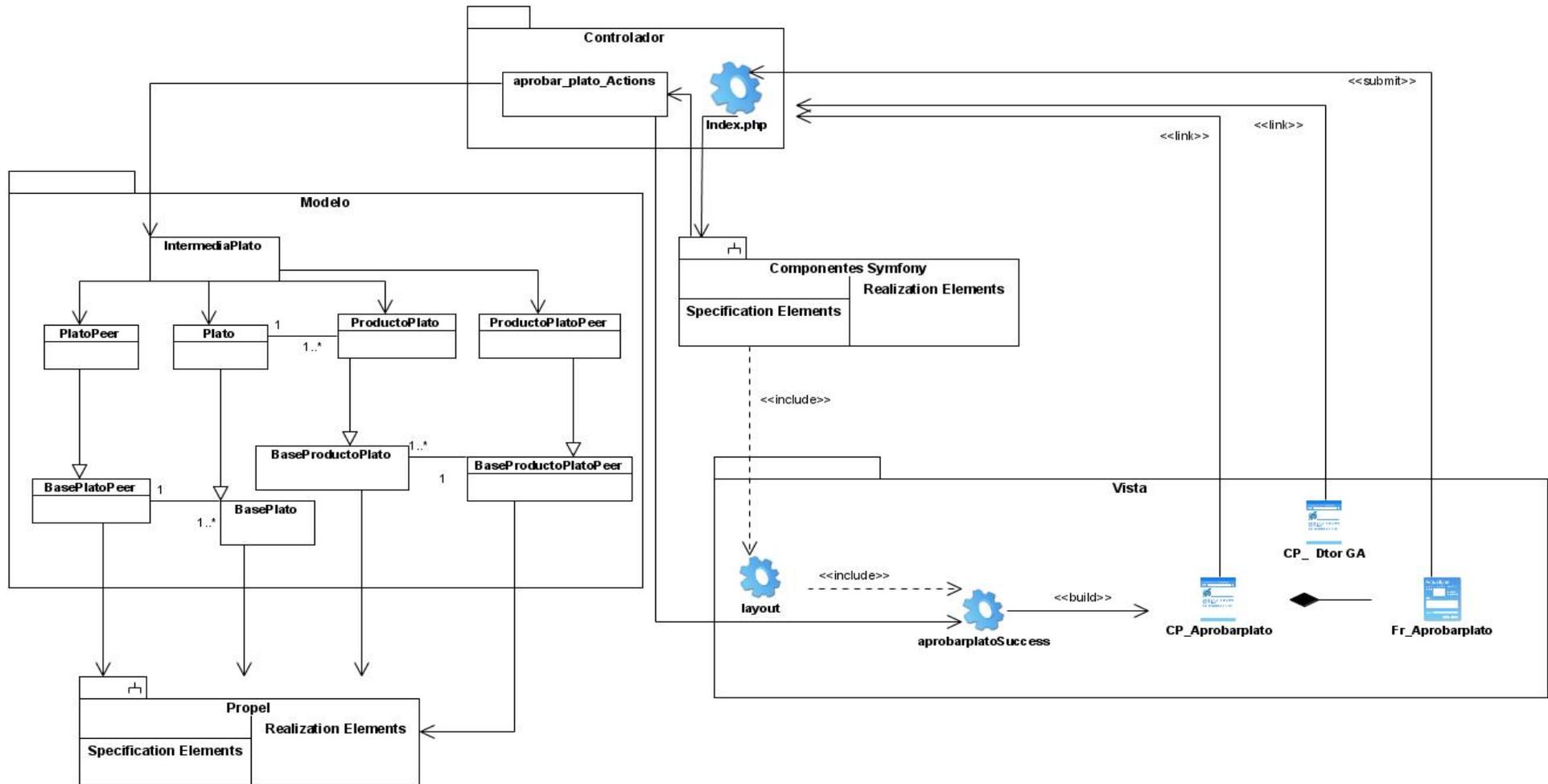


Fig. 31 Diagrama de clases del diseño Web del CU Aprobar plato

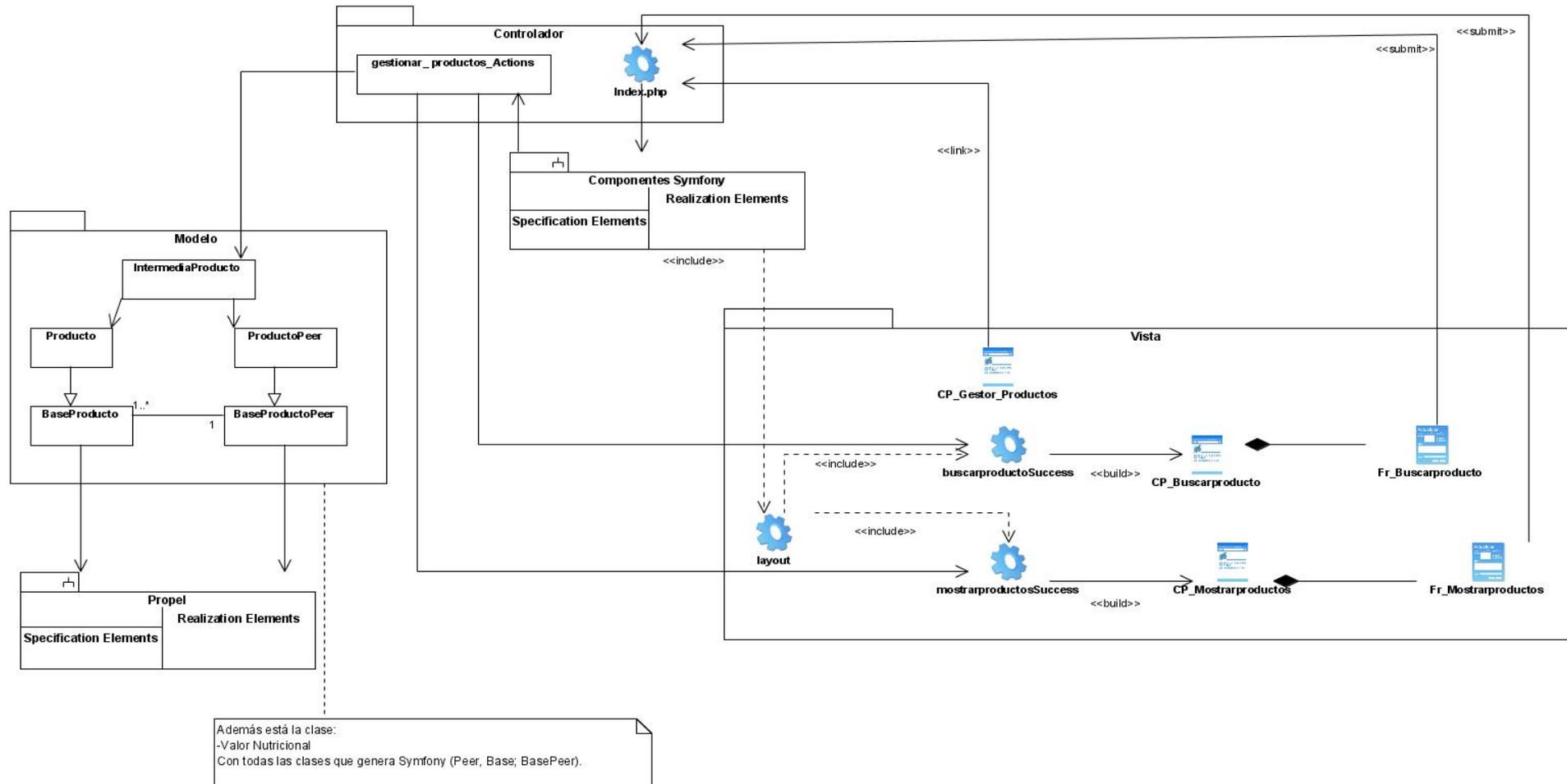


Fig. 32 Diagrama de clases del diseño (Web) del CU Gestionar productos

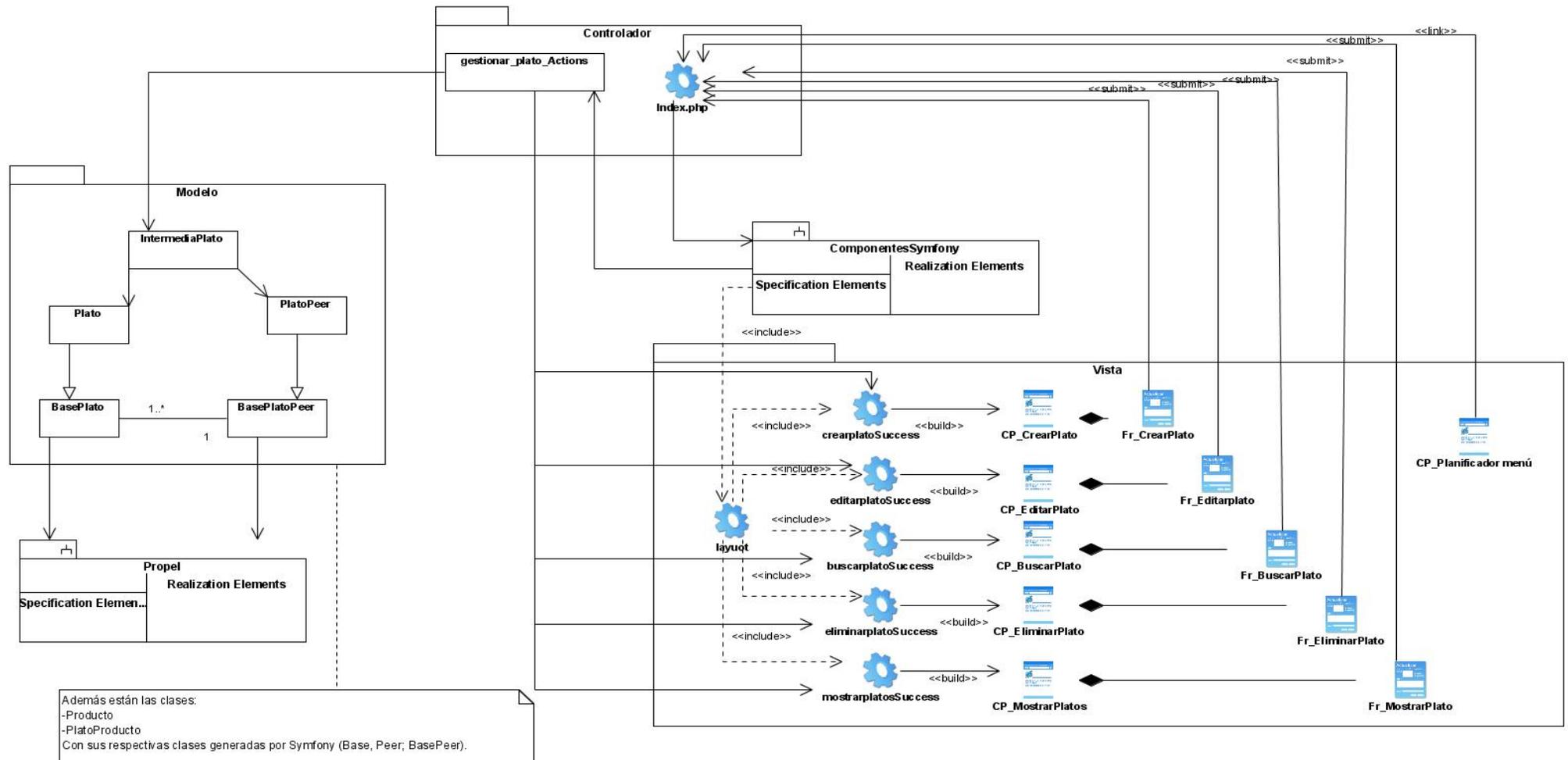


Fig. 33 Diagrama de clases del diseño (Web) del CU Gestionar plato

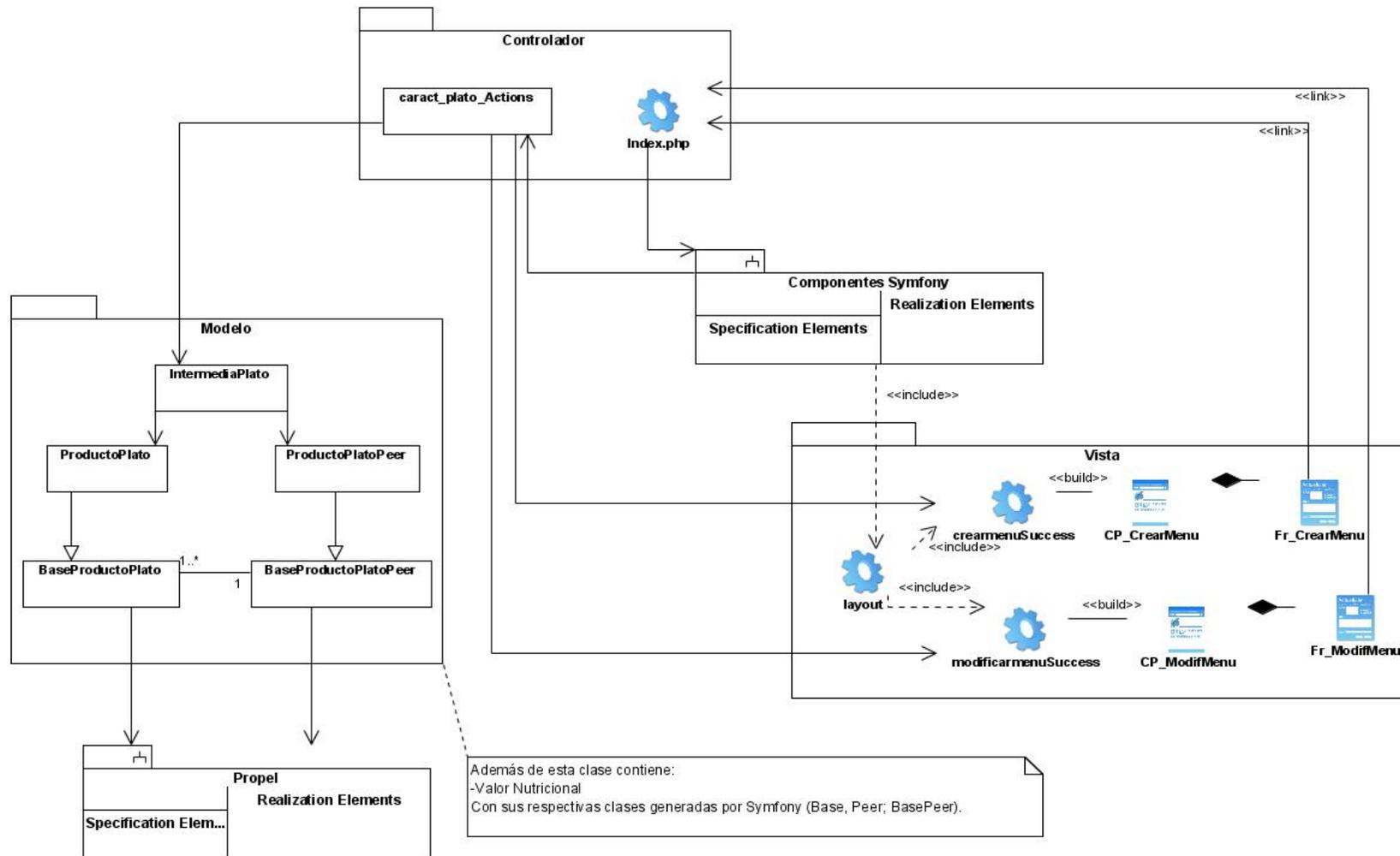


Fig. 34 Diagrama de clases del diseño (Web) del CU Hallar características del plato

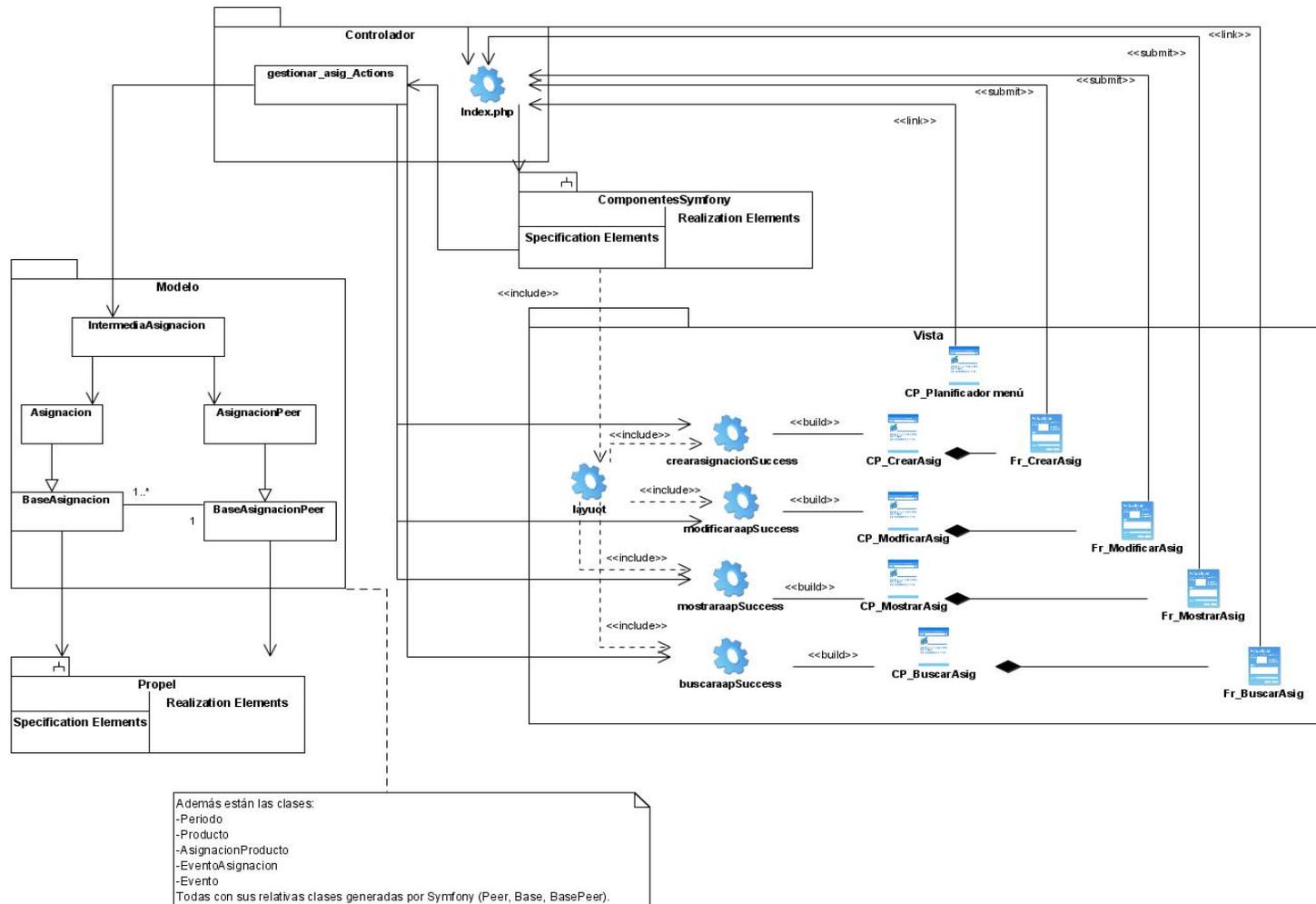


Fig. 35 Diagrama de clases del diseño (Web) del CU Gestionar Asignación de alimentos para el próximo período

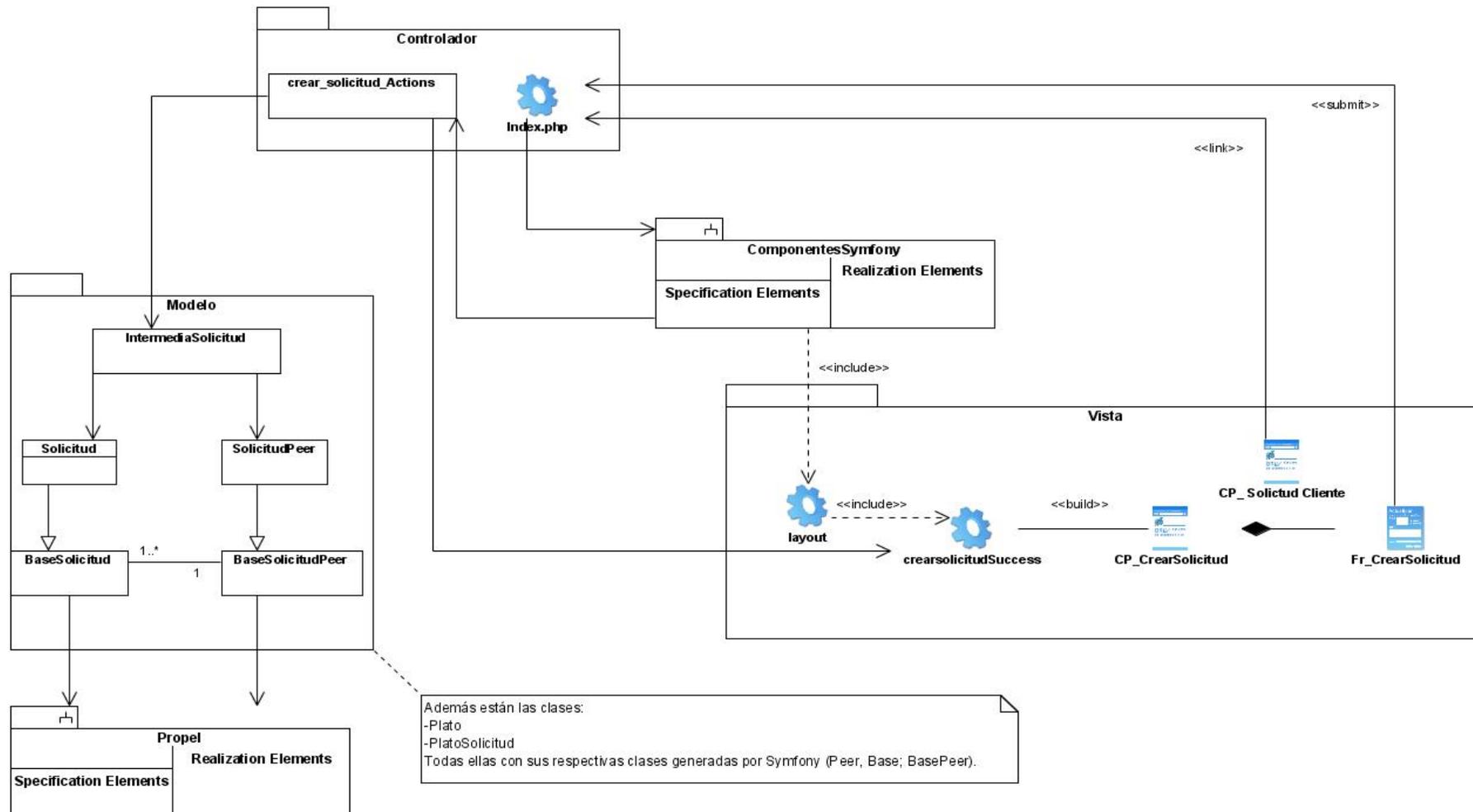


Fig. 36 Diagrama de clases del diseño (Web) del CU Crear solicitud de menú

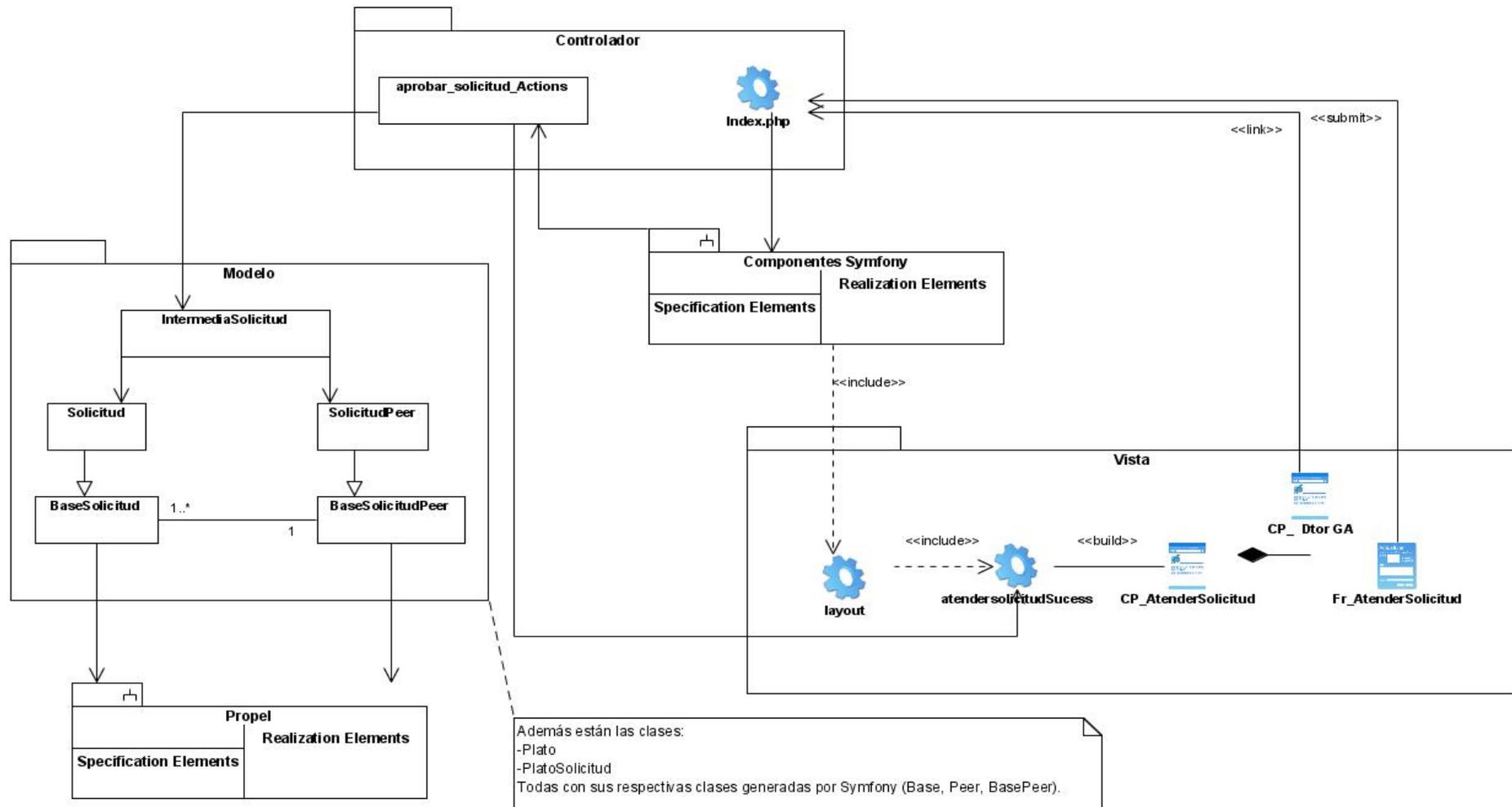


Fig. 37 Diagrama de clases del diseño (Web) del CU Aprobar solicitud menú

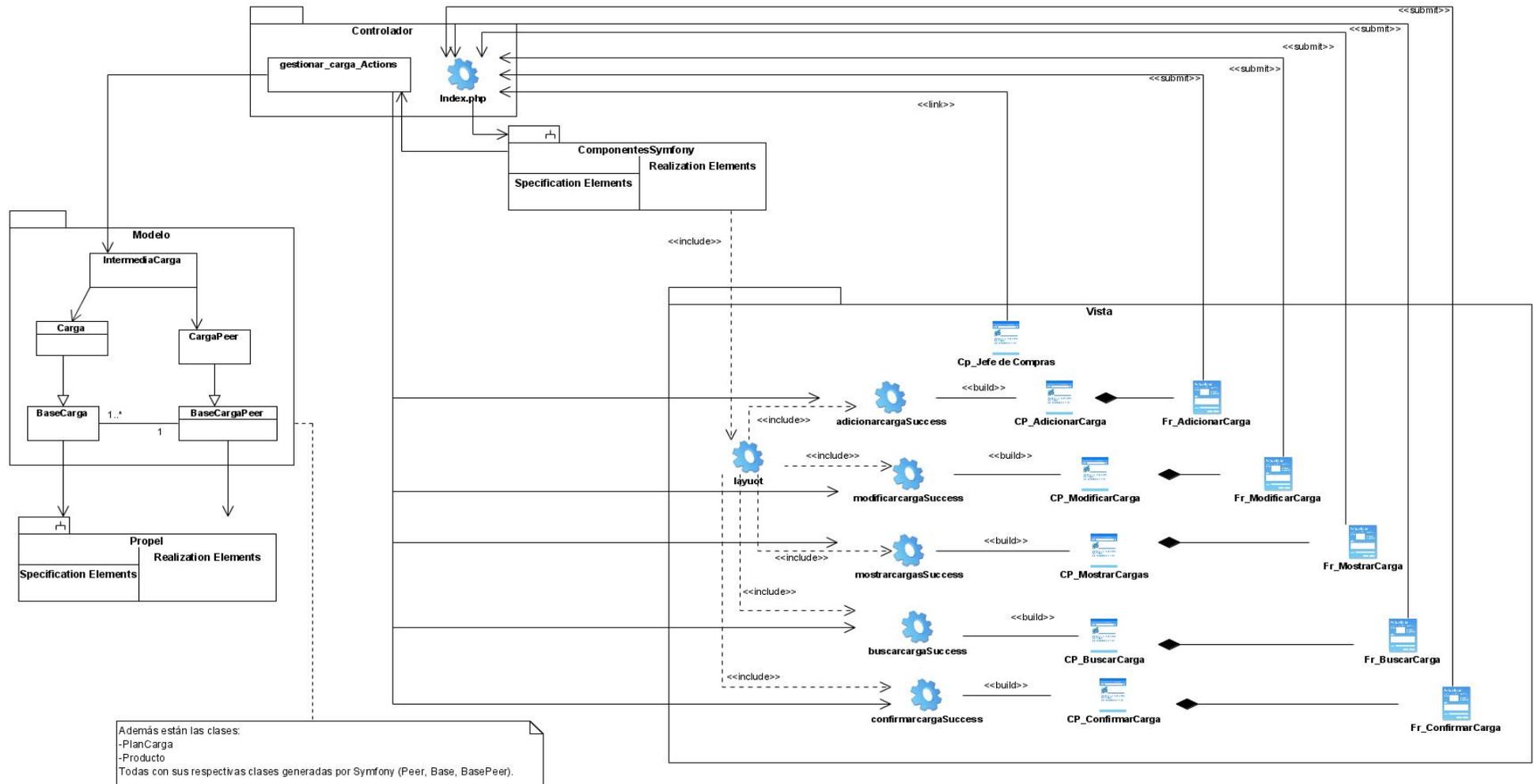


Fig. 39 Diagrama de clases del diseño (Web) del CU Gestionar Carga

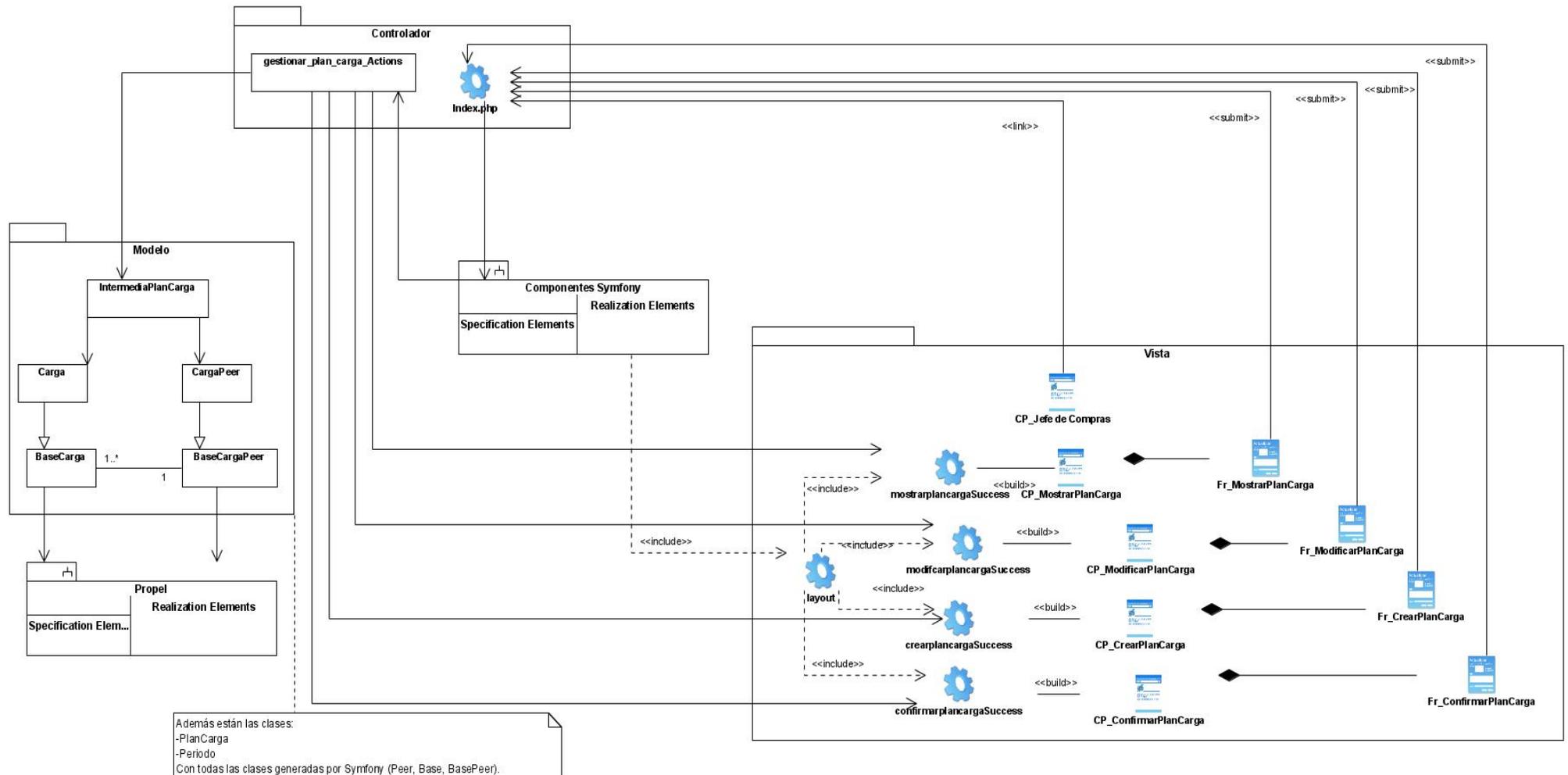


Fig. 40 Diagrama de clases del diseño (Web) del CU Gestionar Plan de Carga

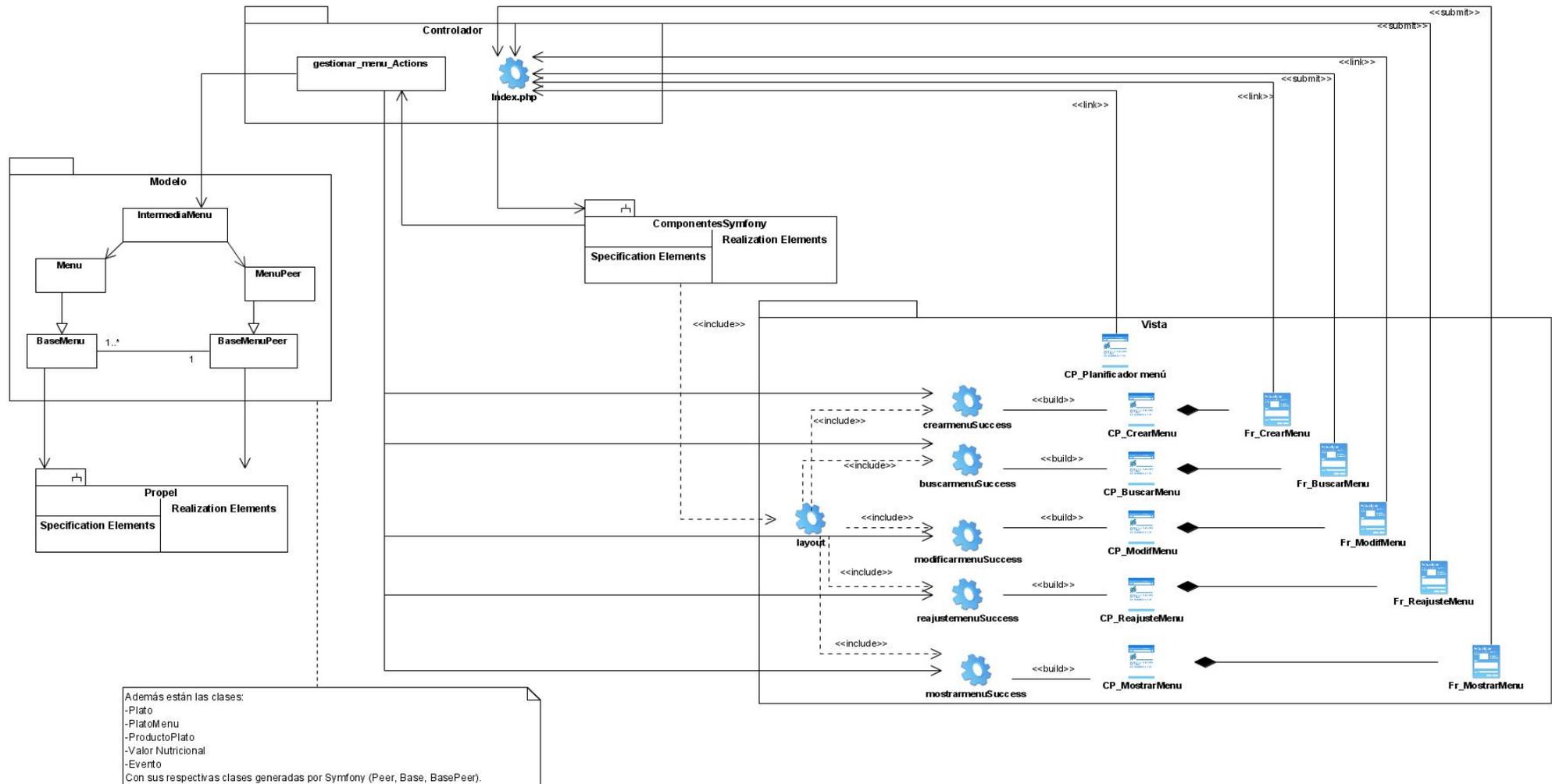


Fig. 41 Diagrama de clases del diseño (Web) del CU Gestionar menú

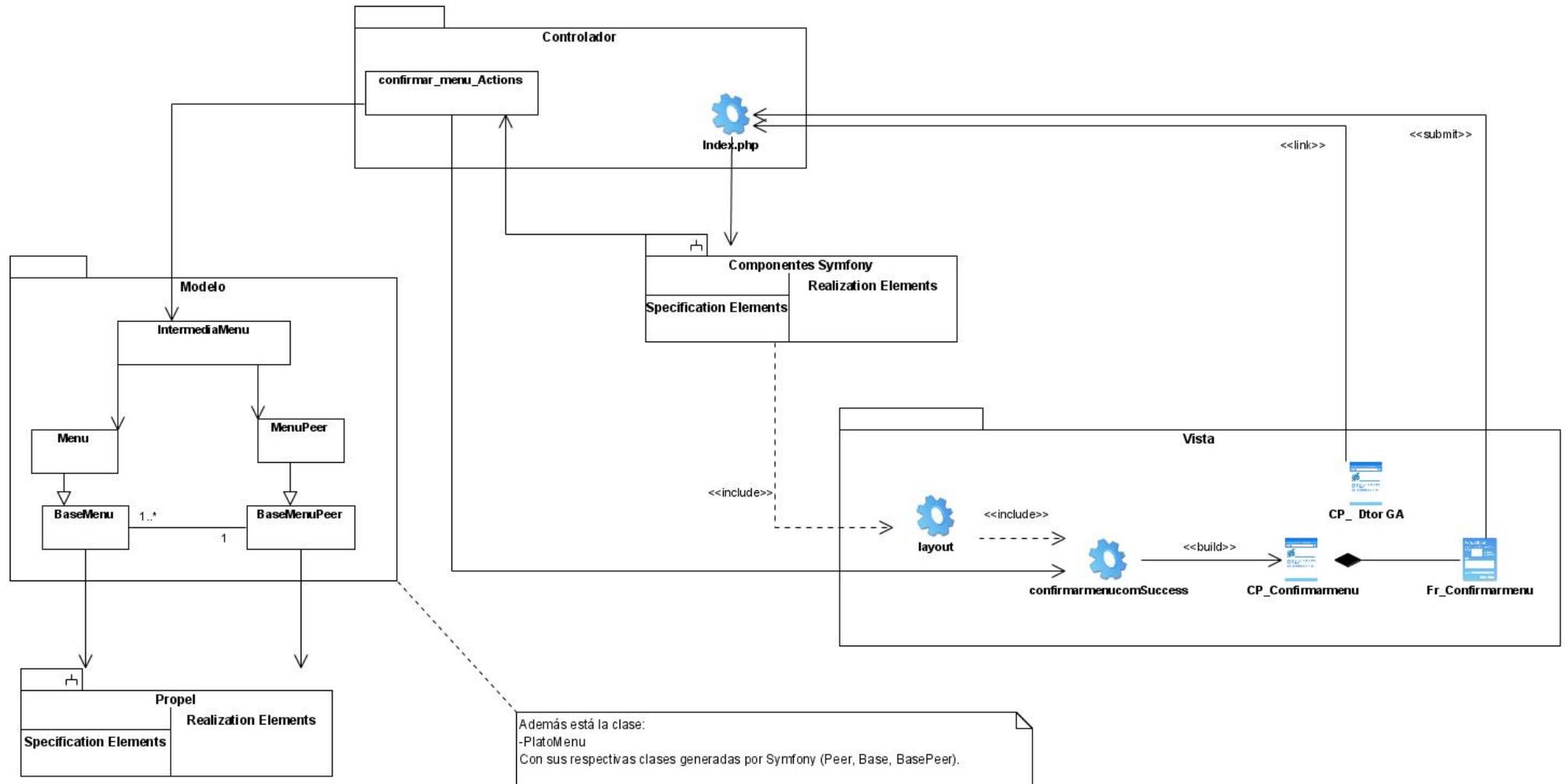


Fig. 42 Diagrama de clases del diseño (Web) del CU Confirmar menú

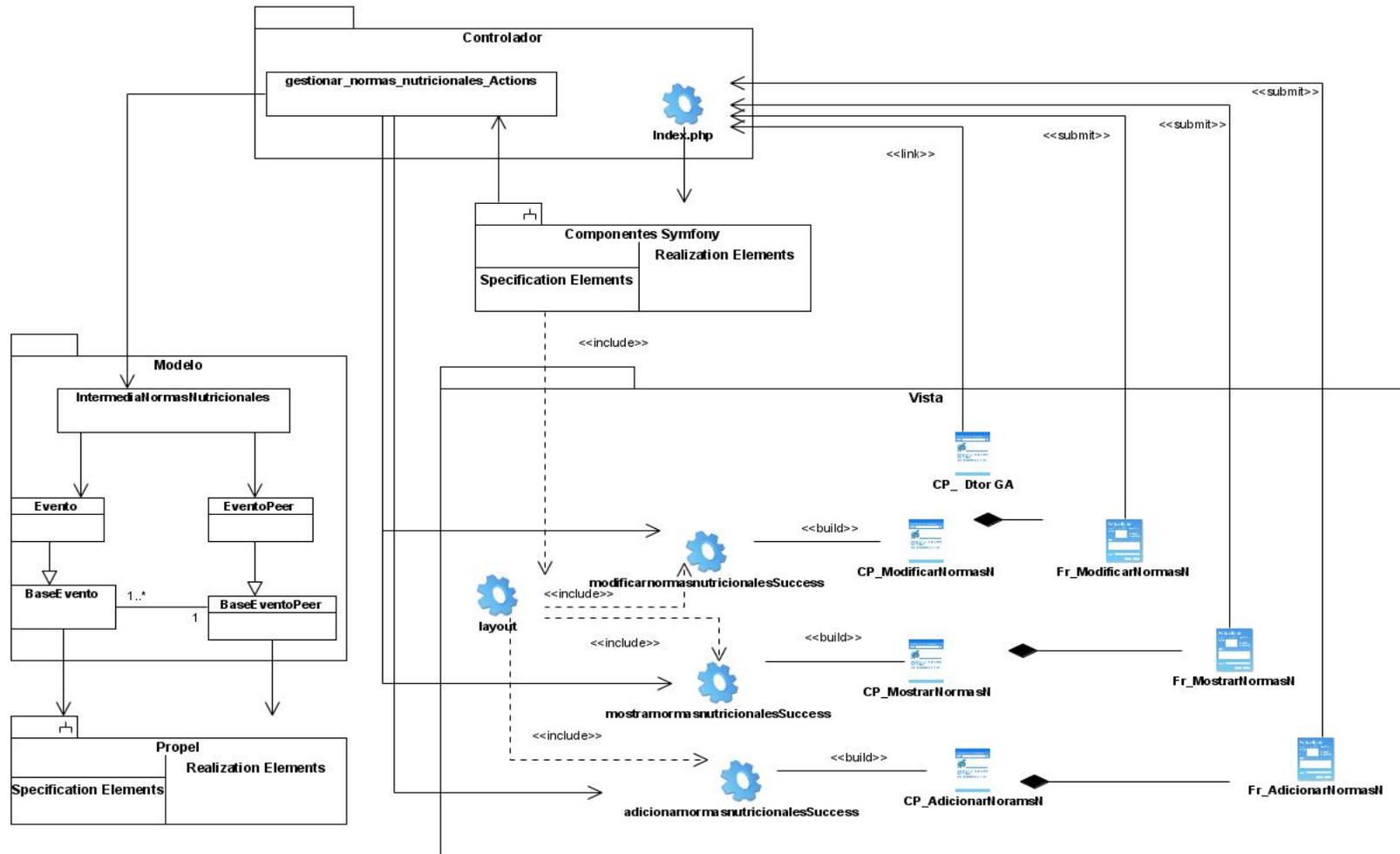


Fig. 43 Diagrama de clases del diseño (Web) del CU Gestionar Normas Nutricionales

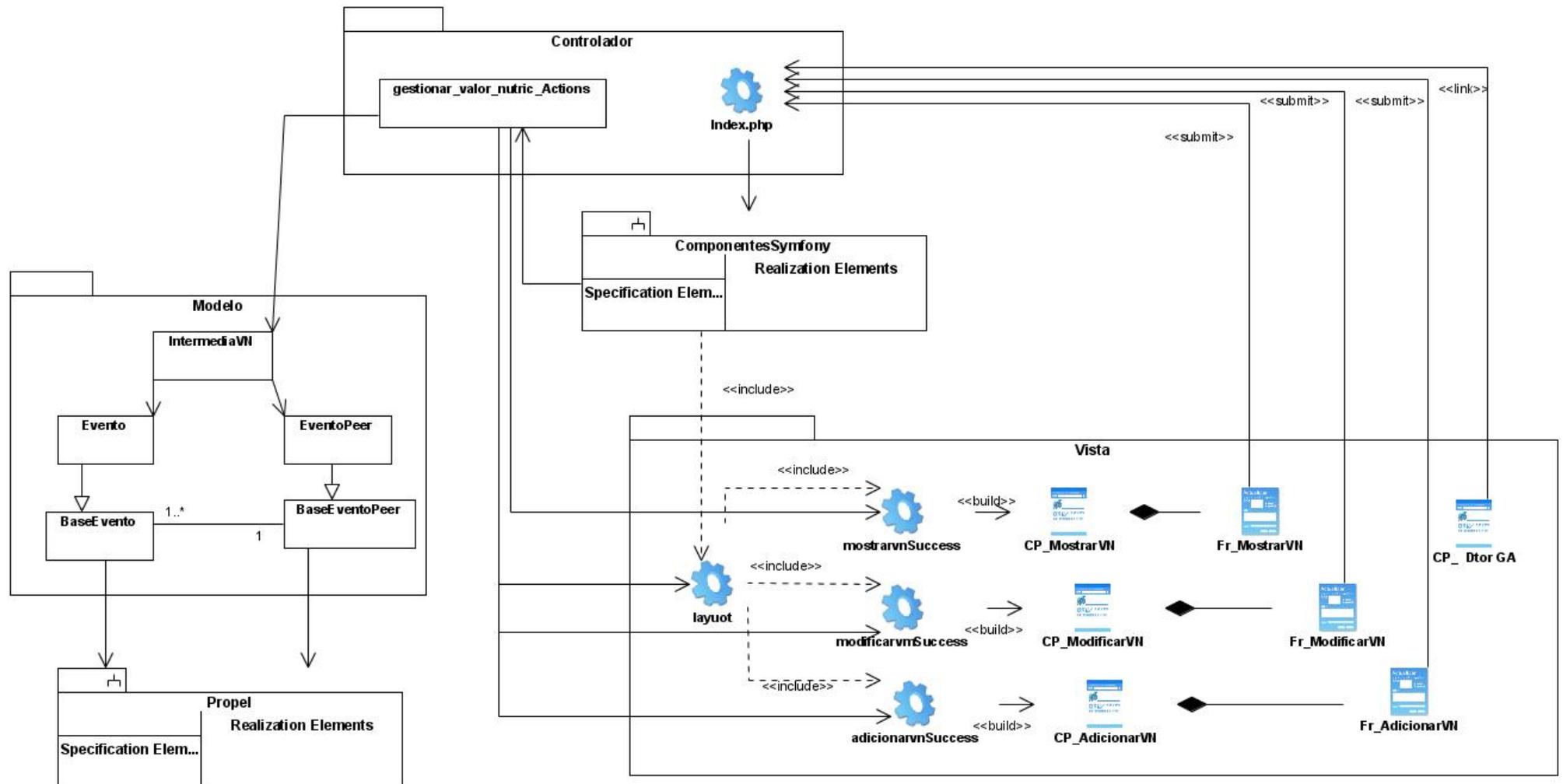


Fig. 44 Diagrama de clases del diseño (Web) del CU Gestionar Valor Nutricional

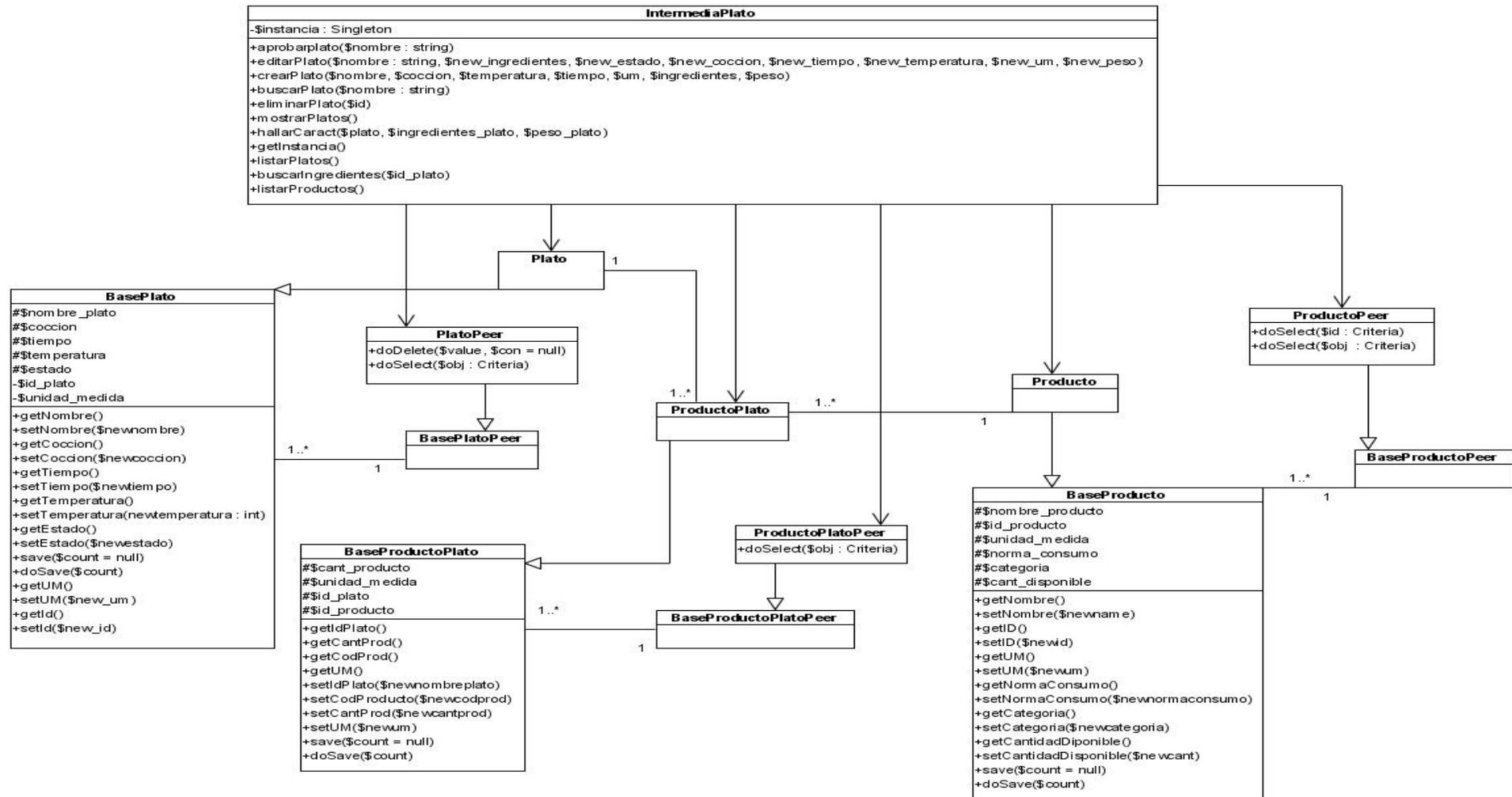


Fig. 45 Fragmento del paquete Modelo

3.5.4 Diagramas de Secuencia

En los diagramas de secuencia para aplicaciones Web es necesario considerar, sin importar el enfoque usado, que hay mensajes que son empleados simbólicamente para reflejar interacciones propias de la tecnología, tal es el caso de los vínculos, navegaciones, redireccionamientos, envíos de formulario y construcción de páginas (link, navigate, redirect, submit, build). Como consecuencia de la interacción entre el actor y la Aplicación Web, muchas de esas interacciones se ejecutan entre los distintos elementos que lo componen, su representación en un diagrama de secuencia contribuye a entender mejor su funcionamiento, colaborando con el objetivo de documentar la solución.

En este epígrafe se pondrán los diagramas de secuencia de dos casos de uso: Aprobar y Gestionar plato (Crear, Modificar, Eliminar, Mostrar y Buscar), los cuales ilustran el funcionamiento que tendrá el sistema ante alguna petición del actor.

En el caso del aprobar plato, por ejemplo, la línea es la siguiente:

El actor solicita en la página principal, la primera que se muestra luego de haberse autenticado, aprobar plato, dicha página como resultado hace un link al Index.php enviándole el nombre del módulo (plato) y la acción (executeListarplatos()) dentro de este que se solicitó. Luego el Index.php le envía esos datos al sfController, componente de Symfony, el cual se encarga de verificar la seguridad del sistema y de decirle a la clase Actions cual es la acción que debe ejecutar. La acción solicitada hace una llamada al método listarplatos() de la clase intermedia, encargada de la lógica del negocio, la misma es la responsable de buscar en la base de datos la información que se solicita, en este caso un listado con los platos que aún no han sido aprobados y le pasa este listado a la Actions, quien seguidamente le envía este listado a la aprobarplatoSuccess para que cree la *client page* CP_AprobarPlato y esta finalmente mostrará en un *Select* una lista de los platos sin aprobar.

Esta interacción tiene lugar cada vez que se hace una solicitud al sistema, es decir, luego de haberse seleccionado la opción que se quiera realizar, que el controlador frontal recibe la petición, se verifica la seguridad y si el usuario tiene los permisos para acceder a la información solicitada se ejecutan una serie de acciones, en las cuales la diferencia de los mensajes se debe las distintas peticiones con las que se debe crear la *client page*, cuando es necesario buscar en la base de datos y cuando no se requieren datos de la misma. Es por ello que se plantea que una vez que se identifica el mecanismo a emplear, se debe documentar una sola vez y de forma general, pues repetir lo mismo en cada diagrama resultaría redundante y cargaría el diagrama con mensajes y objetos que siempre serán los mismos.

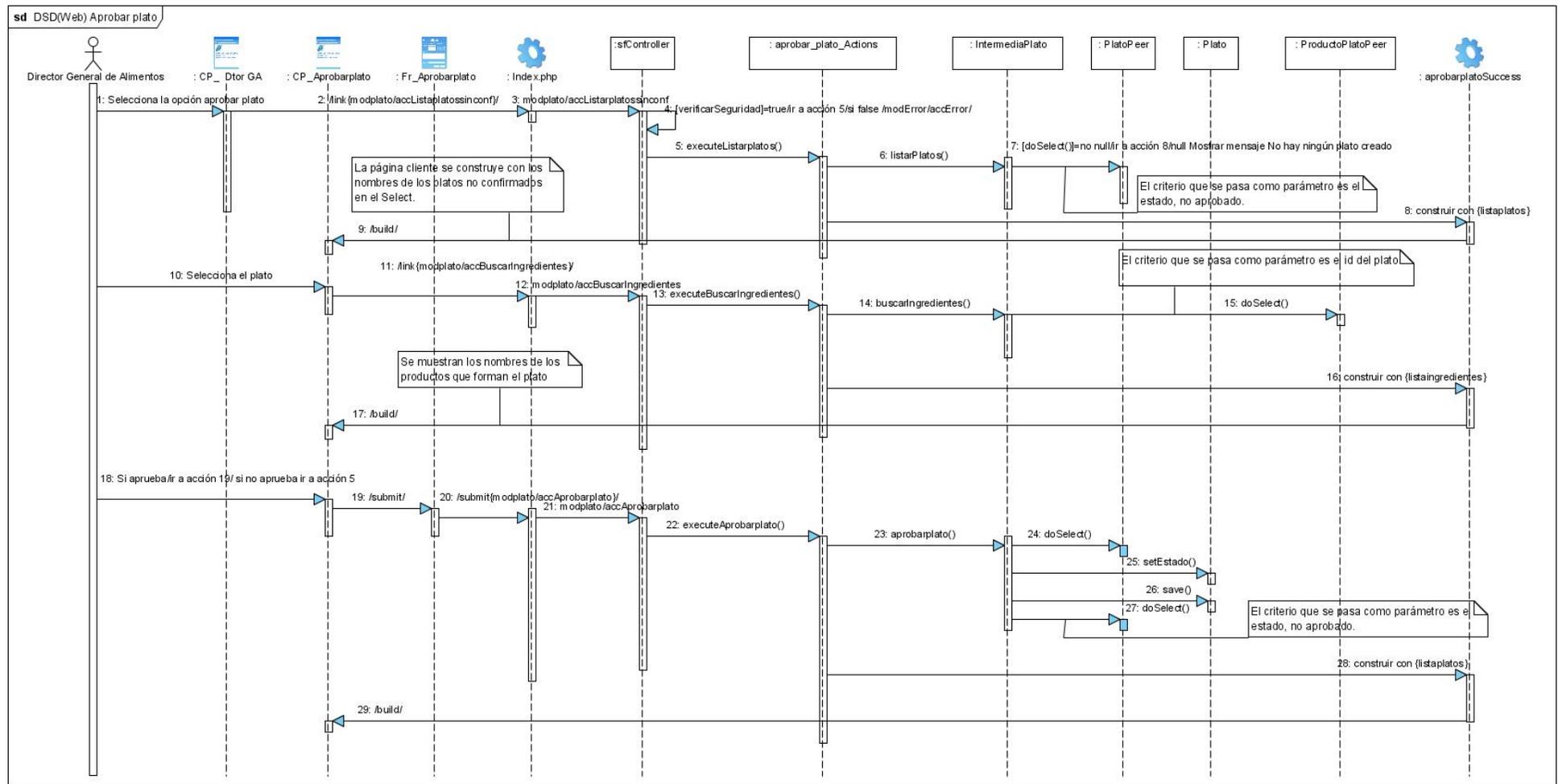


Fig. 46 Diagrama de secuencia del CU Aprobar plato

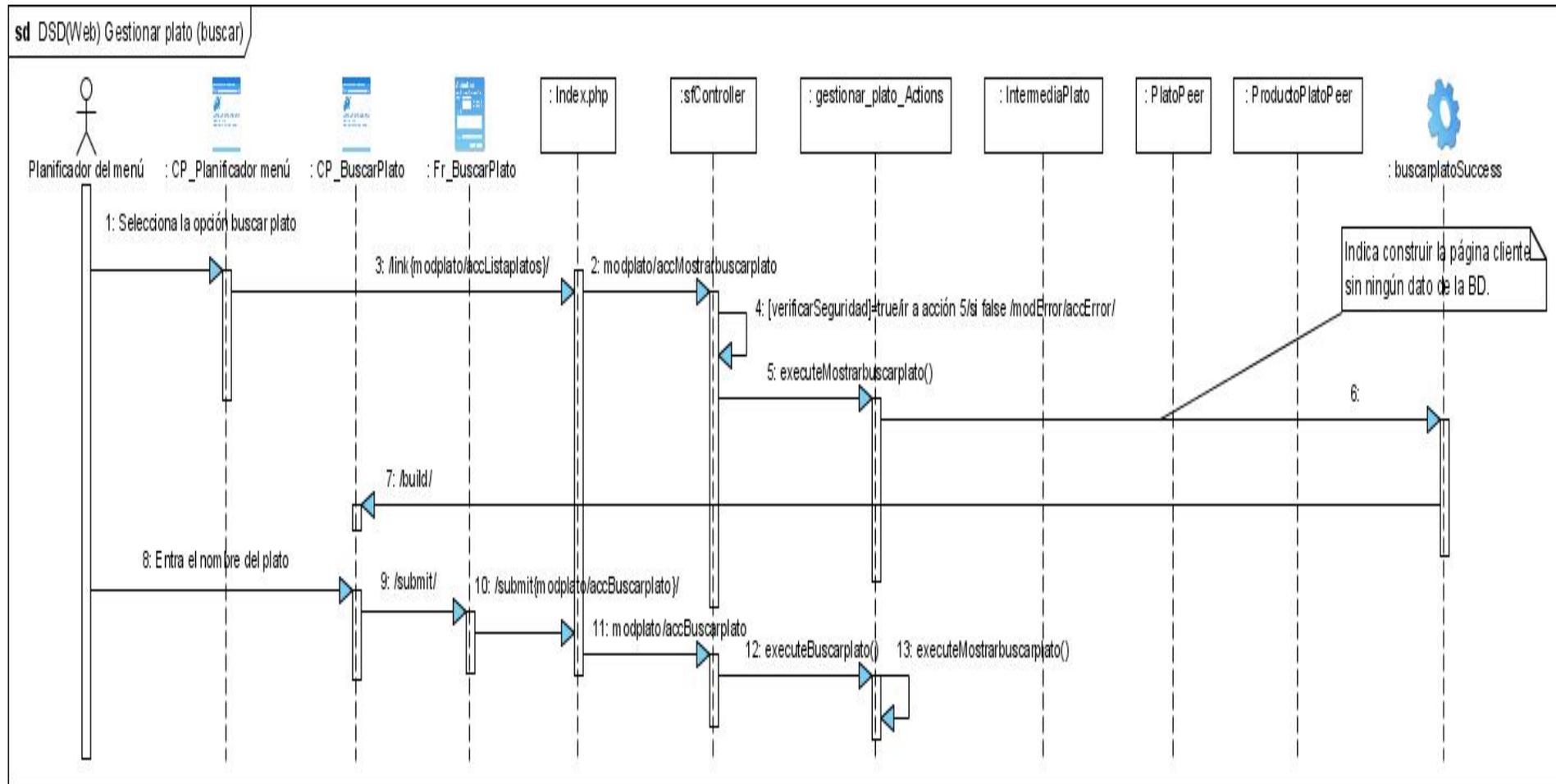


Fig. 47 Diagrama de secuencia del CU Gestionar plato: Esc. Buscar

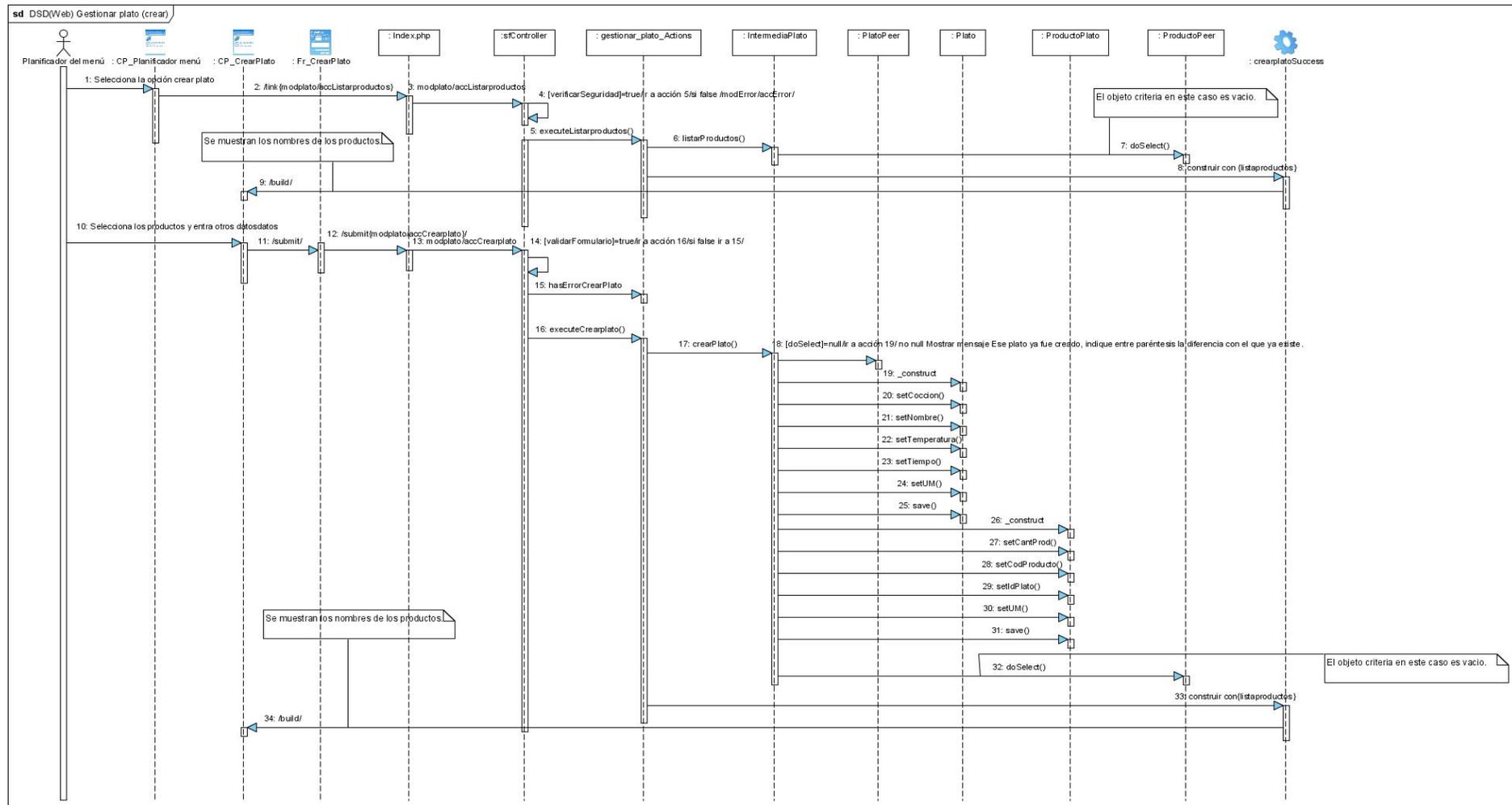


Fig. 48 Diagrama de secuencia del CU Gestionar plato: Esc. Crear

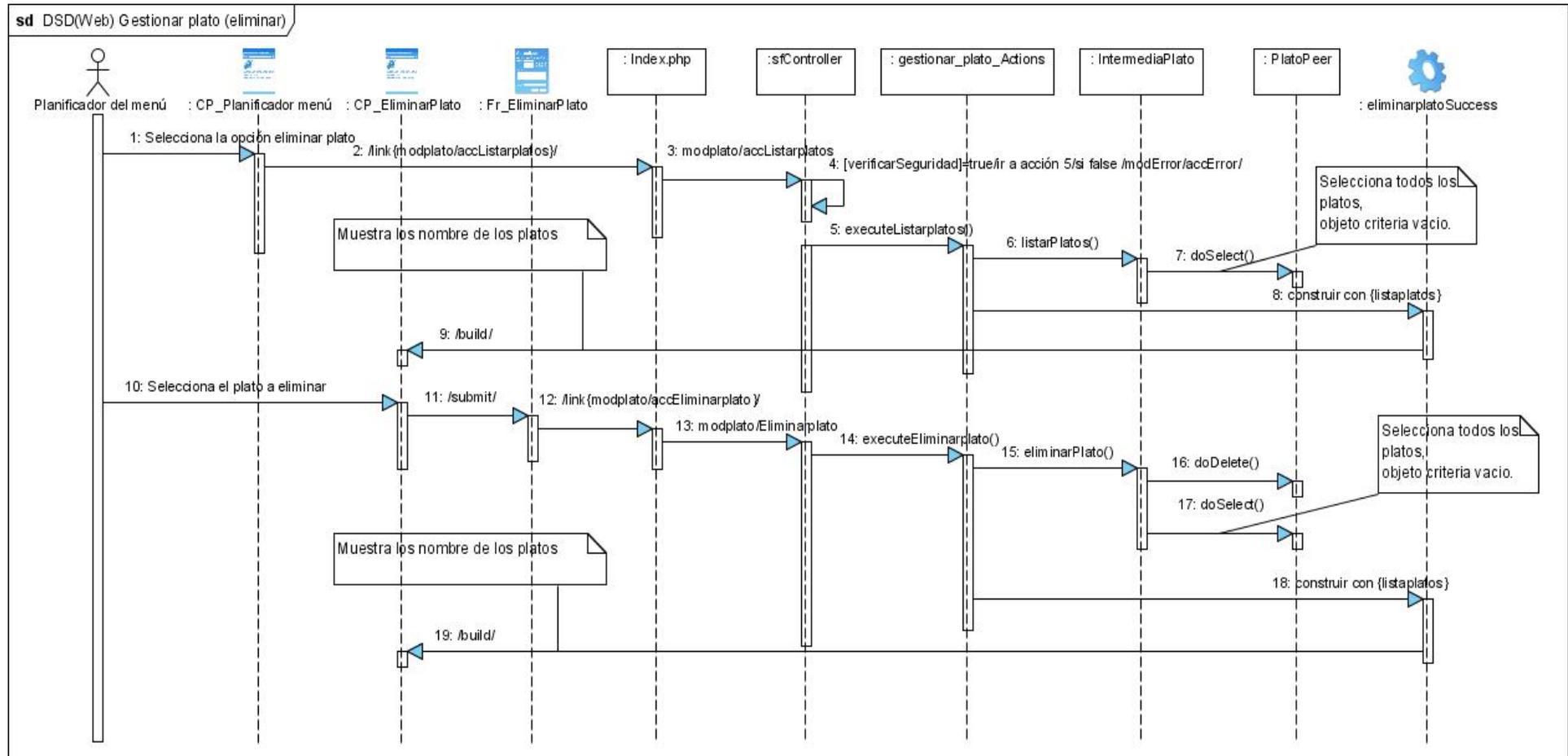


Fig. 49 Diagrama de secuencia del CU Gestionar plato: Esc. Eliminar

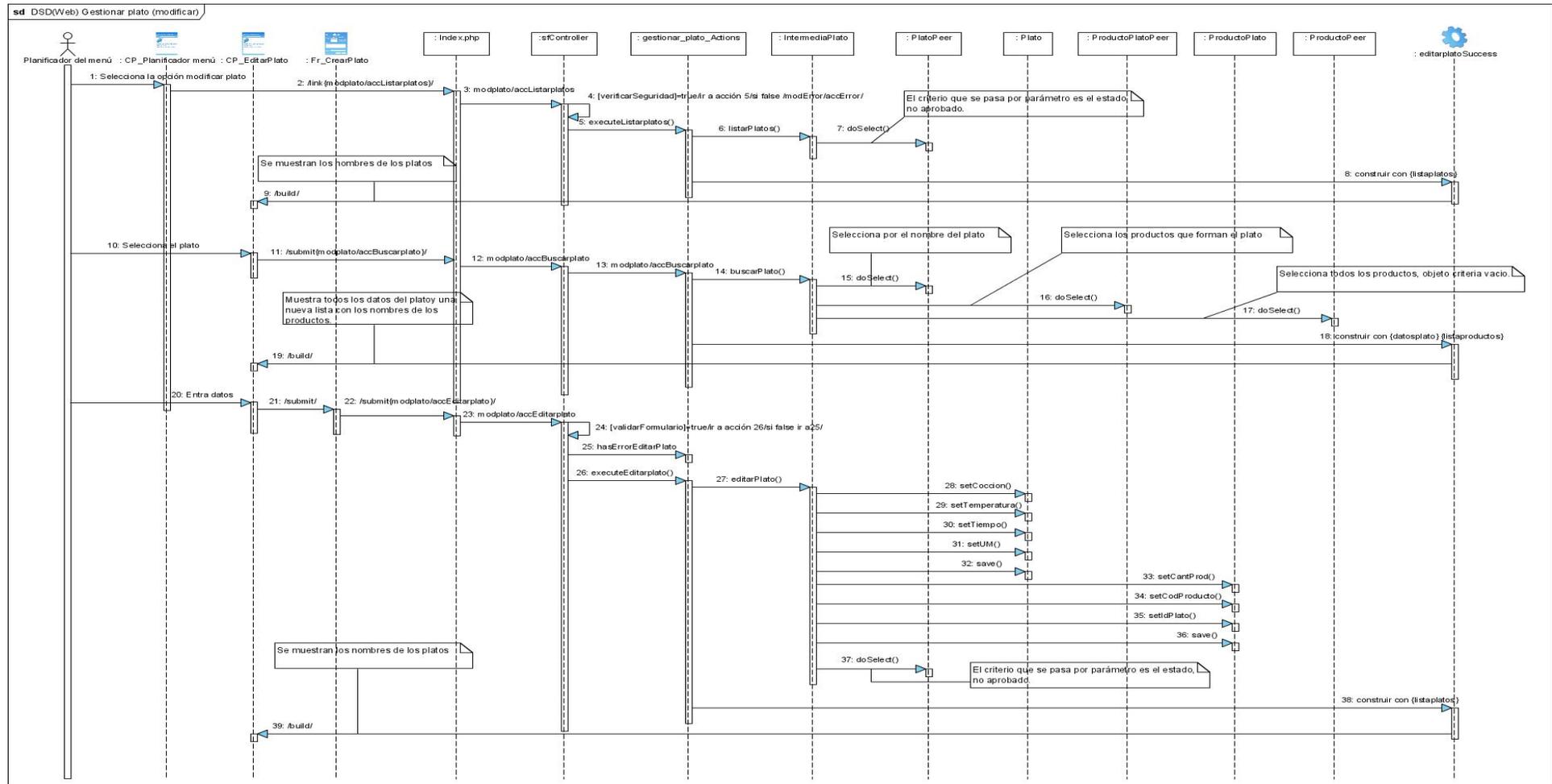


Fig. 50 Diagrama de secuencia del CU Gestionar plato: Esc. Modificar

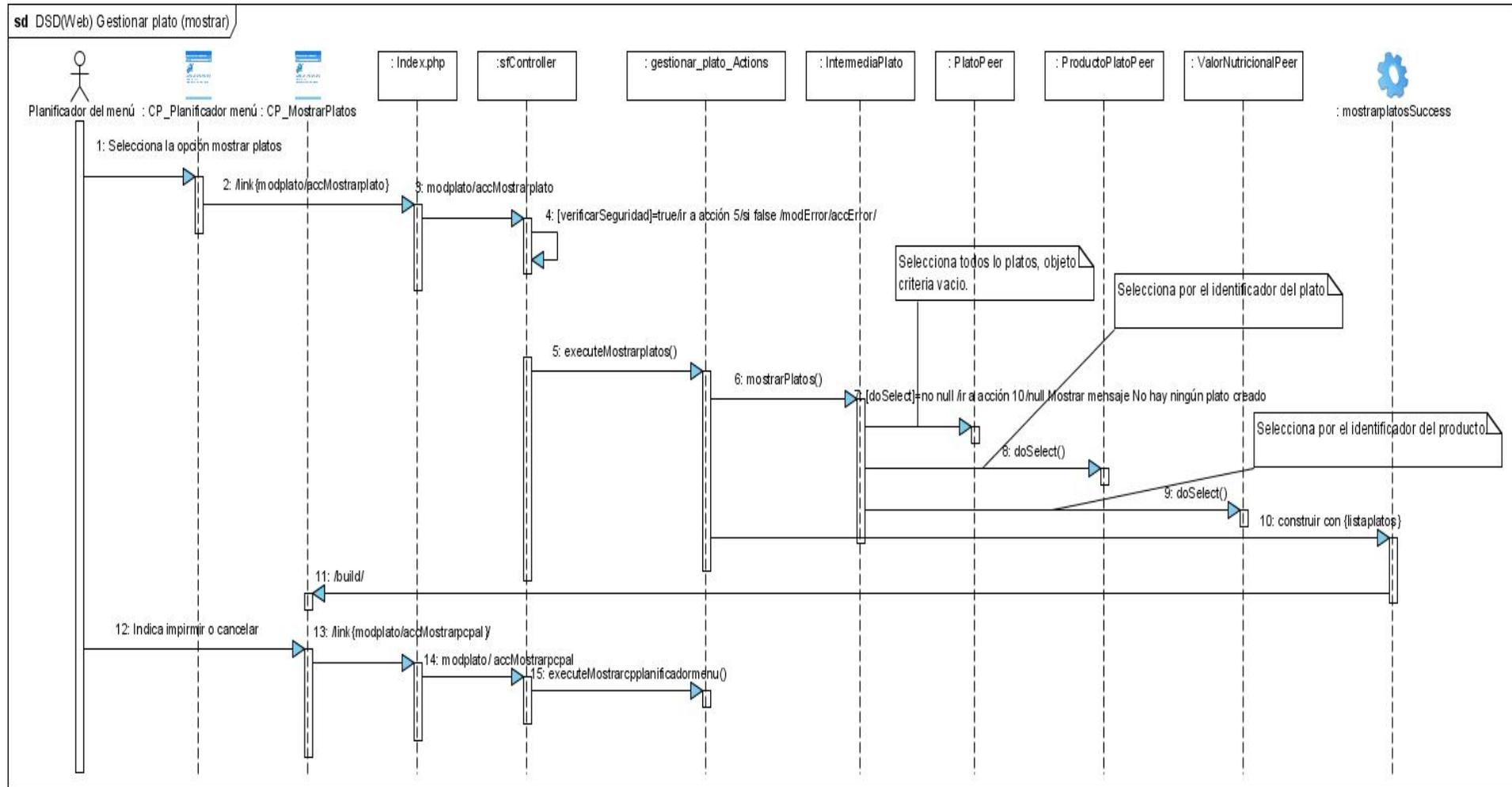


Fig. 51 Diagrama de secuencia del CU Gestionar plato: Esc. Mostrar

Para ver los restantes diagramas de secuencia remitirse al expediente de proyecto.

3.6 Validación del diseño realizado

Una vez terminado el diseño de un software es necesario validar la calidad del mismo, pues permite comprobar si el objetivo de este flujo de trabajo fue cumplido satisfactoriamente y si la implementación del sistema se puede realizar a partir de los resultados obtenidos. Para ello existen varias técnicas de validación, entre ellas tenemos:

- Evaluación de la calidad de las clases diseñadas a través de la generación del código con la herramienta CASE de modelado para comprobar si este es de utilidad para los programadores: este método no puede ser utilizado para validar el diseño del Módulo Gestión del menú debido a las características que posee el diagrama de clases del diseño realizado, pues en este se tiene un paquete genérico (Modelo) para evidenciar las relaciones entre las diferentes clases del mismo, por lo que el código que se generaría no sería de mucha utilidad para los programadores del proyecto.
- Realizar una comparación entre lo implementado y lo modelado: este procedimiento no puede ser aplicado pues en el proyecto aún no se ha comenzado con la implementación.
- Realizar encuestas y/o entrevistas a los programadores con el objetivo de conocer el grado de satisfacción que los mismos adquirieron como resultado de un análisis del diseño realizado: este método es uno de los más conocidos y utilizados, siendo precisamente uno de los escogidos por los analistas del proyecto Alimentos UCI.

La encuesta aplicada a los cuatro programadores consta de 7 aspectos, arrojando los siguientes resultados:

Los prototipos no funcionales abarcan todas las funcionalidades identificadas para la aplicación, de igual manera existe correspondencia con los requerimientos no funcionales al ser las interfaces sencillas, con colores claros, sin sobrecarga de imágenes, además cada usuario que se autentifique en el sistema tendrá acceso solamente a la página que le corresponde.

Todos los programadores han estudiado los diagramas de clases del diseño y secuencia realizados entendiendo correctamente el flujo de mensajes que se representa en los mismos, considerándolos además legibles y apropiados para comenzar con la implementación del sistema.

3.7 Conclusiones

En este capítulo se identificaron las clases del diseño correspondientes a la aplicación Web, además el uso del framework Symfony y con el uso del patrón Modelo-Vista-Controlador han facilitado considerablemente realizar el diseño de estas clases las cuales serán las que se implementarán para darle cumplimiento a las funcionalidades que debe tener el sistema.

Conclusiones Generales

Una vez finalizada esta investigación se puede concluir que:

- Se realizó el análisis de los procesos del negocio relacionados con la planificación y control del servicio de comedores y dentro de este en la gestión del menú, logrando un mayor nivel de entendimiento de las actividades que se desarrollan en el mismo e identificando las posibles a automatizar, lo que permitió la obtención de los artefactos correspondientes al flujo de trabajo del modelamiento del negocio lográndose una mejor comunicación con el cliente.
- Se logró identificar las funcionalidades que debía tener el Módulo “Gestión del menú”, a través de los requerimientos funcionales y los no funcionales, identificándose además los CU del sistema realizando una descripción detallada de los mismos.
- Se diseñaron las clases necesarias para la implementación del Módulo Gestión del menú del sistema de gestión integral de la planificación y control del servicio de comedores en la UCI, lográndose que las mismas cumplieran con los principios y patrones de diseño orientados a objetos y se validaron estas a través de una encuesta realizada a los programadores del proyecto.

Recomendaciones

- Refinar el diseño propuesto para la implementación, por ejemplo en el caso del crear menú que el listado de platos que se muestre dependa del evento para el cual se está planificando ese menú.
- Integrar los módulos Gestión del menú y Gestión de Comensales en el flujo de trabajo de implementación, debido a la gran relación existente entre ellos ya que la mayoría de los procesos que se llevan a cabo en el segundo módulo dependen de los que se realizan en el primero.

Referencias bibliográficas

- (1). **Zavala.** *angelfire. angelfire.* [En línea] 08 de 09 de 2002. [Citado el: 01 de 12 de 2007.] <http://www.angelfire.com/scifi/jzavalar/apuntes/IngSoftware.html>.
- (2). **Sanchez, María A. Mendoza.** *Informatizate. Informatizate.* [En línea] 7 de 06 de 2004. [Citado el: 07 de 12 de 2007.] (Mendoza, 2004)
- (3). Teleformación. *Entorno Virtual de Aprendizaje.* [En línea] 16 de 09 de 2007 [Citado el: 07 de 12 de 2007.] <http://teleformacion.uci.cu/mod/resource/view.php?id=6655>
- (4). **Ugaz, Arq. Max.** Si (Ugaz) (Larman) (Zaninotto, 2005) tío de la Facultad de Ingeniería y Arquitectura . *Proyectos.* [En línea] [Citado el: 09 de 12 de 2007.] <http://www.usmp.edu.pe/publicaciones/boletin/fia/info36/proyectos.html>.
- (5) **ATOS, I. T.** (04 de 03 de 2008). *INES.* [Citado el 31 de 03 de 2008.] de INES: <http://www.ines.org.es/vulcano/wp-content/uploads/2007/09/d6-estudio-de-herramientas-de-certificacion-tid.pdf>
- (6). Sparx Systems. *Sparx Systems.* [En línea] [Citado el: 01 de 04 de 2008.] <http://www.sparxsystems.com.ar/products/ea.html>.
- (7) Sistemas Phoenix S. de R.L. [En línea] [Citado el: 01 de 04 de 2008.] <http://www.campsoft.com.mx/Studio%20MX%202004.htm>
- (8) VitaminaWEB.com. [En línea] [Citado el: 01 de 04 de 2008.] <http://www.vitaminaweb.com/html/articulos/articulo02.php>
- (9). **Oktaba, Hanna.** Introducción a Patrones. *Introducción a Patrones.* [En línea] [Citado el: 24 de 04 de 2008.] <http://www.mcc.unam.mx/~cursos/Algoritmos/javaDC99-2/patrones.html>.
- (10). **Zaninotto, Fabien Potencier y François.** *Symphony la guía definitiva.* s.l. : Apress (ISBN-13: 978-1590597866), 2005.
- (11). **Larman, C.** *UML y Patrones, Tomo I, Capítulo 18.*
- (12). **Febles, Y. E. A. C. y Y. R.** (2005). *Sistema Informático para la Gestión Integral de Comedores. Ciudad Habana, ISPJAE.* (Zavala, 2002)
- (13). Teleformación. *Entorno Virtual de Aprendizaje.* [En línea] (Teleformación, 2007) 22 del 10 del 2007. [Citado el: 13 de 05 de 2008.] <http://teleformacion.uci.cu/mod/resource/view.php?id=8865>.
- (14). SVS NEWSLETTER. *Sociedad de Validación de Sistemas S.L.* [En línea] SVS S.L, 05 de 2006. [Citado el: 17 de 04 de 2008.] <http://www.svshome.com/newsletter/boletin03.html>.
- (15). Teleformación. *Entorno Virtual de Aprendizaje.* [En línea] (Teleformación, 2007) 22 del 10 del 2007. [Citado el: 16 de 03 de 2008.] <http://teleformacion.uci.cu/mod/resource/view.php?id=8879>.

- (16). *arquitecturadeinformacion.cl*. (2004). Recuperado el 05 de 06 de 2008, de *arquitecturadeinformacion.cl*: <http://www.arquitecturadeinformacion.cl/como/mapa2.html>
- (17). *RIVED/HUASCARÁN*. (s.f.). Recuperado el 05 de 06 de 2008, de *RIVED/HUASCARÁN*: http://portal.huascararan.edu.pe/modulos/m_taller/mapa.htm
- (18). Teleformación. *Entorno Virtual de Aprendizaje*. [En línea] (Teleformación, 2008) 28 del 02 del 2008. [Citado el: 30 de 05 de 2008.] <http://teleformacion.uci.cu/mod/resource/view.php?id=21363>
- (19). **Marzo, J. V.** (2007). *vico.org open modeling*. Recuperado el 05 de 06 de 2008, de *vico.org open modeling*: <http://www.vico.org>
- (20). **Sarmiento Almenares, A., & Cutiño Díaz, E.** (2006). *LIMS DE CALIDAD DEL CENTRO DE INGENIERÍA GENÉTICA Y BIOTECNOLOGÍA: ANÁLISIS DEL GRUPO DE RECEPCIÓN DE MUESTRAS Y MANIPULACIÓN DE EXPEDIENTES*.
- (21). **Torres, Sandry Hernández.** *Presentación en el forum Sandry, SISC UCI*.

Bibliografía

- Larman, C. UML y Patrones. En C. Larman, *UML y Patrones* (págs. Tomo I, Capítulo 18).
- Teleformación. Entorno Virtual de Aprendizaje. [En línea] 16 de 09 de 2007.
- Zaninotto, F. P. (2005). *Symfony la guía definitiva*. s1.: Apress (ISBN-13:978-1590597866).
- Febles, Y. E. A. C. y Y. R. (2005). *Sistema Informático para la Gestión Integral de Comedores. Ciudad Habana, ISPJAE.*
- Sparx Systems. Sparx Systems. [En línea] <http://www.sparxsystems.com.ar/products/ea.html>
- Sistemas Phoenix S. de R.L. [En línea] <http://www.campusoft.com.mx/Studio%20MX%202004.htm>
- VitaminaWEB.com. [En línea] <http://www.vitaminaweb.com/html/articulos/articulo02.php>
- Alvarez, Miguel Angel. desarrolloweb. desarrolloweb. [En línea] <http://www.desarrolloweb.com/articulos/332.php>.
- Gamma, H. J. *Design Patterns, Elements of Reusable Object Oriented Software.*
- Torres, S. H. *Propuesta del diseño del sistema de gestión integral de la planificación y control del servicio de comedores en la UCI.*
- Educación, Ministerio de. *El Autoabastecimiento Escolar y la Actividad Laboral: Curso para directores de centros internos*. s.l. : Ministerio de Educación, 2006.

Anexos

Anexo 1

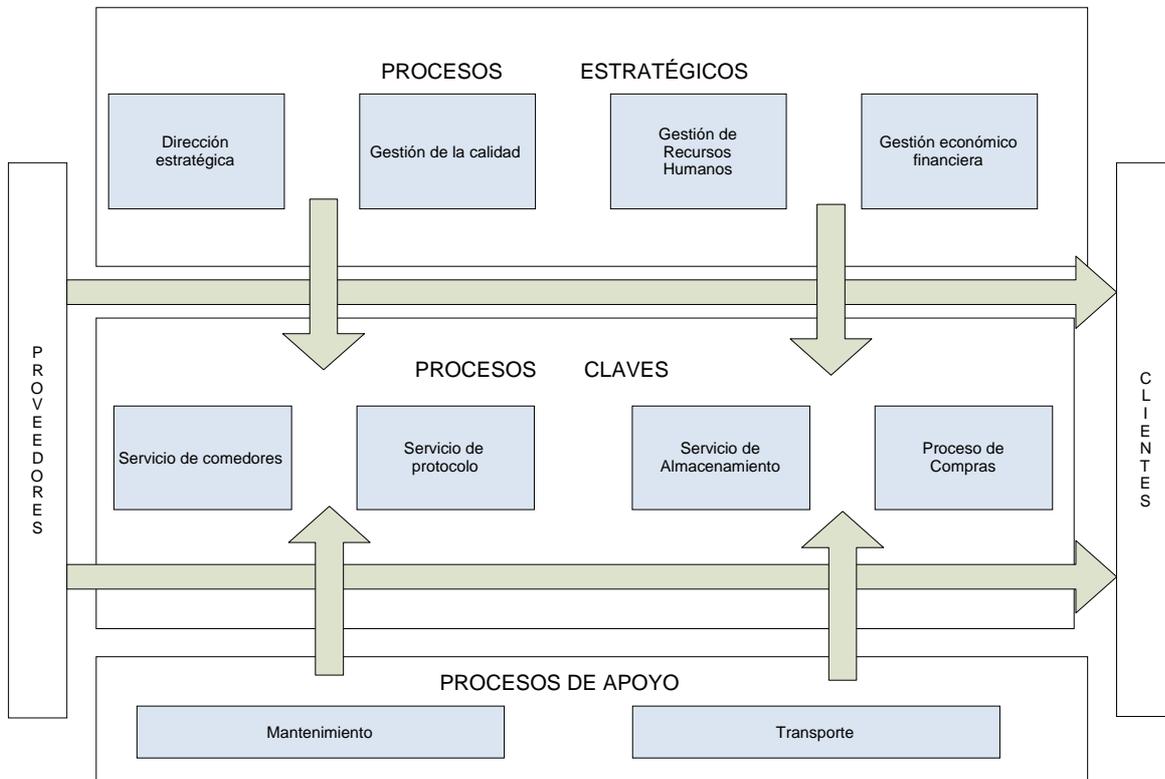


Fig. 52 Mapa de procesos logísticos (21)

Anexo 2

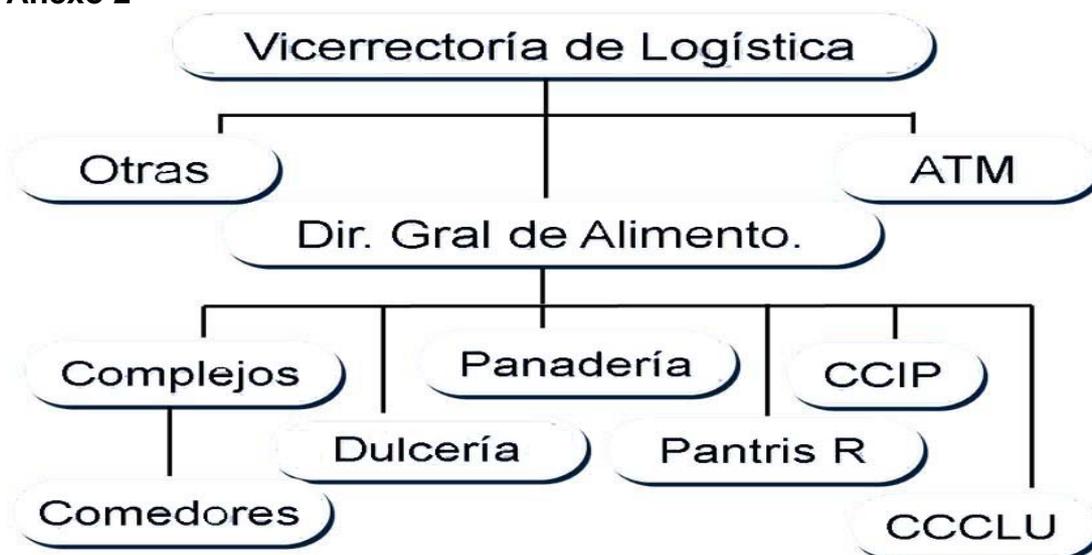


Fig. 53 Organigrama de la Vicerrectoría de Logística
Anexo 3

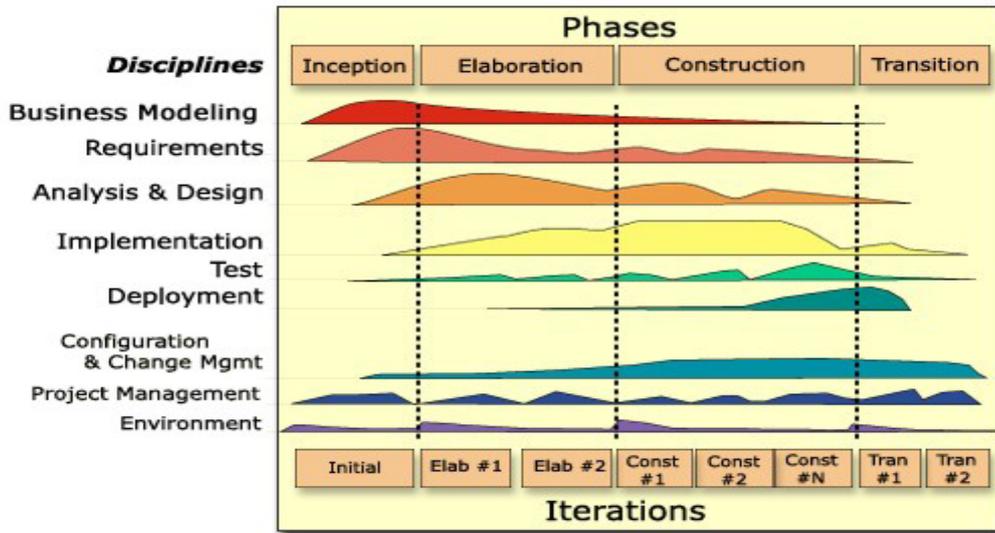


Fig. 54 Gráfica de RUP

Anexo 4

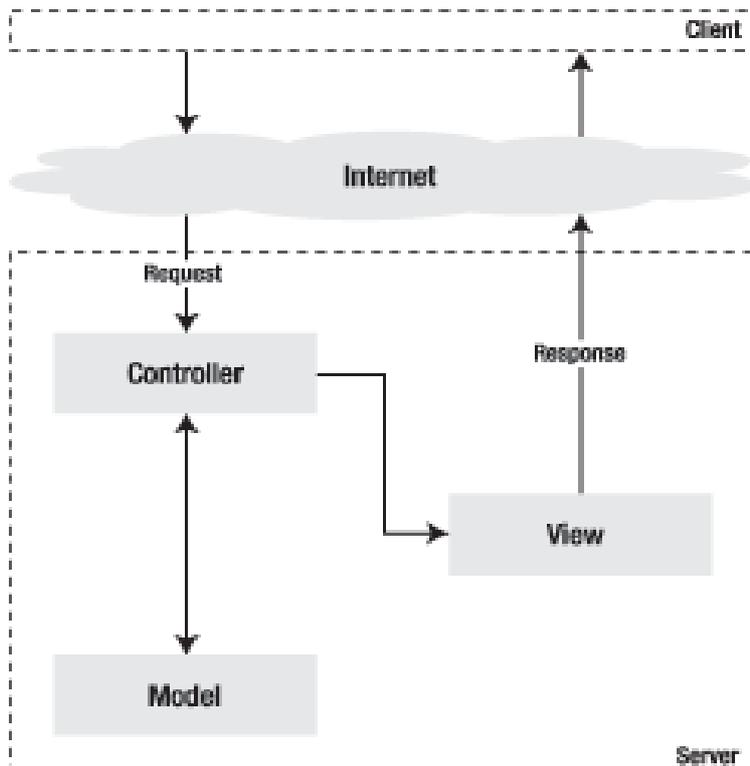


Fig. 55 Representación del Patrón Modelo Vista Controlador

Anexo 7

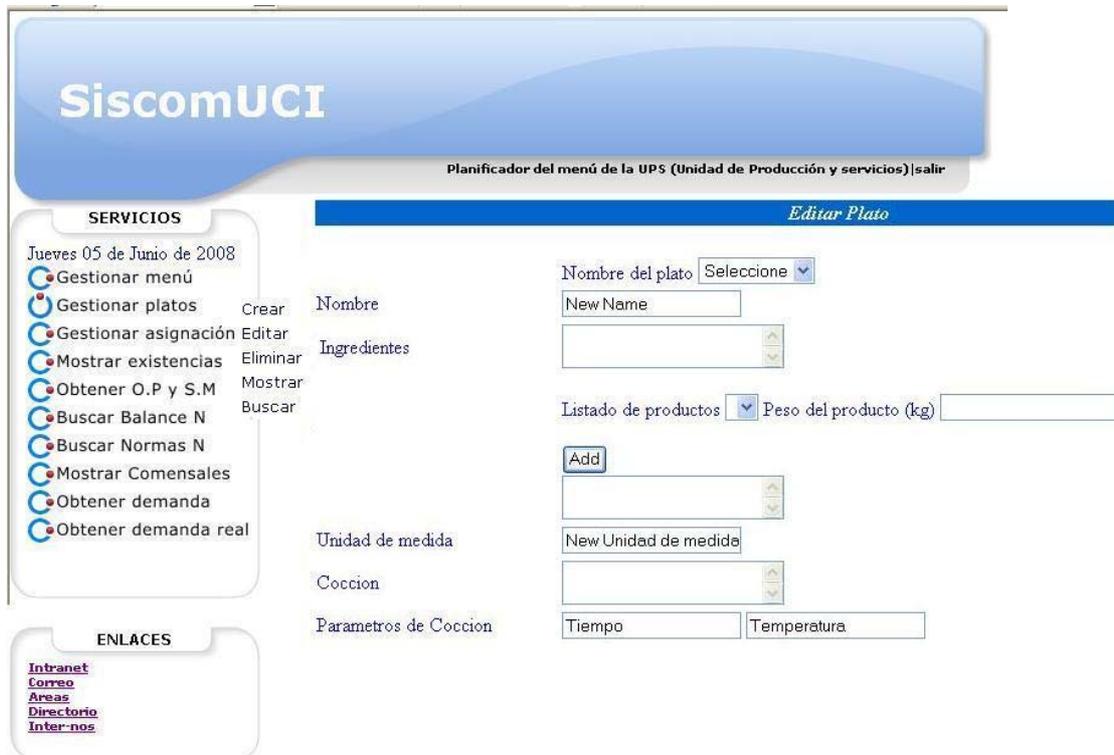


Fig. 58 Prototipo no funcional del CU Gestionar Plato. Sección "Editar Plato"

Anexo 8

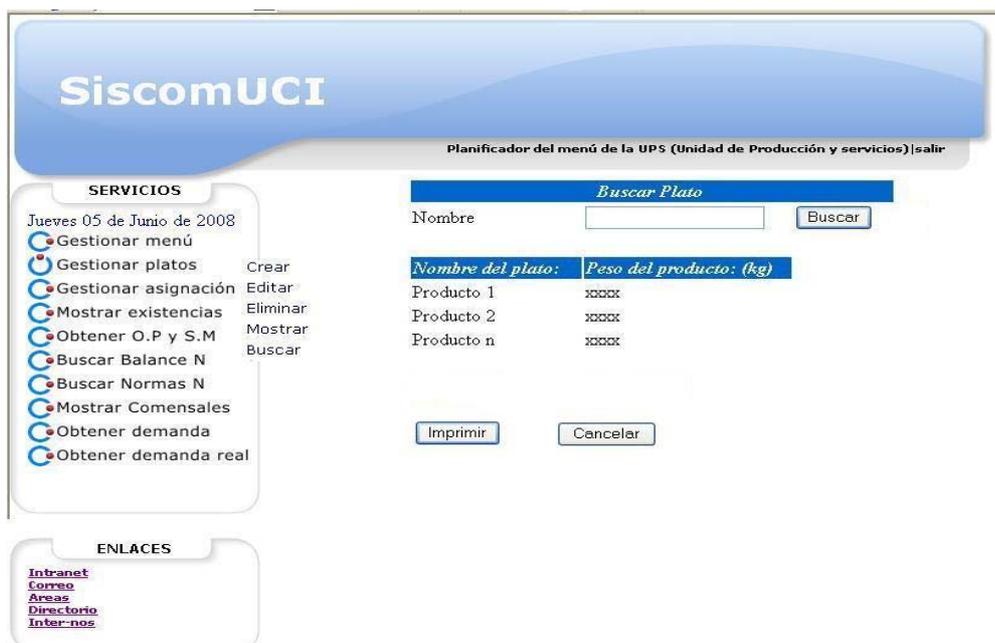


Fig. 59 Prototipo no funcional del CU Gestionar Plato. Sección "Buscar Plato"

Anexo 9

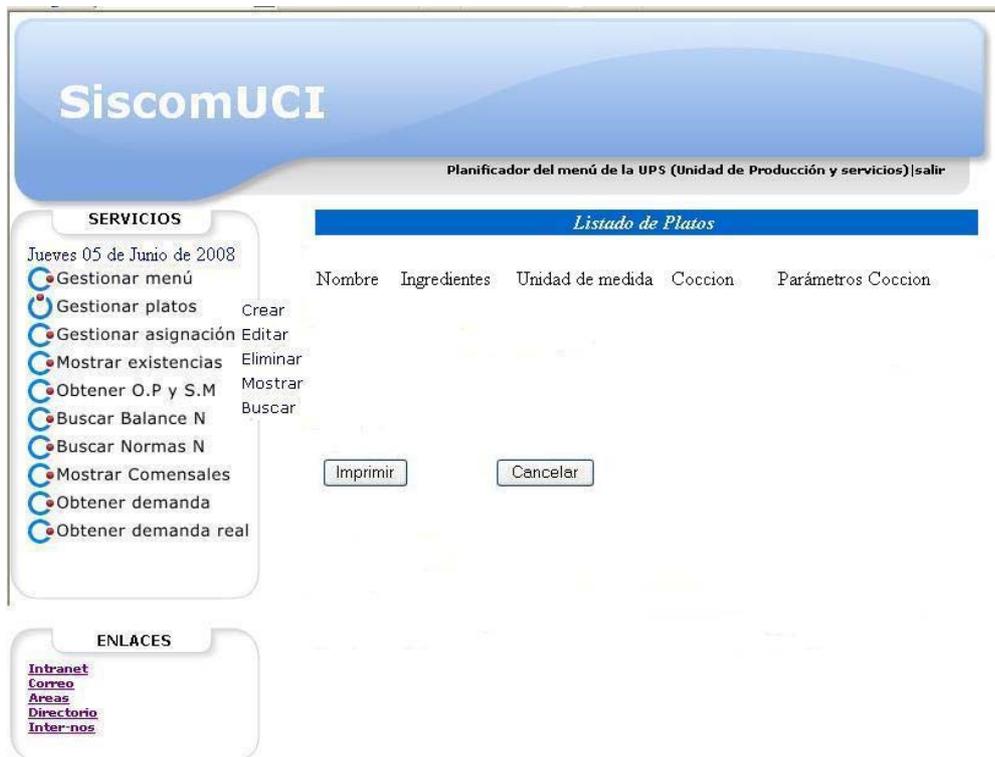


Fig. 60 Prototipo no funcional del CU Gestionar Plato. Sección "Mostrar Plato"

Anexo 10



Fig. 61 Prototipo no funcional del CU Gestionar Plato. Sección "Eliminar Plato"

Anexo 11

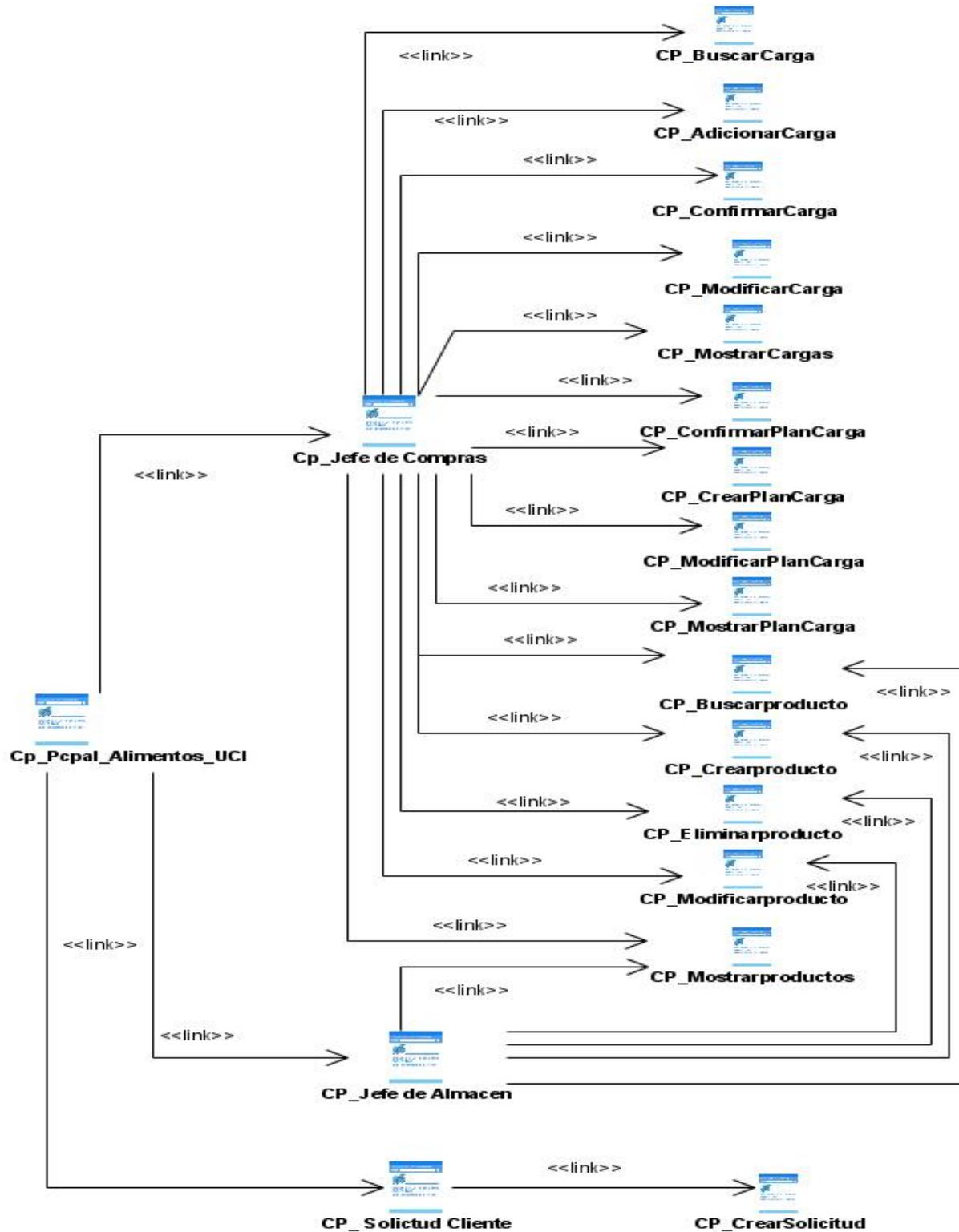


Fig. 62 Mapa de Navegación 1 (Para el Jefe de Almacén, el Jefe de Compras y el Solicitud Cliente)

Anexo 12

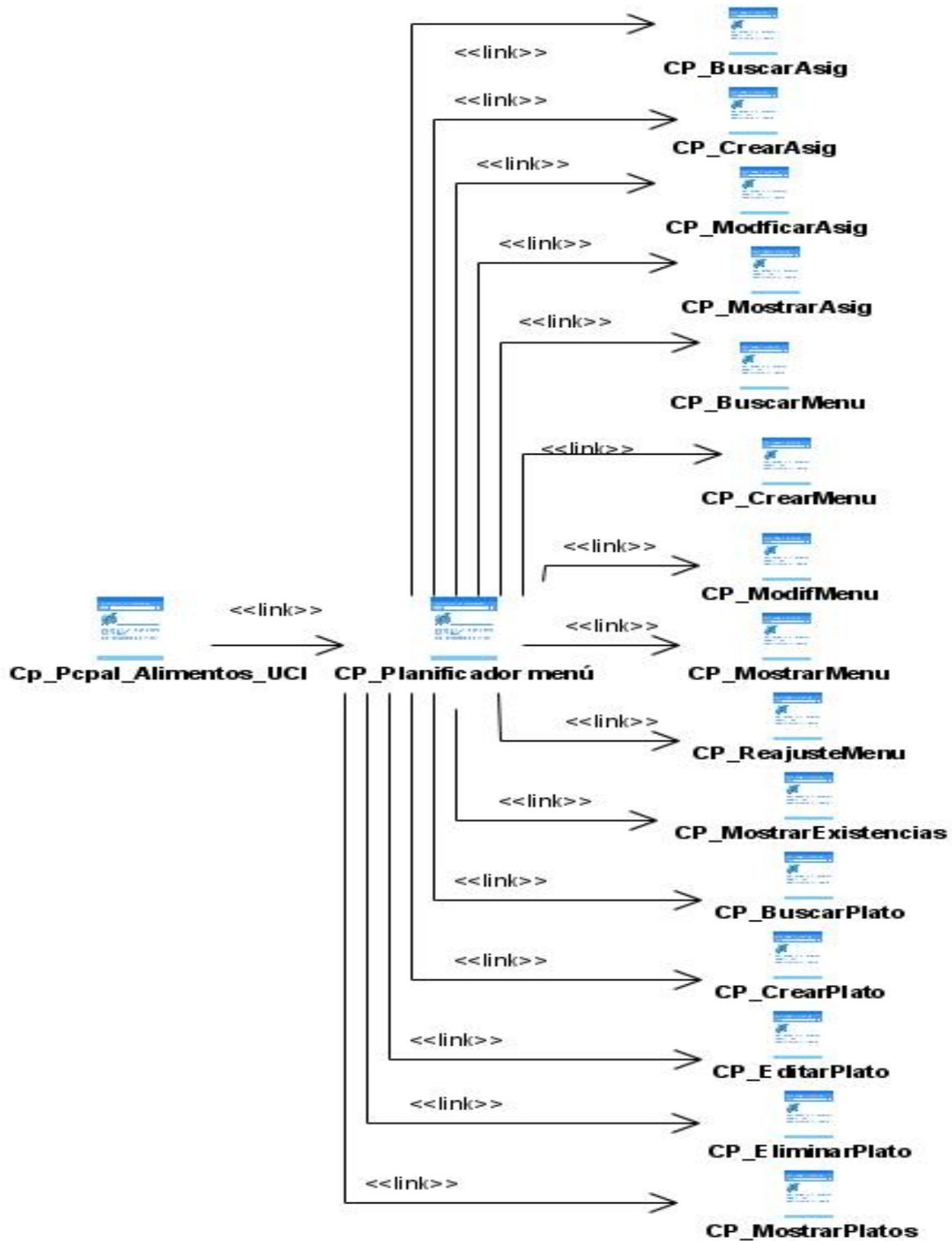


Fig. 63 Mapa de Navegación 2 (Para el Planificador del menú)

Anexo13

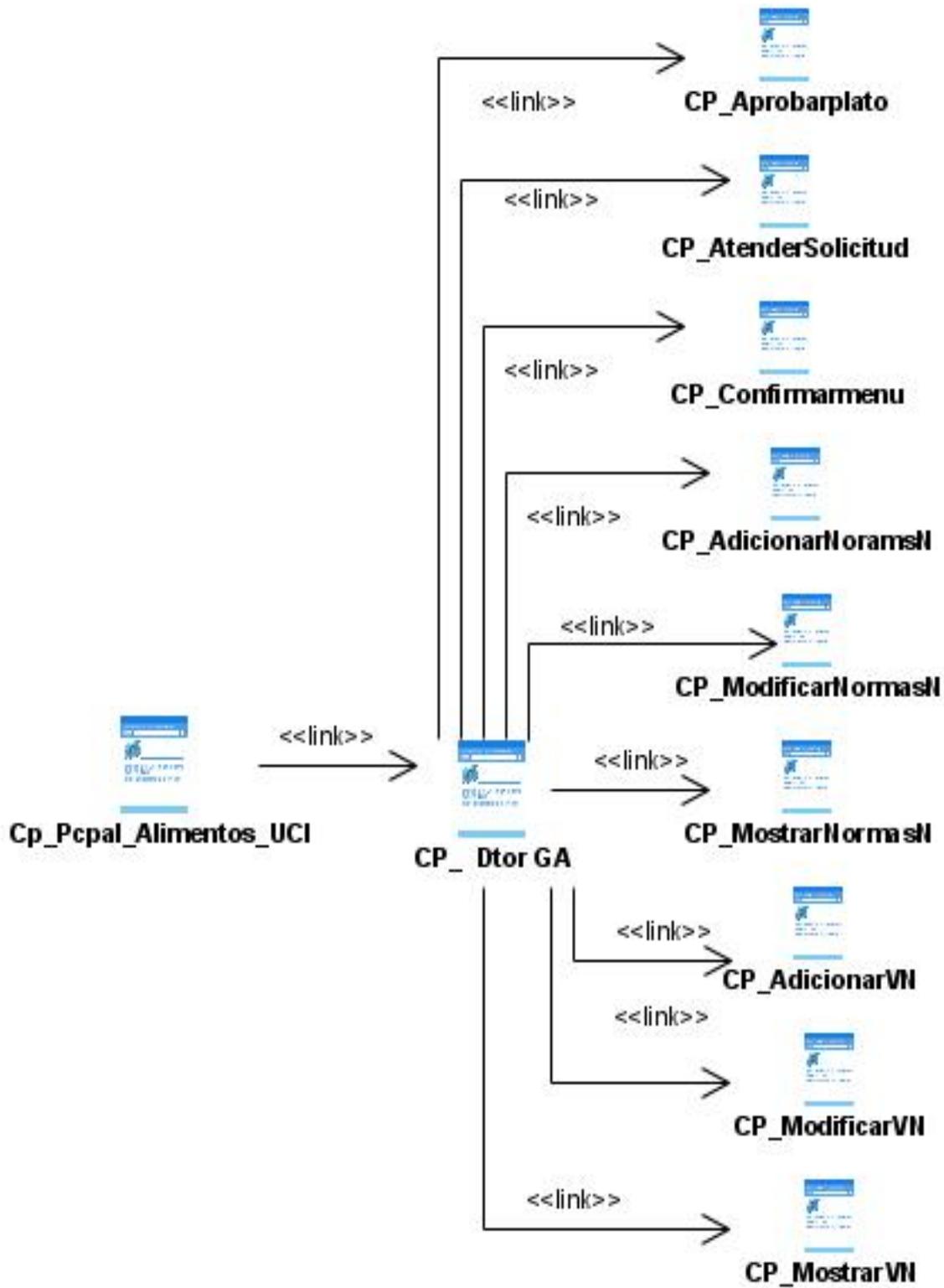


Fig. 64 Mapa de Navegación 3 (Para el Director General de Alimentos)

Glosario de términos

A

ActiveX: tecnología para el desarrollo de páginas dinámicas de Microsoft, cuenta con programación del lado del cliente y del lado del servidor, sus controles son particulares de Internet Explorer.

Ada: lenguaje de programación inspirado em Pascal y de propósito general, incluye orientas a objetos, permite encontrar muchos errores en tiempo de compilación.

Arborescencia: ilustración de la web planeada donde la unidad básica de información responde a cada página que el usuario encontrará en su recorrido.

ASP: Active Server Pages.

ASSETS NS: Sistema de Gestión Integral. Se comenzó a utilizar en el país en el 97, esta versión está desarrollada sobre Microsoft SQL.

ATM: Abastecimiento Técnico Material.

B

Balance Nutricional: dieta que debe presentar como característica que sea adecuada, equilibrada y variada.

C

Carga: se refiere a la transportación de diferentes alimentos que se llevan desde los distintos almacenes proveedores hasta los almacenes de la UCI.

CASE: Computer Aided Software Engineering.

Carbohidratos: tienen una función fundamental: la energética. Constituyen la energía de más fácil utilización. Su aporte tiene una acción ahorradora de proteínas y evitadora de la cetogénesis

CCCLU: Cocina Comedor del Centro Logístico UCI

CCIP: Complejo Comedores IP

C #: lenguaje de programación orientado a objetos, creado por Microsoft para la plataforma .Net, es una evolución de C/C++.

C++: lenguaje de programación orientado a objetos, derivado de C, creado por Bjarne Stroustrup.

Consumo Basal: consumo de energía que realiza nuestro organismo, ejemplo en el recambio de células, la formación de sustancias como hormonas.

CSS: lenguaje de hojas de estilo, la mejor forma de separar los contenidos y su presentación y es imprescindible para crear páginas web complejas.

CU: Casos de Uso

CVS: Sistema Concurrente de Versiones.

D

Decorator: decorador, patrón de diseño, agrega funcionalidad de adorno a un objeto.

Delphi: Borland Delphi, producido comercialmente por la empresa Borland Software Corporation, es un lenguaje de programación y un entorno de desarrollo rápido de software diseñado para la programación de propósito general con énfasis en la programación.

DGA: Dirección General de Alimentos

Digestibilidad proteica: Se define como la proporción de una proteína que normalmente es absorbida por el tracto gastrointestinal.

E

Energía: energía que poseen los alimentos, se mide en términos de caloría o kilocaloría.

F

Framework: simplifica el desarrollo de una aplicación mediante la automatización de algunos de los patrones utilizados para resolver las tareas comunes.

Frecuencia Acumulada: veces que se ha consumido un producto en un período.

Frecuencia Establecida: veces que se oferta un producto en un período.

G

GPL: General Public License.

Grasa: constituye el nutriente energético por excelencia, suministran los ácidos grasos esenciales y proporcionan al organismo las vitaminas liposolubles, aportan energía.

GRASP: General Responsibility Assignment Software Patterns.

GoF: Gang of four

H

HTML: Hyper Text Markup Language (Lenguaje para marcado de hipertexto).

I

Internet Explorer: navegador o visualizador de páginas web producido por Microsoft para Windows, usado para navegar por Internet y recorrer diferentes sitios de la WWW (World Wide Web).

ISPJAE: Instituto Superior Politécnico José Antonio Echeverría.

J

Java: lenguaje de programación orientado a objetos creado por James Gosling, que permite el diseño de aplicaciones para ser ejecutadas en las páginas Web.

JBuilder: es un entorno integrado de desarrollo que permite escribir, compilar, ejecutar y depurar programas en Java.

JSP: JavaServer Pages.

K

KDE: K Desktop Environment.

L

Linux: sistema operativo que se encuentra distribuido bajo Licencia Pública General de GNU o GPL, convirtiéndolo en software libre. Desarrollado por Linus Torvalds y con aportes de programadores de todo el mundo.

M

Mac OS X: sistema operativo basado en Unix.

Microsoft: compañía liderada por Bill Gates que se ha convertido en la diseñadora más grande de programas para microcomputadoras, estableciéndose como la pionera de la industria informática. La compañía ha desarrollado estándares como Windows y se transformó en una líder de la innovación tecnológica.

N

NetBeans IDE: herramienta enfocada para el desarrollo usando el lenguaje Java y un entorno de desarrollo integrado (IDE).

Normas de Consumo: cantidad de gramos que por productos se ofertan de manera directa e indirecta en la alimentación teniendo en cuenta las frecuencias a ofertar en el mes.

Nutrientes: toda sustancia contenida en los alimentos que cumplen diversas funciones en el organismo.

O

Oracle JDeveloper: te permite crear diagramas UML para definir el proyecto y pasar a implementarlo.

P

Pascal: lenguaje de programación estructurado y de alto nivel, creado en la década de los 70, por Niklaus Wirth, es un lenguaje excelente para aprender, soporta la recursividad y brinda la posibilidad de trabajar con punteros.

Pentium III: microprocesador de arquitectura i686 fabricado por Intel.

Perl: lenguaje de programación creado por Larry Wall, el cual no establece una filosofía concreta de programación aunque reúne varias, muy utilizado para la construcción de aplicaciones Web, es práctico para extraer la información de archivos de texto.

PHP: es un lenguaje de programación creado por Rasmus Lerdorf, diseñado para realizar páginas dinámicas.

PostgreSQL: servidor de base de datos relacional orientada a objetos de software libre, avanzado y de código abierto, presenta licencia BSD (Berkeley Software Distribution).

Proteínas: constituyen nuestra estructura, siendo imprescindibles para el crecimiento, para la síntesis de muchas sustancias relacionadas con nuestra inmunidad y las reacciones enzimáticas celulares.

Python: lenguaje de programación creado por Guido van Rossum, que permite la creación de un código elegante y limpio, ahorra tiempo en el desarrollo de programas, actualmente se desarrolla como un proyecto de código abierto.

R

RAM: Random Access Memory.

Recabar: recoger, guardar, recaudar.

RUP: Rational Unified Process

S

SQL: Structured Query Language (Lenguaje de consultas estructurado).

Symfony: completo framework diseñado para optimizar el desarrollo de las aplicaciones web.

T

Termogénesis: manera natural del cuerpo de desechar el exceso de calorías y que no sean depositadas como grasas almacenadas, sobre este concepto básico se basa El Control Metabólico como manera de controlar el peso corpóreo.

V

VPTS: Visual Photo Time Stamp.

VRL: Vicerrectoría de Logística

W

Windows: entorno gráfico creado y diseñado por Microsoft, utilizado por la mayoría de los usuarios a nivel mundial ya que cuenta con una interfaz que hace que el trabajo sea fácil.