

Universidad de las Ciencias Informáticas FACULTAD 4



Título: BOLSA LABORAL ONLINE

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

Autor(es): Reynold Elías Hernández

Cesar Rapetti Martínez

Tutor(es): Ing. Leevan Abón Cepeda

Ciudad de La Habana

20 de junio del 2008

DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

Reynold Elías Hernández

Cesar Rapetti Martínez

Ing. Leevan Abon Cepeda



"La misión del conocimiento no es iluminar a un alma que es oscura de por sí, ni hacer ver a un ciego. Su misión no es descubrir los ojos del hombre, sino guiarlo, gobernarlo y dirigir sus pasos a condición de que tenga piernas y pies para caminar."

Michel Eyquem de Montaigne

AGRADECIMIENTOS

Agradezco a mi familia por todo el apoyo que me han dado durante todos estos años de estudio.

Un agradecimiento especial para mi tutor Leevan Abón Cepeda por su gran ayuda, y por poder contar siempre con él cada vez que hacía falta.

Agradezco además a todos aquellos que de una u otra forma han hecho posible que yo este escribiendo estas líneas aquí.

Reynold Elías Hernández

Agradezco principalmente a mi familia que siempre han confiado en mí, para obtener este logro.

Para mi tutor y para mi compañero de tesis les doy las gracias también por su apoyo en el transcurso de la carrera y del trabajo de diploma.

A mis compañeros, a mis profesores y a la Revolución les estaré eternamente agradecido.

Cesar Rapetti Martínez

DEDICATORIA

Les dedico este trabajo a mi familia principalmente a dos personas que son muy especiales para mí y no puedo dejar de mencionar aquí.

A mi abuela Caridad, una mujer excepcional que le debo gran parte de lo que soy, gracias a sus esfuerzos he podido realizar mis estudios.

A mi padre que a pesar de no poder disfrutar este momento por no encontrarse con vida, fue algo que siempre ambicionó, sólo quiero decirle que jamás dejará de estar conmigo.

Reynold Elías Hernández

Principalmente le dedico este trabajo a mi familia que siempre me ha apoyado en todo, en especial a mi madre. A Yasnay que siempre me apoyó durante 5 años y compartió conmigo los momentos más extraordinarios de mi vida y que nunca pasarán por alto, la recordaré donde quiera que esté y que conste en este papel que todavía la amo. A los amigos que compartieron conmigo momentos buenos y malos. A Willy que como yo sería hoy un ingeniero más que hubiera dado esta revolución y la vida le hizo una mala trastada, donde quiera que estés Willy, tu hermano te dedica este logro que hubiera sido tuyo también.

A las personas que se quedaron en el camino y serán recordados por su valiosa amistad.

Me inspiré en las ideas de mis abuelos, de estudiar y ser alguien en la vida como ellos. Sus consejos siempre fueron los mejores y son un ejemplo de personas que hay que seguir.

Terminaré con un fragmento de un tema de Carlos Varela "...detrás de todos estos años, detrás del miedo y el dolor, vivimos añorando algo, algo que nunca más volvió, detrás de los que no se fueron, detrás de los que ya no están, hay una foto de familia donde lloramos al final y siempre tratando de mirar por el hueco de una aguja."

Cesar Rapetti Martínez

RESUMEN

Debido al creciente desarrollo alcanzado por Cuba urge la necesidad de cambiar métodos algo tradicionales a la hora de contratar un obrero para un determinado puesto laboral.

La automatización de estos procesos trae consigo un cambio radical a la hora de gestionar un puesto de trabajo, permite eliminar en gran medida las irregularidades, demoras y mala selección en el momento de ocupar una vacante.

Con el presente trabajo se procura desarrollar un software capaz de gestionar todo lo referente a una bolsa laboral. Para ello se realiza una investigación de las actuales bolsas laborales existentes en el mundo, así como su aceptación y cambios que ésta trae consigo para una mayor calidad y agilización del proceso de contrata de personas. Posteriormente se efectúa un estudio detallado de cómo se ejecutan los procesos de captación de personal en Cuba, permitiendo éste, llegar a un Modelo de Negocio y posteriormente modelar un diseño del software e implementar una aplicación funcional que automatice gran parte de estos procesos.

Con la aplicación realizada se pretende mejorar y agilizar el proceso de captación de personal para una plaza laborar disponible.

PALABRAS CLAVE: Bolsa laboral, empleo, aplicación, gestión.

AGRADECIMIENTOS.....	I
DEDICATORIA	II
RESUMEN.....	III
INTRODUCCIÓN.....	1
CAPITULO 1: FUNDAMENTACIÓN TEÓRICA.....	3
INTRODUCCIÓN.....	3
1.1- SURGIMIENTO DE LAS BOLSAS LABORALES.....	3
1.2- ¿QUÉ ES UNA BOLSA LABORAL ONLINE?	4
1.3- BOLSA LABORAL ONLINE EN EL MUNDO	4
1.4- FACILIDADES QUE BRINDA UNA BOLSA LABORAL ONLINE.....	5
1.5- DESVENTAJAS DE UNA BOLSA LABORAL ONLINE.....	6
1.6- LAS BOLSAS DE TRABAJO EN INTERNET	7
1.6.1- La virtualidad agiliza los procesos	7
1.6.2- Un contacto directo entre empresas y profesionales	7
1.6.3- Una comunidad de profesionales	8
1.6.4- Aumenta la competencia.....	8
1.6.5- Un observatorio permanente del mercado de trabajo	8
1.7- ESTRUCTURA	9
1.7.1- Módulo de Aspirantes.....	9
1.7.2- Módulo de Empleador	10
1.8- ARQUITECTURA UTILIZADA PARA LA IMPLEMENTACIÓN (ARQUITECTURA EN 3 CAPAS Y MODELO-VISTA-CONTROLADOR (MVC)).....	10
1.8.1- Arquitectura en 3 capas	11
1.8.1.1- Ventajas de las 3 capas	12
1.8.2- Modelo MVC	13
1.8.2.1- Ventajas de MVC	14
1.8.2.2- Desventajas de MVC.....	15
1.9- TECNOLOGÍAS USADAS.....	15
1.9.1- Personal Home Page (PHP).....	15
1.9.1.1- Ventajas de PHP	15

1.9.1.2 Desventajas.....	15
1.9.1.3- ¿Por qué utilizar PHP y no otras opciones?	16
1.10- POSTGRESQL	16
1.10.1- ¿Qué es PostgreSQL?	16
1.10.2- Ventajas de PostgreSQL.....	17
1.10.3- Características	17
1.11- RRHH-XML.....	18
1.11.1- RRHH	19
1.11.2- XML	19
1.11.3- HTML	19
1.12- RATIONAL UNIFIED PROCESS (RUP).....	20
1.13- LENGUAJE UNIFICADO DE MODELADO (UML).....	21
1.14- HERRAMIENTAS UTILIZADAS	21
1.14.1- Dreamweaver 8.....	21
1.14.2- Visual Paradigm.....	22
1.14.3- PG_ADMIN	22
1.14.4- Apache.....	23
CONCLUSIONES.....	24
CAPITULO II: CARACTERÍSTICAS DEL SISTEMA.	25
INTRODUCCIÓN.....	25
2.1- ANÁLISIS DE LOS PROCESOS ACTUALES.....	25
2.2- OBJETO DE AUTOMATIZACIÓN	25
2.3- DESCRIPCIÓN GENERAL DE LA PROPUESTA DE SISTEMA.....	25
2.4- MODELO DEL NEGOCIO.....	26
2.5- DIAGRAMA DE CASOS DE USO DEL NEGOCIO	27
2.6- DESCRIPCIÓN DEL CASO DE USO DEL NEGOCIO.....	27
2.7- DIAGRAMAS DE ACTIVIDADES DEL NEGOCIO	28
2.8- MODELO DE OBJETO DEL NEGOCIO.....	29
2.9- REQUERIMIENTOS FUNCIONALES	30
2.10- REQUERIMIENTOS NO FUNCIONALES	30
2.11- CASOS DE USO Y ACTORES DEL SISTEMA.....	32
2.11.1- Definición de los actores	32

2.11.2- Listado de casos de uso.....	32
2.12- DIAGRAMA DE CASOS DE USO DEL SISTEMA	36
2.13- DESCRIPCIÓN DE LOS CASOS DE USO EXPANDIDA	37
CONCLUSIONES	54
CAPÍTULO III: ANÁLISIS Y DISEÑO DEL SISTEMA.....	55
INTRODUCCIÓN.....	55
3.1- ANÁLISIS	55
3.1.1- ¿Qué es el Análisis?	55
3.1.2- Diagramas de clases del Análisis	56
3.2- DISEÑO	59
3.2.1- ¿Qué es el diseño?	59
3.2.2- Diagramas de clases de diseño	60
3.3- DIAGRAMAS DE ITERACIÓN.....	65
3.3.1- Diagrama de secuencia.....	65
3.4- MODELO DE DATOS	77
3.4.1- Diagrama de clases de entidad	77
3.4.2- Modelo físico de datos	78
3.5- DESCRIPCIÓN DE LAS TABLAS MÁS IMPORTANTES DE LA BASE DE DATOS.	78
3.6- DESCRIPCIÓN DE CLASES:	81
CONCLUSIONES	84
CAPITULO IV: IMPLEMENTACIÓN Y PRUEBA.....	85
INTRODUCCIÓN:.....	85
4.1- DIAGRAMA DE DESPLIEGUE.....	85
4.2- DIAGRAMA DE COMPONENTES:.....	86
4.2.1- ¿Qué es Componente?	86
4.3- PRUEBA:	93
4.3.1- Modelo de pruebas	94
CONCLUSIONES	101
CONCLUSIONES	102
RECOMENDACIONES	103

BIBLIOGRAFÍA.....	104
GLOSARIO.....	106
ANEXOS.....	108
ANEXO 1.....	108
ANEXO 2.....	130
ANEXO 3.....	132
ANEXO 4.....	136
ANEXO 5.....	142
ANEXO 6.....	144
ANEXO 7.....	149
ANEXO 8.....	153

INTRODUCCIÓN

En estos momentos no existe en Cuba una aplicación capaz de gestionar una plaza laboral, no hay información suficiente acerca de los empleos ofertados para la población, ni las empresas o entidades tienen un amplio conocimiento de todas las personas desempleadas en un área determinada. Existe una necesidad imperiosa de estandarizar todo lo que es la información relacionada con recursos humanos, todo lo antes expuesto propicia una situación problemática en cualquier región del país, por todo lo antes expuesto se crea un problema a resolver: contribuir a una eficaz gestión de un puesto laboral.

La necesidad de éste trabajo viene debido a que la mayoría de las personas se encuentran desorientadas cuando se enfrentan a una búsqueda de empleo. En la actualidad la gestión de un puesto laboral se realiza de forma manual, y es algo incómoda para el que desea encontrar un trabajo, producto que en muchas ocasiones tiene que viajar grandes distancias y cuando llega ocurre que otro ya se le adelantó, aunque éste tenga menos aptitudes para la plaza, sólo por llegar primero la obtuvo, además que con esa forma de gestionar empleo se cometen muchas irregularidades, donde el principal problema viene dado en la selección del personal, debido a que no existe en Cuba lo que se llama competencia, donde él mejor sea el que se quede con el puesto de trabajo, es decir, él de mayor currículum, ya que con que uno lo solicite basta, se le otorga, pues muchas veces no hay conocimiento de las plazas que se ofertan en un centro de trabajo y sólo se presenta aquel cual conoce personal que ejerce profesión ahí.

Producto al mismo, el campo de acción de éste trabajo es la gestión empresarial de una bolsa laboral basada en tecnología online para las empresas cubanas que soliciten mano de obra laboral.

La bolsa laboral online no tiene antecedentes en nuestro país, aunque sí se aplica en gran medida en otros países, los pertenecientes a la Unión Europea y en los Estados Unidos, aplicándolo a través de la red de redes, Internet, siendo ya el sistema más utilizado por los gestores de personal en éstos.

El presente trabajo tiene en su objeto de estudio la gestión empresarial de una bolsa basada en la tecnología online y desarrollada en PHP con un marco estandarizado de recursos humanos, basado en RRHH-XML.

Como objetivo general se tiene la investigación e implementación de un sistema que sea capaz de desarrollar la gestión de todo lo referente a una bolsa laboral, como objetivos específicos encontramos:

1. Diseño de un marco teórico sobre la gestión de una bolsa laboral basado en RRHH- XML.
2. Diseño de la aplicación Web en PHP.
3. Implementación de un prototipo funcional que sirva como base para la gestión de una bolsa laboral basado en RRHH- XML.

Para guiar la investigación se plantea como hipótesis: es posible crear una aplicación para la gestión laboral online, mediante la cual se mejorará la calidad de selección del personal para un puesto de trabajo determinado.

Las tareas a desarrollar son:

- Selección de las herramientas para llevar a cabo el proyecto y la elección de la plataforma en la que se desarrollará la aplicación. Fundamentando su elección.
- Selección de la metodología de Análisis y Diseño de sistemas informáticos, que facilite y garantice la creación con calidad del sistema.
- Implementación de la aplicación hasta obtener una primera versión la cual puede ser mejorada y ampliada con nuevos módulos.
- Investigación y aplicación del estándar RRHH-XML.

En la práctica se espera que este trabajo sea muy útil, debe de brindar la oportunidad a aquellas personas que tienen una computadora con acceso a red nacional tener la posibilidad de optar por un puesto de trabajo de forma no presencial, ya sea en cualquier empresa. Además de poder ver, que puestos de trabajo está ofertando una empresa, todo esto de forma Online. Crear y definir un estándar RRHH-XML para la generación de todo lo referente a la gestión de recursos humanos generados por una empresa.

Definiéndose en este Trabajo de Diploma cuatro capítulos:

- Capítulo I. Fundamentación teórica: se expone el estado del arte del problema antes mencionado a nivel internacional, nacional y de la UCI, tendencias, técnicas, tecnologías, metodologías y software que se usan en la actualidad.

- Capítulo II. Características del sistema: se explicará el objeto de informatización, información y documentación que se maneja, se hace una propuesta del sistema, se modela el negocio, se especifican requisitos funcionales y no funcionales que debe cumplir el sistema, y se definen los casos de uso.

- Capítulo III. Análisis y diseño del sistema: se crea el modelo de análisis, el modelo de clases de análisis, el modelo de clases del Diseño y los diagramas de interacción por cada realización de casos de uso, la descripción de las clases y el diseño de la Base de Datos, con la descripción de las tablas de la Base de Datos

- Capítulo IV. Implementación: se realizan los diagramas de Despliegue y Componentes además de las pruebas al sistema.

CAPITULO 1: FUNDAMENTACIÓN TEÓRICA

Introducción

A partir de la revolución industrial, con la llegada del sistema de trabajo fabril, se extendió la visión común sobre la función del trabajador. Se entendía que el personal representaba un costo. En la actualidad el capital humano es considerado como un recurso más de la empresa, e incluso los directivos entienden que la formación del trabajador, su experiencia o grado de integración contribuyen a mejorar el funcionamiento de la propia organización.

La revolución industrial trajo consigo un constante desarrollo de la tecnología, producto a ello, es imperante deshacerse de métodos algo tradicionales por otros más modernos, en lo que el principal factor sería el uso de las tecnologías, excluyente de esto no va a ser la gestión de un puesto de trabajo.

Para llevar a cabo el reclutamiento de personal se han utilizado durante décadas distintos procedimientos, desde el rumor hasta el típico anuncio en prensa han servido para cubrir vacantes. Sin embargo, y desde hace pocos años, Internet se ha convertido en uno de los medios más utilizados en la capacitación y selección de trabajadores, pues su uso presenta múltiples ventajas, siendo posible gracias a la implementación de bolsas laborales online.

En el presente capítulo se brinda una detallada información de todo lo referente a este tema.

1.1- Surgimiento de las Bolsas Laborales

Las bolsas laborales fueron en su inicio ideadas como organismos sindicales para suprimir la opresión generada por los capitalistas en Francia. Las Bolsas del Trabajo facilitaban así mismo una amplia gama de servicios que por supuesto en mucho favorecían a sus agremiados. Las Bolsas del Trabajo representaron, en su momento, una alternativa libertaria real que puso, como comúnmente se dice, los pelos de punta de no pocos capitalistas franceses que en ellas veían el germen de su aniquilamiento.

Además de correr a su cargo el servicio fundamental de colocación de los obreros, todas estas Bolsas del Trabajo habían constituido bibliotecas, organizado cursos profesionales, económicos, científicos y técnicos, servicios de asistencia a los compañeros en viaje. Políticamente las Bolsas del Trabajo existían ya hacía un siglo, es decir, el 2 de marzo de 1790, cuando un informe de De Corcelles promueve el proyecto. Sometido a examen por el departamento de los Trabajos públicos, el proyecto desaparece.

Por espacio de cincuenta años el nombre de *Bolsa del Trabajo* desapareció del vocabulario. En 1845, De Molinari, redactor jefe del *Journal des Economistes*, reencontró o tal vez la concibió la idea de una Bolsa del Trabajo obrera. El 24 de febrero de 1875. En esta fecha se la creación en la Rue de Flandre de una Bolsa del Trabajo. Todavía se necesitarían once años para que apareciese sobre las mesas de

la oficina del Consejo municipal (5 nov. 1886) el siguiente informe debido. Esta vez la causa de la Bolsa del Trabajo por fin vencía y el 3 de febrero de 1887 el Consejo municipal ponía solemnemente en manos de los sindicatos parisinos el inmueble de la calle Jean-Jacques Rousseau. Este es el origen aparente de las Bolsas del Trabajo, aunque hay que consignar que la iniciativa del Consejo municipal de París no fue imitada en ninguna otra parte.

Las Bolsas del Trabajo tal como existían en ese entonces, salvo el nombre, son muy anteriores a la inauguración del inmueble de la calle Jean-Jacques Rousseau. Los equivalentes de las mismas se pueden hallar en dos o tres federaciones obreras creadas por la Internacional. En aquella sociedad la Bolsa del Trabajo debía ser en primer lugar una asociación de resistencia. Asociaciones de resistencia contra las reducciones de trabajo, contra la prolongación excesiva de la duración del trabajo y también contra el aumento, o mejor dicho (dado que el mecanismo hace inevitable estos aumentos) contra un aumento exagerado del precio de los objetos de consumo.

La función inmediata de las Bolsas del Trabajo consiste en mantener lo más posible el equilibrio entre los precios pagados al trabajo y el precio de adquisición de los productos. Por fin, el 9 de agosto de 1900 las Bolsas del Trabajo recibían el primero de los cuadros generales de los empleos vacantes, los cuales aparecieron ya regularmente después de aquella fecha. [14]

1.2- ¿Qué es una bolsa laboral online?

No es más que una aplicación que permite a las personas optar por un puesto de trabajo determinado de manera online y no presencial. En los países capitalistas, este mercado existe bajo el aspecto de instituciones intermediarias entre los obreros desocupados y los dueños del capital, para la compra-venta de fuerza de trabajo.

1.3- Bolsa laboral online en el mundo

Los portales de empleo comienzan a plantear dificultades a las empresas de bolsa de trabajo, tanto así que éstas han optado también por abrir sucursales online. Las bolsas de trabajo en Internet han crecido notablemente gracias a la divulgación de los beneficios derivados del uso de las nuevas tecnologías.

Ya son bastantes las Empresas que están creando en sus Web un apartado con el fin de hacer selección de personal, pero, al fin, los portales de empleo son los más utilizados por los demandantes y por las empresas que lanzan las ofertas en la Red, haciendo que los sistemas tradicionales de selección

y reclutamiento estén disminuyendo, mientras que las Web de empleo y las Web corporativas están aumentando de forma relevante su uso.

Desde la puesta en marcha de los primeros canales de búsqueda de empleo online, el número de personas que se decanta por este sistema está aumentando de forma espectacular. Según datos de la consultora Media Metrix Europe, más de dos millones de europeos utilizan actualmente Internet para buscar trabajo. Concretamente, en Trabajos.com hay más de 100.000 currículos y alrededor de 7.000 ofertas de empleo.

¿Cuáles pueden ser las razones de este progresivo aumento de búsqueda de trabajo online? Sin duda, los usuarios de este servicio son conscientes de que la Red permite optimizar los procesos de selección de personal, por parte de las empresas, así como aumentar la eficacia de la búsqueda o cambio de empleo, por parte de los trabajadores. Es, por tanto, un método que convence a las dos partes del proceso. [1]

1.4- Facilidades que brinda una bolsa laboral online.

El reclutamiento online proporciona a las empresas ventajas tales que hacen que actualmente más de una cuarta parte de los responsables del reclutamiento, prefieran utilizar ya Internet:

- La bolsa de trabajo online es el medio más rápido y económico de ofertar y/o demandar empleo.
- Recaba información para la planificación del desarrollo profesional de cada empleado.
- Determina elementos objetivos, puntuales y medibles, para la calificación del desempleo de cada trabajador (evita la subjetividad en el proceso de la evaluación).
- Una de esas ventajas es la publicación inmediata de la oferta de selección de personal.
- Para los candidatos enviar su currículum mediante correo electrónico, rellenar formularios estandarizados online, poder hacer su propia gestión y actualización de expediente, y hacer el seguimiento a sus ofertas de interés, suponen facilidades importantes.
- Aquellos que buscan trabajo o quieren cambiar de empleo han encontrado en las empresas de selección online un soporte amplio, inmediato e interactivo, que les permite realizar una búsqueda más eficaz, con un importante ahorro de tiempo.
- La comodidad, flexibilidad, y rapidez del medio a la hora de crear y enviar el expediente son otras de las ventajas fundamentales de esta forma de selección.
- También el automatismo en la preselección. Hay que tener en cuenta que la mayor parte de los procesos de RRHH. Podrán muy pronto ser administrados electrónicamente, consiguiendo todos con ello una mayor efectividad y rentabilidad.

- Envía por e-mail a los candidatos las ofertas de trabajo que mejor se ajustan a su perfil profesional y sus expectativas en función de parámetros previamente definidos por éste (categoría de trabajo, localidad, tipo de puesto, banda de salario, tipo de relación laboral).

- La información disponible tanto para candidatos como para empresas sobre cuestiones laborales, desde legislación, a la redacción de un currículum, pasando por como asumir un nuevo cargo en una estructura empresarial.

- La empresa dispone de mucho más espacio para anunciarse, puede modificar su anuncio una vez publicado, y está visible 24 horas al día, siete días a la semana. Gestionar una búsqueda por otros soportes puede prolongarse 25 días como media, (creatividad y edición del anuncio, inserción, respuesta de los candidatos), mientras que a través de Internet, el tiempo medio para dar con un candidato se reduce a cinco días.

- Realizar una búsqueda personalizada en función de las necesidades de cada uno; especificar el sector laboral, la ciudad donde se desea trabajar; hasta comparar varias ofertas similares y decidirse por la que convenga; incluso recibir información sobre pruebas de selección, autoempleo, franquicias, teletrabajo, tipos de contrato, legislación laboral, etc. [1]

1.5- Desventajas de una bolsa laboral online

En muchos países el uso de Internet todavía no está muy difundido. La cantidad de personas conectadas a Internet no es la totalidad de la población.

Los engaños que diariamente muchos usuarios sufren en la red, falsedad en los datos por miedo al mal uso de los mismos e inserción de información inexistente

Muchas bolsas de trabajo online ofrecen la posibilidad de realizar pruebas a los demandantes de empleo, pero este tipo de demandantes "proporciona un boceto muy general de los demandantes". Para obtener resultados satisfactorios es necesario realizar pruebas más sofisticadas que permitan evaluar conocimientos, pero también habilidades, capacidades directivas y potencial de mejora.

Hoy en día muchos profesionales y empresas ofertan y demandan trabajo utilizando métodos tradicionales.

Queda aún mucho trabajo por hacer para conseguir redefinir los procesos de la empresa en el departamento de recursos humanos, así como los sistemas de información para poder aprovechar todas las oportunidades que Internet nos ofrece. [1]

1.6- Las bolsas de trabajo en Internet

El impacto de Internet en los procesos de selección de recursos humanos es cada vez mayor: gracias a las bolsas de trabajo virtual desaparecen buena parte de las fases intermedias de dichos procesos, lo cual los agiliza enormemente. Asimismo, puede hacerse una gestión desde una perspectiva de mercado de trabajo global; se favorece el contacto directo entre profesionales y empresas; se crean comunidades virtuales de profesionales; se incrementa la competencia entre empresas y candidatos, lo que lleva a la adquisición de nuevos aprendizajes para obtener los mejores resultados; y se configuran observatorios permanentes del mercado de trabajo, que ofrecen grandes posibilidades de gestión y análisis.

Cada día descubrimos nuevas bolsas de trabajo surgidas en la red, y es lógico. Internet es, sobre todo, un entorno de comunicación y, por lo tanto, un espacio apropiado para que empresas y profesionales se encuentren y se reconozcan a la hora de resolver sus necesidades ocupacionales. Este fenómeno cambia considerablemente la gestión de los procesos de selección de las empresas, sobre todo en lo que concierne a las fases de reclutamiento de candidatos.

1.6.1- La virtualidad agiliza los procesos

Con la virtualidad desaparecen buena parte de las fases intermedias de los procesos de selección, causantes muchas veces de la paralización de los procesos y la demora del resultado final. Actualmente, el proceso de publicación, comunicación, reclutamiento y primera preselección se hace todo al mismo tiempo, mediante un mismo sistema y en muchos casos bajo la supervisión de la misma persona que gestiona el proceso global. Y, lo que es más importante, el sistema informático hace de una manera automática buena parte de este proceso sin necesidad de ningún intermediario entre los interesados. [1]

La calidad de las bolsas de trabajo virtuales reside en la rapidez a la hora de gestionar la información solicitada y en la garantía de veracidad de los datos introducidos por profesionales y empresas. Es decir, se valora la potencia del sistema de comunicación y la calidad de la información. [1]

1.6.2- Un contacto directo entre empresas y profesionales

El poder de decisión recae directamente sobre los protagonistas del proceso: profesionales y empresas, candidatos y oferentes son quienes tendrán que decidir ante quién quieren presentarse, qué información quieren transferir de ellos mismos y qué grado de confidencialidad quieren establecer con sus interlocutores. Igualmente podrán decidir qué rol quieren desempeñar en este sistema de comunicación.

Pero el motivo de la comunicación no es siempre la inserción laboral. La bolsa de trabajo es utilizada cada vez más para establecer contacto con profesionales autónomos (<http://www.forvm.org/>) o tele trabajadores (<http://www.centraldeteletrabajo.com/>) que ofrecen su talento y sus servicios para colaborar con las empresas generadoras de actividad.[1]

1.6.3- Una comunidad de profesionales

Las bolsas de trabajo también se van configurando como entornos donde se agrupan los profesionales y las empresas dinamizadoras de la nueva economía, comunidades virtuales que van más allá de responder a necesidades puntuales de inserción; quieren constituirse en entornos referentes de ocupación que acompañen a los profesionales a lo largo de su vida laboral y donde puedan encontrar información, recursos y servicios que les permitan estar al día y adaptarse a la evolución permanente del mercado (<http://www.parqueempresarial.com/>).[1]

1.6.4- Aumenta la competencia

En un entorno tan transparente como es Internet, y visto el momento actual de rápidos movimientos económicos, la competencia entre profesionales, y también entre empresas, es palpable. Hay quienes la utilizan como un arma, pero hay quienes buscan sistemas para esquivarla.

Esta competencia requiere nuevos aprendizajes; las empresas tendrán que aprender a hacerse atractivas para los profesionales, a mostrar sus políticas de recursos humanos, a proponer nuevas maneras de fidelizar sus equipos. Vemos cómo las empresas compiten para atraer a los profesionales de más talento, buscan los medios para salir destacadas en las cabeceras de las bolsas virtuales, insertan su logotipo para prestigiarse ante la comunidad de profesionales y despliegan procesos activos de búsqueda de profesionales en la red.

Al mismo tiempo, los profesionales también tendrán que aprender a desarrollar su profesionalidad de manera permanente para no perder competitividad en el mercado de trabajo, sea para conservar su puesto de trabajo actual o, sobre todo, para promocionarse en la propia organización o cambiando de empresa.[1]

1.6.5- Un observatorio permanente del mercado de trabajo

La rápida gestión de los datos ofrece una información clave para entender cómo evoluciona el mercado de trabajo y valorar quién está bien situado para mantenerse en una buena posición.

Los profesionales pueden utilizar las bolsas virtuales de trabajo como termómetro para medir la calidad de su talento, pueden comparar su perfil con el del resto de los profesionales de la red y, sobre todo, con el perfil que buscan las empresas. También pueden valorar la capacidad de atraer el interés de las empresas oferentes de nuevos puestos de trabajo y, por lo tanto, su capacidad de insertarse.

Al mismo tiempo, una bolsa virtual de trabajo es un observatorio permanente y actualizado para la empresa que compite en la nueva economía. Conocer de primera mano quién genera nuevos puestos de trabajo y de qué perfiles y cuál es el volumen de ofertas que las empresas del propio sector tienen en activo puede representar una información clave a la hora de tomar decisiones estratégicas. [1]

Las empresas, en definitiva, encuentran en el medio online las características necesarias para realizar una selección eficaz: ahorro de tiempo y costes, flexibilidad, amplias bases de datos y globalización

El uso de la Red se está generalizando progresivamente y una de las aplicaciones que más éxito está teniendo es el servicio de búsqueda de empleo online. A pesar de las posibles barreras de acceso a Internet por una parte de los usuarios y de algunas empresas de sectores determinados, la utilidad de este sistema se confirma día a día.

1.7- Estructura

Las bolsas de trabajo virtuales generalmente se encuentran contenidas en páginas Web dedicadas para este fin y están estructuradas en forma de módulos.

Estas páginas están implementadas en forma interactiva por la alta interacción que necesitan los usuarios con ellas, nos permite ejecutar instrucciones como respuesta a sus acciones.

Las páginas Web o sitios de bolsas de trabajo están conformadas por módulos, el módulo de aspirantes y el módulo de empleadores.[1]

1.7.1- Módulo de Aspirantes

En el módulo de aspirantes, los interesados deben llenar un formulario de registro en el que ingresarán sus datos personales y correo electrónico, además de ingresar su nombre de usuario y contraseña.

Luego de esto, los aspirantes llenarán otro formulario en el que ingresarán su currículum.

Ya inscrita, la persona podrá tener acceso a las ofertas de trabajo publicadas recientemente y aspirar a las mismas. Según vayan publicando nuevas ofertas de empleo, así mismo el aspirante podrá recibirlas en su correo electrónico o viéndolas en el sitio.

Todo el currículum que vayan ingresando se almacena directamente en una base de datos.[1]

1.7.2- Módulo de Empleador

Este módulo es el que le da el acceso a las empresas para ver los usuarios que hay registrados y el currículum de cada cual. Hay distintos tipos de accesos, como por ejemplo:

1. Las empresas accedan directamente a la base de datos y pueden ver todos los currículum de las personas.
2. Las empresas publican una oferta empleo y pueden ver el currículum de cada persona que cumple con las condiciones que ella necesita.

Al entrar a la página los candidatos pueden revisar todas las ofertas de empleo. Si encuentra alguna que cumpla con sus expectativas usted solicita directamente desde el sitio.[1]

1.8- Arquitectura utilizada para la implementación (arquitectura en 3 capas y modelo-vista-controlador (MVC))

Los tiempos han cambiado, la tecnología ha cambiado de una forma que nunca se hubiese imaginado, los avances que se han logrado en últimos años es mucho mayor que los que se ha conseguido en todo el resto de la historia. La tecnología de la información no está exenta de estos cambios. Los sistemas que en un primer momento fueron desarrollados dentro de las posibilidades de la técnica que existía en ese entonces, se presentan ineficaces para la tecnología y las necesidades de los negocio actuales. El mayor problema reside en que los sistemas, como son utilizados hoy, fueron desarrollados para una realidad del ayer.

La historia de los sistemas informáticos comienzan con una arquitectura mono capa o centralizada que aunque podía ser vista hasta hace algunos años, ya prácticamente se ha dejado de utilizar. Consistía principalmente de un “gran” ordenador central (*mainframe*) y una serie de terminales que no ejecutaban ningún proceso (*Terminales bobas*). Tanto el acceso a los datos, la lógica de la aplicación y la presentación de la información estaba completamente implementada en un sólo bloque monolítico de software. Cualquier modificación sobre la aplicación debía ser hecha en este único módulo. Un avance sobre este modelo fue realizado a partir de bases de datos basadas en servidores de archivos. En este caso, la base de datos consiste en uno o más archivos reconocibles por el sistema operativo. En esta arquitectura, el programa que permite el acceso y administración de la base de datos debe estar muy estrechamente unido a la aplicación cliente. Un avance más a la arquitectura anterior consiste en dividir los sistemas de una sola capa en dos capas bien diferenciadas. Estas aplicaciones están compuestas por una capa de interfaz con el usuario, que es la capa en donde el usuario interactúa con su PC y que

además generalmente concentra toda la lógica del negocio, y una capa de acceso a datos, cuya función generalmente la realiza un servidor de base de datos y típicamente reside en un servidor central bajo un entorno controlado. Uno de los problemas en este tipo de arquitecturas es la dificultad de manipular los cambios en la capa que interactúa con el cliente. Otro problema es la dificultad de compartir procesos comunes. Luego de largas horas de trabajo frente a la máquina para lograr un proceso en particular, este código es difícilmente reutilizable en otras aplicaciones.[15]

1.8.1- Arquitectura en 3 capas

Como bien dice el nombre esta compuesta por tres capas, la de interfaz (presentación), la de acceso a datos y una capa de reglas o lógica de negocio que es quien realmente representa a la empresa y debe obviar tanto la estructura de los datos como su ubicación. Por ejemplo, en un aplicación Web generalmente el cliente esta representado por un navegador que muestra las páginas enviadas por el servidor que administra la lógica del negocio y que permite también el ingreso de datos.

1. **Acceso a datos:** sus funciones incluyen el almacenamiento, la actualización y la consulta de todos los datos contenidos en el sistema. En la práctica, esta capa es esencialmente un servidor de bases de datos aunque podría ser cualquier otra fuente de información. Gracias a esta división, es posible agregar soporte para una nueva base de datos en un período de tiempo relativamente corto. La capa de datos puede estar en el mismo servidor que las de lógica de negocio y presentación, en un servidor independiente, o incluso estar distribuida entre un conjunto de servidores.

2. **Lógica de negocio:** el comportamiento de la aplicación es definido por los componentes que modelan la lógica de negocio. Estos componentes reciben las acciones a realizar a través de la capa de presentación, y llevan a cabo las tareas necesarias utilizando la capa de datos para manipular la información del sistema. Tener la lógica de negocio separada del resto del sistema también permite una integración más sencilla y eficaz con sistemas externos, ya que la misma lógica utilizada por la capa de presentación puede ser accedida desde procesos automáticos que intercambian información con los mismos.

3. **Presentación:** la capa de presentación representa la parte del sistema con la que interactúa el usuario. En una aplicación Web, un navegador puede utilizarse como cliente del sistema, pero esta no es la única posibilidad, también puede generarse una aplicación que cumpla las funciones de un cliente “ligero” para interactuar con el usuario. [15]

1.8.1.1- Ventajas de las 3 capas

- **Permite la reutilización:** la aplicación está formada por una serie de componentes que se comunican entre sí a través de interfaces y que cooperan para lograr el comportamiento deseado. Esto permite no solamente que estos componentes puedan ser fácilmente reemplazados por otros, por ejemplo porque se necesita mayor funcionalidad sino también que los mismos puedan ser utilizados para otras aplicaciones.

- **Acompaña el crecimiento:** cada uno de los componentes de la aplicación pueden colocarse en el mismo equipo o distribuirse a través de una red. De esta manera, proyectos de gran envergadura pueden dividirse en pequeños proyectos más simples y manejables, que se pueden implementar en forma progresiva, agregando nuevos servicios según la medida de crecimiento de la organización.

- **Uso eficiente del hardware:** debido a que los componentes pueden ser distribuidos a través de toda la red, se puede hacer un uso más eficiente de los recursos de hardware. En vez de necesitarse grandes servidores que contengan la lógica de negocios y los datos, es posible distribuirlos en varias máquinas más pequeñas, económicas y fáciles de ser reemplazadas.

- **Distintas presentaciones:** debido a que separa la presentación de la lógica de negocios, es mucho más sencillo realizar tantas presentaciones diferentes como dispositivos con capacidades e interfaces se tenga.

- **Encapsula los datos:** debido a que las aplicaciones cliente se comunican con los datos a través de peticiones que los servidores responden ocultando y encapsulando los detalles de la lógica de la aplicación, obtenemos un nivel de abstracción que permite un acceso a los datos consistente, seguro y auditable. Con esto se pretende que si hay cambios en la capa de datos, la capa de negocios se haga cargo de administrar tales cambios y el cliente, en la mayor parte de los casos ni se entere.

- **Ahorra tiempo y costos:** en el desarrollo de nuevas aplicaciones y la integración en el resto de los procesos de gestión de la empresa.

- **Mejor calidad en las aplicaciones:** como las aplicaciones son construidas en unidades separadas, estas pueden ser probadas independientemente y con mucho más detalle, esto conduce a obtener un producto mucho más sólido.

Una de sus mayores ventajas es que los sistemas se independizan en cierta forma de la capacidad tecnológica y el tamaño del negocio, por lo que pueden acompañar de manera eficiente el crecimiento de las empresas que los utilizan. Dadas las características del modelo, se puede implementar y dejar operativa una solución de negocios en tiempos extremadamente cortos, permitiendo conseguir una

ventaja competitiva particular respecto a otros negocios. También permite la modificación del sistema en períodos de tiempo reducidos, incluso cuando es necesario agregar características especiales a las aplicaciones.[15]

1.8.2- Modelo MVC

Un patrón de diseño es una solución de calidad para un problema recurrente de diseño. Tiene como propósito proporcionar una base para poder realizar una actividad, mejorando la calidad del producto que esa actividad de cómo resultado.

¿Cómo se puede escoger un patrón adecuado? Esta es una pregunta un poco difícil ya que la mayoría de las actividades de desarrollo o producción no se ajustan perfectamente a un patrón definido, por eso es importante llevar a cabo un análisis para poder visualizar bien cual es el patrón que mas se ajuste a las necesidades de desarrollo.

En resumen un patrón de diseño puede verse como una plantilla que puede ser aplicada en muchas situaciones diferentes para dar una buena solución.

Para el diseño de aplicaciones con sofisticadas interfaces se utiliza el patrón de diseño Modelo-Vista-Controlador. La lógica de una interfaz de usuario cambia con más frecuencia que los almacenes de datos y la lógica de negocio. Si realizamos un diseño ofuscado, es decir, un diseño que mezcle los componentes de interfaz y de negocio, entonces la consecuencia será que, cuando necesitemos cambiar el interfaz, tendremos que modificar trabajosamente los componentes de negocio. Mayor trabajo y más riesgo de error.[6]

”El propósito de este patrón es simplificar la implementación de aplicaciones de acuerdo a las peticiones de los usuarios y de los datos a desplegar”.

Se trata de realizar un diseño que desacople la vista del modelo, con la finalidad de mejorar la reusabilidad. De esta forma las modificaciones en las vistas impactan en menor medida en la lógica de negocio o de datos.

Este patrón se ve frecuentemente utilizado en aplicaciones Web, donde la vista es la pagina HTML y el código provee de datos dinámicos a la página. Las aplicaciones Web complejas continúan siendo más difíciles de diseñar que las tradicionales de escritorio, el patrón MVC se presenta como una solución para ayudar a disminuir esta complejidad.[10]

Elementos del patrón

- Modelo: datos y reglas de negocio
- Vista: muestra la información del modelo al usuario

- Controlador: gestiona las entradas del usuario
1. El **modelo** es el responsable de:
 - Acceder a la capa de almacenamiento de datos. Lo ideal es que el modelo sea independiente del sistema de almacenamiento.
 - Define las reglas de negocio (la funcionalidad del sistema). Un ejemplo de regla puede ser: "Si la mercancía pedida no está en el almacén, consultar el tiempo de entrega estándar del proveedor".
 - Lleva un registro de las vistas y controladores del sistema.
 - Si estamos ante un modelo activo, notificará a las vistas los cambios que en los datos pueda producir un agente externo (por ejemplo, un fichero bath que actualiza los datos, un temporizador que desencadena una inserción, etc.). [10]
 2. El **controlador** es responsable de:
 - Recibe los eventos de entrada (un clic, un cambio en un campo de texto, etc.).
 - Contiene reglas de gestión de eventos, del tipo "SI Evento Z, entonces Acción W". Estas acciones pueden suponer peticiones al modelo o a las vistas. Una de estas peticiones a las vistas puede ser una llamada al método "Actualizar()". Una petición al modelo puede ser "Obtener_tiempo_de_entrega(nueva_orden_de_venta)". [10]
 3. Las **vistas** son responsables de:
 - Recibir datos del modelo y los muestra al usuario.
 - Tienen un registro de su controlador asociado (normalmente porque además lo instancia).
 - Pueden dar el servicio de "Actualización()", para que sea invocado por el controlador o por el modelo (cuando es un modelo activo que informa de los cambios en los datos producidos por otros agentes). [10]

1.8.2.1- Ventajas de MVC

Favorece a la óptima utilización de memoria. No se desperdician recursos.

- Modelo compartido, controlador específico.
- Controlador compartido, modelo específico.
- Otras posibilidades.

Fácilmente extensible.

Favorece la abstracción. Simplifica la reutilización de código.

1.8.2.2- Desventajas de MVC

Su principal desventaja es al la hora de la interacción con el usuario, en la parte de editores, donde esta no es su funcionalidad, por lo que hace un gasto masivo e innecesario de memoria, contrario a la ideología de MVC, también puede presentar anomalías fruto de un uso incorrecto de estos editores. Otra desventaja puede ser la accesibilidad.

1.9- Tecnologías usadas

1.9.1- Personal Home Page (PHP)

Es un lenguaje de programación pensado en el Web de forma que es ideal para la creación de páginas dinámicas. PHP es la versión libre del sistema equivalente al de Microsoft, ASP.[12]

1.9.1.1- Ventajas de PHP

- Muy sencillo de aprender.
- Similar en sintaxis a C y a PERL
- Soporta en cierta medida la orientación a objeto. Clases y herencia.
- El análisis léxico para recoger las variables que se pasan en la dirección lo hace PHP de forma automática. Librándose el usuario de tener que separar las variables y sus valores.
- Se puede incrustar código PHP con etiquetas HTML.
- Excelente soporte de acceso a base de datos.
- La comprobación de que los parámetros son validos se hace en el servidor y no en el cliente (como se hace con javascript) de forma que se puede evitar que chequear que no se reciban solicitudes adulteradas. Además PHP viene equipado con un conjunto de funciones de seguridad que previenen la inserción de órdenes dentro de una solicitud de datos.
- Se puede hacer de todo lo que se pueda transmitir por vía HTTP. [12]

1.9.1.2 Desventajas

- Todo el trabajo lo realiza el servidor y no delega al cliente. Por tanto puede ser más ineficiente a medida que las solicitudes aumenten de número.
- La legibilidad del código puede verse afectada al mezclar sentencias HTML y PHP.
- La orientación a objetos es aún muy deficiente para aplicaciones grandes.[12]

1.9.1.3- ¿Por qué utilizar PHP y no otras opciones?

- PHP no soporta directamente punteros, como el C, de forma que no existen los problemas de depuración provocados por estos.
 - Se pueden hacer grandes cosas con pocas líneas de código. Lo que hace que merezca la pena aprenderlo.
 - El código PHP es mucho más legible que el de PERL, todo el que haya programado PERL podrá corroborar esta afirmación.
 - Viene acompañado por una excelente biblioteca de funciones que permite realizar cualquier labor (acceso a base de datos, encriptación, envío de correo, XML, creación de PDF).
 - Al poderse encapsular dentro de código HTML se puede recoger el trabajo del diseñador gráfico e incrustar el código PHP posteriormente.
 - Esta siendo utilizado con éxito en varios millones de sitios Web.
 - Hay multitud de aplicaciones PHP para resolver problemas concretos (weblogs, tiendas virtuales, periódicos,...) listas para usar.
 - Es multiplataforma, funciona en todas las plataformas que soporten apache.
 - Es software libre. Se puede obtener en la Web y su código esta disponible bajo la licencia GPL.
- [12]

1.10- PostgreSQL

Seguramente muchos han tenido más de una vez la duda de que base de datos montar para nuestros servidores.

1.10.1- ¿Qué es PostgreSQL?

PostgreSQL es un Sistema de Gestión de Bases de Datos Objeto-Relacionales (ORDBMS) que ha sido desarrollado de varias formas desde la década de 1980.

PostgreSQL es ampliamente considerado como una de las alternativas de sistema de bases de datos de código abierto y se ha ganado la reputación de ser confiable. Puedes crear varios tipos de datos y hacer herencias entre objetos, tiene transacciones, integridad referencial, vistas, y multitud de funcionalidades, pero es lento y pesado.

PostgreSQL se diseñó como una base de datos orientada a objetos, es decir, una ORDBMS. Esto significa, que las tablas no son tablas, sino objetos, y las tuplas son instancias de ese objeto.

Cumple la prueba ACID (Atomicidad, Consistencia, Integridad, Durabilidad) y tiene soporte completo para llaves foráneas, JOINS, vistas, subconsultas, disparadores (triggers), y procedimientos almacenados (en varios lenguajes). Incluye la mayoría de los tipos de datos de los estándares SQL92 y SQL99. También soporta almacenamiento de objetos grandes (imágenes, sonido y video). Una documentación muy completa (aunque la documentación se encuentra en inglés). Han incorporado la llamada MVCC (multiversion concurrency control) con lo que los bloqueos de escritura actúan sólo en la sesión del cliente, no en las de los demás clientes. [13]

1.10.2- Ventajas de PostgreSQL

- **Instalación Ilimitada**

No hay costo asociado a la licencia del software.

- **Ahorros considerables en costos de operación**

PostgreSQL ha sido diseñado y creado para tener un mantenimiento y ajuste mucho menor que otros productos, conservando todas las características, estabilidad y rendimiento.

- **Extensible**

El código fuente está disponible para todos sin costo. Si se necesita extender o personalizar PostgreSQL de alguna manera, pueden hacerlo con un mínimo esfuerzo, sin costos adicionales.

- **Multiplataforma**

PostgreSQL está disponible en Linux, casi cualquier Unix (34 plataformas en la última versión estable), BeOS y en Windows.

- **Diseñado para ambientes de alto volumen**

PostgreSQL usa una estrategia de almacenamiento de filas llamada MVCC para conseguir una mejor respuesta en ambientes de grandes volúmenes.[13]

1.10.3- Características

- **Soporte SQL92/SQL99**

PostgreSQL implementa un subconjunto extendido de los estándares SQL92 y SQL99

- **Transacciones**

Permiten el paso entre dos estados consistentes manteniendo la integridad de los datos.

BEGIN WORK;

.....

Sentencias SQL;

.....

COMMIT WORK;

- **Integridad referencial**

PostgreSQL soporta integridad referencial, la cual es utilizada para garantizar la validez de los datos de la base de datos PK y FK.

- **Bloqueos de tabla y filas**

Postgres ofrece varios modos de bloqueo para controlar el acceso concurrente a los datos en tablas. Algunos de estos modos de bloqueo los adquiere PostgreSQL automáticamente antes de la ejecución de una declaración, mientras que otros son proporcionados para ser usados por las aplicaciones.

- **Constraints y triggers**

Tienen la función de mantener la integridad y consistencia en la BD.

Ejecución de acciones antes o después de un evento de BD.

- **Múltiples tipos de datos predefinidos**

Como todos los manejadores de bases de datos, PostgreSQL implementa los tipos de datos definidos para el estándar SQL3 y aumenta algunos otros.

- **Soporte de tipos y funciones de usuario**

PostgreSQL soporta operadores, funciones métodos de acceso y tipos de datos definidos por el usuario. Incorpora una estructura de datos Array.

- **Conectividad TCP/IP, JDBC y ODBC**

- **Interfaz con diversos lenguajes**

C, C++, Java, Delphi, Python, Perl, PHP, Bash...[13]

1.11- RRHH-XML

Es un formato estándar de transmisión de datos del área de Recursos Humanos. Al establecer una estructura estándar de datos, HR-XML se está convirtiendo en el "idioma" que todos los departamentos de Recursos Humanos tienden a utilizar y comprender. Si un candidato se comunica directamente con la empresa en el "idioma" que su departamento de RRHH puede gestionar automática y eficazmente, su comunicabilidad y, por extensión, empleabilidad, aumenta espectacularmente.

Todo sistema de comunicación Profesional-Empresa basado en HR-XML constituye un valor añadido de una importancia que el tiempo no hará sino ratificar.[2]

1.11.1- RRHH

Es el consorcio internacional dedicado a definir las etiquetas y la estructura de etiquetas de la información del área de Recursos Humanos. Una de las utilidades de esta taxonomía es estructurar la información curricular de un profesional. El estándar propone una definición genérica y universal de los datos del profesional en apartados como Empleo, Formación, Datos de Contacto, Logros, Competencias Profesionales, Idiomas, etc.[2]

1.11.2- XML

Es el estándar de Extensible Markup Language. XML no es más que un conjunto de reglas para definir etiquetas semánticas que nos organizan un documento en diferentes partes. XML es un metalenguaje que define la sintaxis utilizada para definir otros lenguajes de etiquetas estructurados.

Las Bases de Datos basadas en información formateada en HR-XML son fácilmente mantenibles e inter-operables.

Todas las bases de datos que utilicen el mismo estándar pueden intercambiar ficheros en HR-XML sin excesivos costes de adaptación y transmisión. Gracias a ellos las bases se conectan entre sí con facilidad, ya no son islas de información independiente e incomunicada, y pueden constituirse fácilmente en una red de comunicación automatizada.[4]

1.11.3- HTML

Para publicar información y distribuirla globalmente, se necesita un lenguaje entendido universalmente, una especie de lengua franca de publicación que todas las computadoras puedan comprender potencialmente. El lenguaje de publicación usado por la World Wide Web es el HTML (acrónimo de HyperText Markup Language, Lenguaje para el Formato de Documentos de Hipertexto).

El HTML da la posibilidad para:

- Publicar documentos en línea con encabezados, textos, tablas, listas, fotos, etc.
- Obtener información en línea a través de vínculos de hipertexto, haciendo clic con el botón de un ratón.
- Diseñar formularios para realizar transacciones con servicios remotos, para buscar información, hacer reservas, pedir productos, etc.
- Incluir hojas de cálculo, vídeo clips, sonidos, y otras aplicaciones directamente en sus documentos. [5]

1.12- Rational Unified Process (RUP)

El mundo de la informática no para de hablar de procesos de desarrollo, el modo de trabajar eficientemente para evitar catástrofes que llevan a un gran porcentaje que llevan a que un gran porcentaje de proyectos se terminen sin éxito. El objetivo de un proceso de desarrollo es dar una mayor calidad al software a través de una mayor transparencia y control sobre el proceso.

En los últimos tiempos la cantidad y variedad de los procesos de desarrollo ha aumentado de forma impresionante, Se podría decir que en estos últimos años se han desarrollado dos corrientes en lo referente a los procesos de desarrollo, los llamados métodos pesados y los métodos ligeros. La diferencia fundamental entre ambos es que mientras los métodos pesados intentan conseguir el objetivo común por medio de orden y documentación, los más ligeros (también llamados métodos ágiles) tratan de mejorar la calidad de software por medio de una comunicación directa e inmediata entre las personas que intervienen en el proceso.

RUP pertenece a lo métodos pesados, a su vez es uno de los procesos más generales de los existentes actualmente, ya que en realidad está pensado para adaptarse a cualquier proyecto, no solamente de software.

RUP es un proceso de ingeniería de software (forma disciplinada de asignar tareas y responsabilidades en una organización de desarrollo), que tiene como objetivo asegurar la producción de software de calidad dentro de plazos y presupuestos predecibles. Aumenta la productividad de los desarrolladores mediante acceso a bases de conocimientos, plantillas y herramientas. Se centra en la producción y mantenimiento de modelos del sistema más que en producir documentos. Es una guía de cómo usar UML (Lenguaje Unificado de Modelado) de la forma más efectiva, existen herramientas de apoyo a todo el proceso: modelamiento visual, programación, pruebas, etc.

RUP es iterativo e incremental (permite una comprensión creciente de los requerimientos a la vez que se va haciendo crecer el sistema).

RUP pretende implementar las mejores prácticas actuales en ingeniería de software:

- Desarrollo Iterativo del Software.
- Administración de Requerimientos.
- Uso de Arquitecturas Basadas en Componentes.
- Modelamiento Visual del Software.
- Verificación de la Calidad del Software.
- Control de Cambios.

Verificación de Cualidades es otra de sus ventajas, en un producto no sólo la funcionalidad es esencial, también el rendimiento y la confiabilidad, esta metodología ayuda a planificar, diseñar, implementar, ejecutar y evaluar pruebas que verifiquen estas cualidades, el aseguramiento de la calidad es parte del proceso de desarrollo y no la responsabilidad de un grupo independiente.

Un proyecto realizado siguiendo RUP se divide en cuatro fases:

- 1- Inicio (puesta en marcha).
- 2- Elaboración (definición, análisis y diseño).
- 3- Construcción (implementación).
- 4- Transición (fin del proyecto y puesta en producción).

RUP se basa en casos de uso para descubrir lo que se espera del software y esta muy orientado a la arquitectura del sistema, documentándose lo mejor posible, basándose en UML como herramienta principal.

RUP es un proceso muy general y muy grande, por lo que antes de usarlo habrá que adaptarlo a las características de la empresa.[8]

1.13- Lenguaje Unificado de Modelado (UML)

UML es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. Se usa para entender, diseñar, configurar, mantener y controlar la información sobre los sistemas a construir.[3]

1.14- Herramientas Utilizadas

1.14.1- Dreamweaver 8

Proporciona una potente combinación de herramientas visuales de diseño, funciones de desarrollo de aplicaciones y soporte para la edición del código, características todas ellas que permiten a los desarrolladores y diseñadores más expertos o menos expertos crear rápidamente sitios Web y aplicaciones basados en estándares.

Dreamweaver 8 se ha creado a partir de las versiones anteriores, pero se ha ampliado su funcionalidad en ámbitos esenciales como XML y CSS, y se ha agilizado el flujo de trabajo para que los usuarios puedan hacer más cosas en menos tiempo. Al incorporar las más avanzadas metodologías de diseño profesional, Dreamweaver 8 pone de manifiesto el potencial que representa para la comunidad Web el desarrollo abierto basado en estándares y mejora, al mismo tiempo, las herramientas básicas que necesitan los profesionales. Dreamweaver es compatible con todas las principales tecnologías de

servidor como, por ejemplo, ColdFusion, PHP, ASP, ASP.NET y JSP, para que los desarrolladores, más expertos o menos expertos, puedan dar vida a sus diseños.

Por mejores prácticas entendemos las tecnologías y metodologías de desarrollo que permiten a los usuarios desarrollar de forma rápida y eficiente aplicaciones Web de calidad y conformes con los estándares. Trabajar con las mejores prácticas es una garantía de que los sitios Web permitirán a los usuarios impulsar la envoltura de la Web y, al mismo tiempo, funcionarán sin problemas en todos los navegadores, con la consiguiente reducción de los costes de mantenimiento. Dreamweaver, todo resulta tan simple como arrastrar y colocar. Dreamweaver 8 incorpora las mejoras prácticas y los estándares de la industria, entre los que cabe mencionar el soporte avanzado para CSS, entradas XML y RSS y requisitos de accesibilidad.[9]

1.14.2- Visual Paradigm

Visual Paradigm para UML es una de las herramientas UML CASE del mercado, considerada como muy completa y fácil de usar, con soporte multiplataforma y que proporciona excelentes facilidades de interoperabilidad con otras aplicaciones. Fue creada para el ciclo vital completo del desarrollo del software que lo automatiza y acelera, permitiendo la captura de requisitos, análisis, diseño e implementación. Visual Paradigm-UML también proporciona características tales como generación del código, ingeniería reversa y generación de informes. Tiene la capacidad de crear el esquema de clases a partir de una base de datos y crear la definición de base de datos a partir del esquema de clases. Permite invertir código fuente de programas, archivos ejecutables y binarios en modelos UML al instante, creando de manera simple toda la documentación. Está diseñada para usuarios interesados en sistemas de software de gran escala con el uso del acercamiento orientado a objeto, además apoya los estándares más recientes de las notaciones de Java y de UML. Incorpora el soporte para trabajo en equipo, que permite que varios desarrolladores trabajen a la vez en el mismo diagrama y vean en tiempo real los cambios hechos por sus compañeros.

1.14.3- PG_ADMIN

Esta herramienta normalmente viene con la distribución de PostgreSQL Database, es multiplataforma, se puede instalar ya sea en Linux, Solaris, Windows, etc. Tiene muchas opciones para ahorrarnos la tarea de crear la base de datos a líneas de comando. El uso de esta herramienta no supe el conocimiento que se debe de tener al administrar una base de datos pero si facilita el manejo del SGBD. El pgAdmin es una de las herramientas más populares para administrar bases de datos, tiene abundante posibilidades

(opciones) a la hora de administrar estas, pertenece a la familia de software de Open Source (código abierto), esta plataforma es desarrollada para la administración de PostgreSQL.

La aplicación puede usarse en Linux, FreeBSD, OpenSUSE, Solaris, Mac OSX y plataformas de Windows para la gestión de PostgreSQL 7.3 y versiones superiores, no es diseñado solamente para la gestión de bases de datos en PostgreSQL, sino que también puedes usar versiones comerciales de este y derivados como son: EnterpriseDB, Mammoth PostgreSQL, Greenplum Bizgres y base de datos. Se diseña para responder las necesidades de todos los usuarios, desde simples sentencias SQL a las bases de datos, hasta otras más complejas.

El apoyo en la interfaz gráfica ofrece y hace la administración fácil. La aplicación también incluye un editor de sintaxis SQL, un servidor, editor de código, un SQL / lote / shell de la programación de agente de empleo, el apoyo a la replicación Slony-I del motor y mucho más. La conexión con el servidor puede ser realizada utilizando TCP / IP o Unix Domain Sockets (sobre plataformas * nix), y puede ser cifrado SSL para la seguridad, no se requieren drivers adicionales para comunicarse con el servidor de base de datos. Posee una amplia gama de documentación, posee una interna multilingüe, el acceso a los datos, acceso nativo a PostgreSQL (no necesita capa ODBC), potente herramienta de consulta con el color de sintaxis, muy rápido datagrid para la visualización y entrada de datos.[7]

1.14.4- Apache

Apache es hoy en día el servidor WEB más utilizado en todo el mundo. Funciona sobre infinidad de sistemas y arquitecturas, se trata de un programa libre, y que su código fuente está disponible. Su proceso de instalación 'nativo' no es sencillo. Sin embargo todas las distribuciones de Linux y otros sistemas UNIX incorporan este magnífico software en su listado de paquetes posibilitando la instalación usando las herramientas del sistema.

La flexibilidad que ofrece Apache es difícil de encontrar en otros servidores Web. Ofrece de todo; desde la posibilidad de usarlo como Proxy cache o Proxy inverso, a usarlo como balanceador de carga o servidor de aplicaciones.

Además puedes incluir toda la suite propia de un servidor de aplicaciones: Pearl, PHP, SQL y Java, con sus correspondientes módulos. Es decir, que a través de apache, y con las correspondientes aplicaciones puedes mantener todo un servicio de información a través de Internet, consulta a base de datos todo tipo de peticiones y envíos.

Apache incluye además la implementación de la seguridad SSL, la más utilizada en Internet, por lo que se pueden cifrar y enviar datos de forma segura.

Sin embargo toda esta flexibilidad genera un grado de complejidad que es posible que algunos administradores de sistemas no están dispuestos a asumir.[11]

Conclusiones

A modo de resumen, en este capítulo se profundizó principalmente en el surgimiento de las bolsas en el mundo, así como su aceptación en la población desarrollada principalmente, por su fácil operación y agilización del proceso de contrata. Además se expuso con argumentos, las herramientas a utilizar, así como la arquitectura, las tecnologías a aplicar y la diversidad de ventajas y desventajas que en ellas se aplican para la fácil y confiable creación de nuestra aplicación.

CAPITULO II: CARACTERÍSTICAS DEL SISTEMA.

Introducción

En el presente capítulo se hace un análisis profundo del funcionamiento del negocio, así como su representación gráfica y su descripción. Se modela el Sistema y se plantean todos los conceptos relacionados con el mismo. Nos representa además los actores que interactúan con el sistema, así como su función en el mismo y enumera los requisitos funcionales y no funcionales que debe tener la aplicación, lo que proporciona un primer acercamiento al sistema.

2.1- Análisis de los procesos actuales

Actualmente los procesos se ejecutan de una forma bastante tediosa, donde la empresa informa por distintas vía, ya sea por radio o por propia divulgación de los trabajadores la necesidad de completar alguna plaza bacante existente en la entidad, después de esta parte del proceso, el desempleado que opta por le puesto de trabajo se dirige a la empresa con su currículum y solicita el empleo, ahí es cuando el departamento de recursos humanos le realiza todo el proceso, realiza la entrevista al optante, pide el expediente, este es revisado por el departamento y en caso de ser aceptado es llevado al área de recursos humanos para preparar todo lo referente a la inscripción del optante como trabajador.

2.2- Objeto de automatización

De todo este conjunto de procesos varios pueden automatizarse como son: informar por parte de la empresa la plaza vacante, solicitar empleo, entregar currículum por parte del optante y avisarle al desempleado que ha sido aceptado para el puesto de trabajo.

2.3- Descripción general de la propuesta de sistema

El sistema se encargará de automatizar lo más posible la selección de un obrero para la ubicación laboral. Esta aplicación Web permitirá que el usuario opte por una plaza laboral disponible en una empresa determinada.

El usuario al entrar a la aplicación Web podrá ver las ofertas laborales disponibles que existen.

Si desea optar por alguna de estas ofertas tendrá que registrarse.

Si ya se registro entonces iría a la opción de autenticarse, de lo contrario tendría que ir a la opción de nuevo usuario.

Aquí se le pediría una variedad de datos personales como son el nombre, carné de identidad, dirección, nivel de escolaridad, entre otros más que juntos conforman la planilla de inscripción.

Después de inscrito ya puede optar por algún puesto de trabajo de los ofertados en el sitio.

El sistema le dará la posibilidad al usuario de en algún momento dado de actualizar estos datos personales.

También el usuario tendrá la posibilidad de ver su expediente cada vez que desee.

El usuario tendrá la posibilidad de obtener un reporte de su currículo en formato RRHH-XML.

2.4- Modelo del negocio

A la hora de lograr sus objetivos, una empresa desarrolla su actividad por un flujo continuo de procesos de negocio. Cada uno de estos procesos se caracteriza por una serie de datos que son producidos y manipulados mediante un conjunto de tareas, en las que ciertos agentes (por ejemplo trabajadores) participan de acuerdo a un flujo de trabajo determinado. Además, estos procesos se hallan sujetos a un conjunto de reglas de negocio, que determinan la estructura de la información y las políticas de la empresa. Por tanto, la finalidad del modelado del negocio es describir cada proceso del negocio, especificando sus datos, actividades (o tareas), roles (o agentes) y reglas de negocio.

Los objetivos del modelamiento del negocio son:

- Comprender la estructura y la dinámica de la organización en la cual se va a implantar un sistema.
- Comprender los problemas actuales de la organización e identificar las mejoras potenciales.
- Asegurar que los consumidores, usuarios finales y desarrolladores tengan un entendimiento común de la organización.
- Derivar los requerimientos del sistema que va a soportar la organización.

Actor del negocio	Justificación
Desempleado	Es el que inicia el proceso, primeramente busca ofertas de trabajo y posteriormente puede optar por una plaza laboral determinada.

Trabajador del negocio	Justificación
Empresa	Es la encargada de seleccionar al obrero para el puesto de trabajo.

2.5- Diagrama de casos de uso del negocio



2.6- Descripción del caso de uso del negocio.

Nombre del CU del Negocio	Buscar ofertas de trabajo
Actores del negocio:	Desempleado
Trabajadores del negocio:	Empresa
Resumen: Se inicia cuando el desempleado busca alguna empresa que este ofertando puestos de trabajo, y esta le informa si tiene (le da una descripción del puesto) o no alguna vacante laboral.	
Acción del actor	Respuesta del negocio
1- El desempleado va a una empresa en busca de alguna oferta de trabajo.	
	2- La empresa le muestra las plazas ofertadas.
	3- La empresa describe las características de cada empleo ofertado.
Prioridad:	-
Mejoras:	Al automatizar este proceso se podrá ver las plazas de trabajo ofertadas por una empresa con sus descripciones sin la necesidad de ir hasta ella.
Cursos alternos:	

Nombre del CU del Negocio	Solicitar
Actores del negocio:	Desempleado
Trabajadores del negocio:	Empresa
Resumen: Se inicia cuando el desempleado solicita algún puesto laboral ofertado, donde la empresa empieza a realizar los trámites para aceptar o denegar al desempleado que opta por el puesto de trabajo.	
Acción del actor	Respuesta del negocio
1- El desempleado solicita un puesto laboral.	
	2- La empresa pide el expediente.
	3- La empresa empieza a revisar el expediente para ver si acepta o rechaza el pedido del desempleado.
Prioridad:	-
Mejoras:	Al automatizar este proceso se podrá optar por una plaza laboral sin necesidad de presentarse en la empresa.
Cursos alternos:	

2.7- Diagramas de Actividades del Negocio

Diagrama de actividad de: Buscar ofertas de trabajo

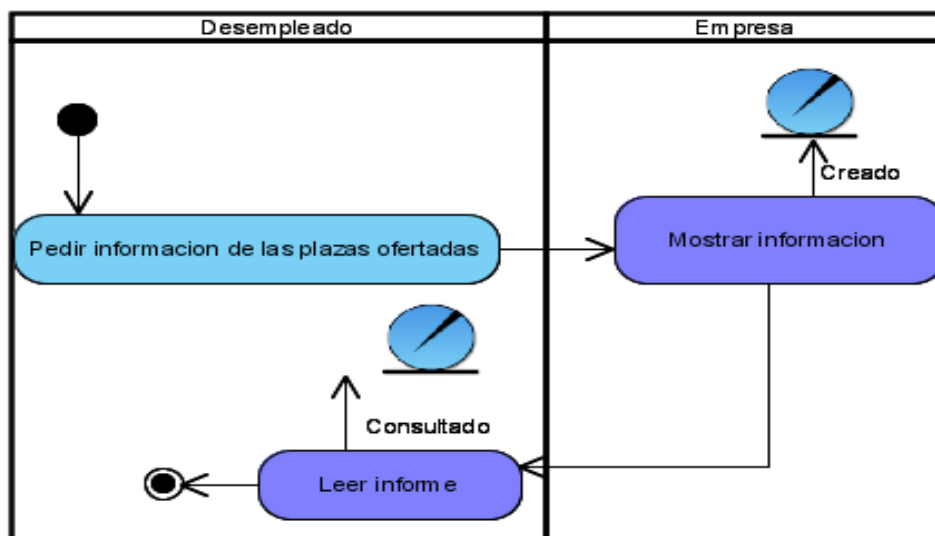
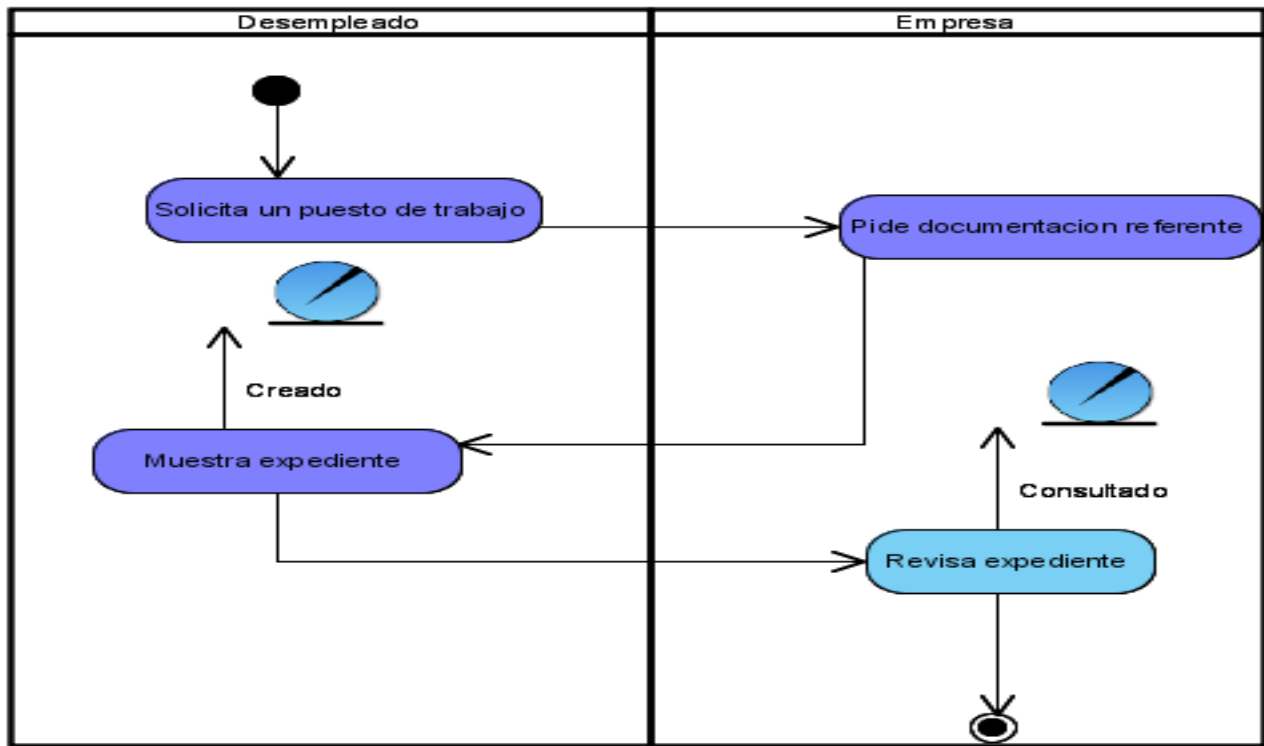
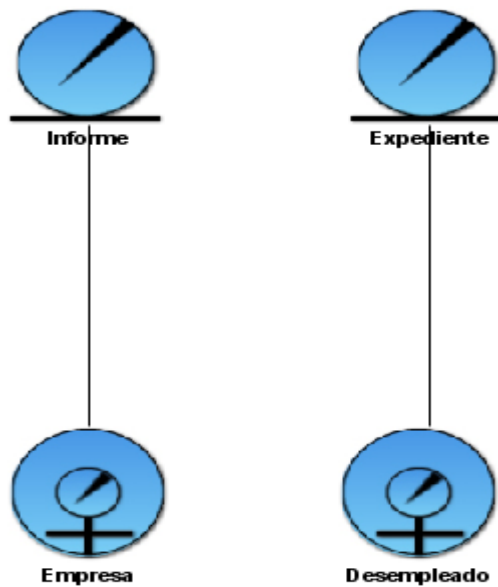


Diagrama de actividad de: Solicitar



2.8- Modelo de objeto del negocio



2.9- Requerimientos Funcionales

RF1: Mostrar los anuncios existentes.

RF2: Permitir gestionar expediente laboral del desempleado (crear, modificar).

RF3: Poder optar por una plaza.

RF4: Dar la posibilidad al desempleado de ver sus datos.

RF5: Poder por parte de la empresa revisar un expediente determinado.

RF6: Dar la posibilidad a la empresa de publicar anuncios, o eliminar alguno que haya realizado.

RF7: Poder el desempleado generar en un formato RRHH-XML su expediente.

RF8: Permitir crear y modificar un usuario.

RF9: Permitir buscar solicitud de trabajo en otra provincia.

RF10: Avisar al optante en caso de ser seleccionado.

RF11: Generar estadísticas de las ofertas realizadas en un período determinado.

RF12: Gestionar datos del usuario.

RF13: Actualizar nomenclador instituciones.

RF14: Actualizar nomenclador provincia.

RF15: Actualizar nomenclador municipios.

RF16: Actualizar nomenclador rol.

RF17: Actualizar nomenclador especialidad.

RF18: Actualizar nomenclador grado.

2.10- Requerimientos no funcionales

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable.

Apariencia o interfaz externa:

- ✓ Diseño sencillo, con pocas entradas, permitiendo que no sea necesario mucho entrenamiento para utilizar el sistema.

Usabilidad:

- ✓ El software tendrá siempre la posibilidad de ayuda disponible para cualquier tipo de usuario.

Soporte:

- ✓ Se requiere un servidor de bases de datos con las siguientes características:
 - Soporte para grandes volúmenes de datos y velocidad de procesamiento.

- Tiempo de respuesta rápido en accesos concurrentes.
- ✓ Por parte del cliente se requiere un navegador capaz de interpretar JavaScript.
- ✓ Versión de PHP 5.0.

Portabilidad:

- ✓ Necesidad de que el sistema sea multiplataforma.

Seguridad:

- ✓ Autenticación (contraseña de acceso)
- ✓ Garantizar que las funcionalidades del sistema se muestren de acuerdo al nivel de usuario que este activo.
- ✓ Protección contra acciones no autorizadas o que puedan afectar la integridad de los datos.
- ✓ Verificación sobre acciones irreversibles (eliminaciones).

Software:

- ✓ Una computadora personal con plataforma del sistema operativo Windows 2000 o superior; o Linux.
- ✓ Apache 2.0 o superior como servidor Web, con módulo PHP 5 disponible y debe estar configurado con la extensión pgsql incluida.
- ✓ PostgreSQL como Sistema Gestor de Base de Datos.

Hardware:

El usuario final debe tener un servidor con las siguientes características:

- ✓ Tarjeta de red.
- ✓ 128 Mb. de RAM o superior.
- ✓ 40 Gb. de disco duro o superior.
- ✓ Pentium II a 133 MHz de velocidad en su procesador o más.

Una computadora que sirva de cliente:

- ✓ Pentium a 200 MHz. de velocidad de procesamiento o superior.
- ✓ 32 Mb. de memoria RAM superior.
- ✓ Tarjeta de red.

2.11- Casos de uso y actores del sistema

2.11.1- Definición de los actores

Actores	Justificación
Desempleado	Es la persona que esta sin trabajo y esta buscando alguna plaza laboral.
Empresa	Es la entidad que necesita cubrir alguna plaza laboral bacante.
Usuario	Es el desempleado y la empresa para la aplicación, definiéndose según el tipo de usuario el privilegio de acceso.
Administrador	Es el encargado de administrar el sitio.

2.11.2- Listado de casos de uso

CU1	Mostrar ofertas
Actor	Desempleado
Descripción	Mediante la aplicación el desempleado genera una lista con todas las ofertas de empleos existentes.
Referencia	RF1

CU2	Gestionar expediente
Actor	Desempleado
Descripción	Después de haberse registrado el desempleado, este tiene la opción de crear su currículum; si ya se registro lo puede modificar.
Referencia	RF2

CU3	Solicitar plaza
Actor	Desempleado
Descripción	EL desempleado al revisar las ofertas laborales si se interesa por alguna, realiza la solicitud de esta.
Referencia	RF3

CU4	Ver datos
Actor	Desempleado
Descripción	El desempleado solicita a la aplicación ver sus datos, y esta es la encargada de consultar la base de datos y mostrarlos.
Referencia	RF4

CU5	Revisar expedientes
Actor	Empresa
Descripción	Si en algún momento la empresa le interesa ver el expediente de algún solicitante esta va a la opción de revisar expedientes y ahí revisa el currículum.
Referencia	RF5

CU6	Gestionar anuncio
Actor	Empresa
Descripción	La empresa publica o elimina un nuevo anuncio, dando una pequeña descripción del puesto laboral ofertado o elimina uno que ya fue publicado.
Referencia	RF6

CU7	Generar reportes
Actor	Empresa
Descripción	El desempleado para su fácil manipulación puede generar su currículum en un formato RRHH-XML.
Referencia	RF7

CU8	Gestionar registro de usuario
Actor	Usuario
Descripción	Al acceder al sitio para poder optar por los beneficios que brinda, hay que ser usuario registrado.
Referencia	RF8

CU9	Buscar solicitud en otra provincia
Actor	Desempleado
Descripción	Al acceder al sitio al usuario solamente se le muestran las ofertas laborales de su provincia, para poder acceder a otras provincias.
Referencia	RF9

CU10	Avisar
Actor	Empresa
Descripción	La empresa brinda a través del sitio información acerca de los usuarios aceptados para la plaza laboral.
Referencia	RF10

CU11	Generar estadística
Actor	Usuario
Descripción	El usuario tiene la posibilidad de ver con que frecuencia ofertan plazas.
Referencia	RF11

CU12	Gestionar datos del usuario
Actor	Usuario
Descripción	En caso del que usuario cambie su dirección, teléfono o algún otro dato, debe modificarlo para así tener actualizada la base de datos.
Referencia	RF12

CU13	Actualizar nomenclador instituciones
Actor	administrador
Descripción	El administrador del sitio debe de conformar una lista con las instituciones laborales, teniendo la posibilidad de adicionar o eliminar alguna institución.
Referencia	RF13

CU14	Actualizar nomenclador provincia
Actor	administrador
Descripción	El administrador del sitio debe de conformar una lista con las provincias, teniendo la posibilidad de adicionar o eliminar alguna provincia.
Referencia	RF14

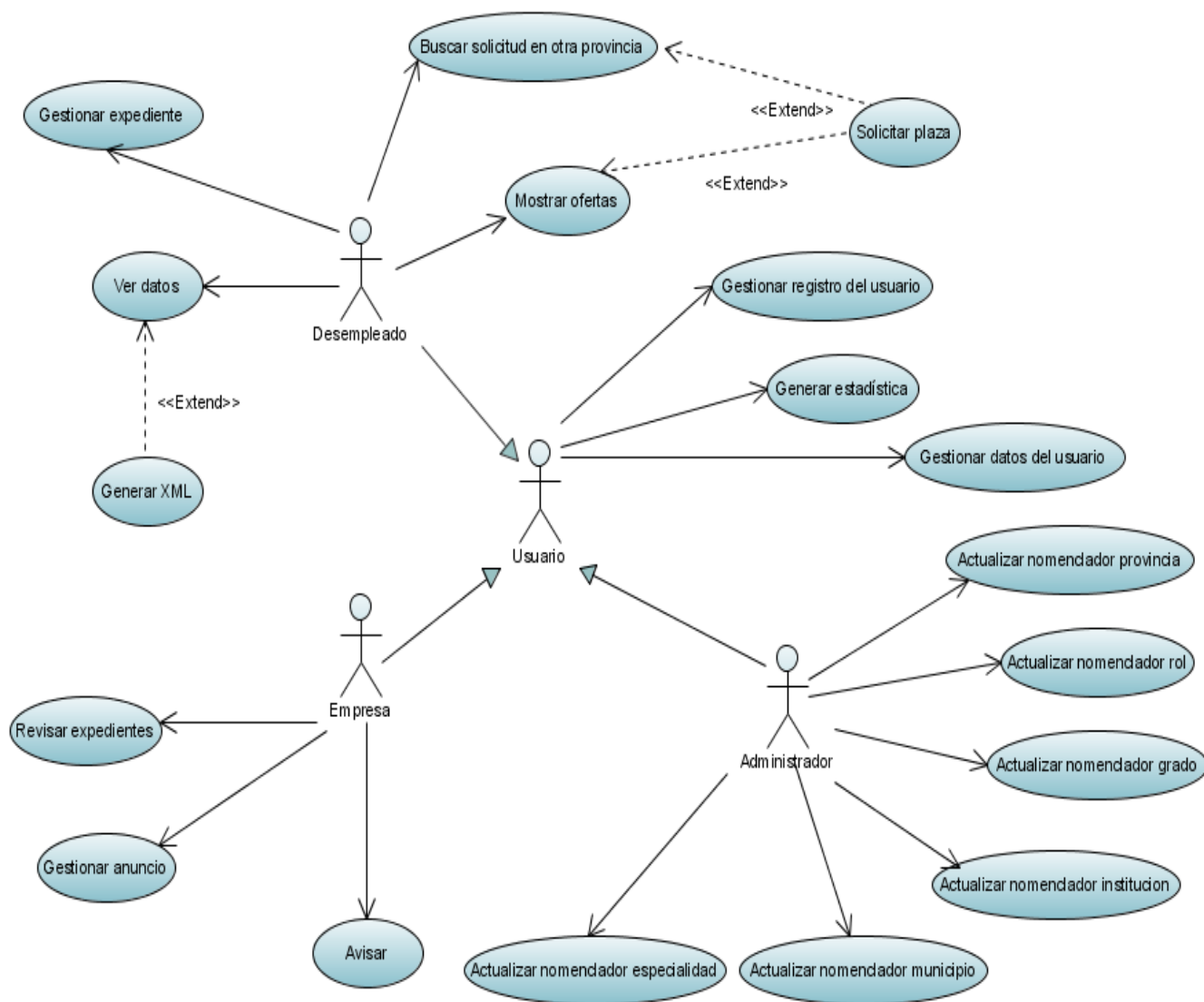
CU15	Actualizar nomenclador municipios
Actor	administrador
Descripción	El administrador del sitio debe de conformar una lista con los municipios, teniendo la posibilidad de adicionar o eliminar algún municipio.
Referencia	RF15

CU16	Actualizar nomenclador rol
Actor	administrador
Descripción	El administrador del sitio debe conformar una lista de los roles existentes, teniendo la posibilidad de adicionar o eliminar algún rol.
Referencia	RF16

CU17	Actualizar nomenclador especialidad
Actor	administrador
Descripción	El administrador del sitio debe de conformar una lista con las especialidades profesionales, teniendo la posibilidad de adicionar o eliminar alguna especialidad.
Referencia	RF17

CU18	Actualizar nomendador grado
Actor	Administrador
Descripción	El administrador del sitio debe de conformar una lista con los grados escolares, teniendo la posibilidad de adicionar o eliminar algún grado.
Referencia	RF18

2.12- Diagrama de Casos de Uso del Sistema



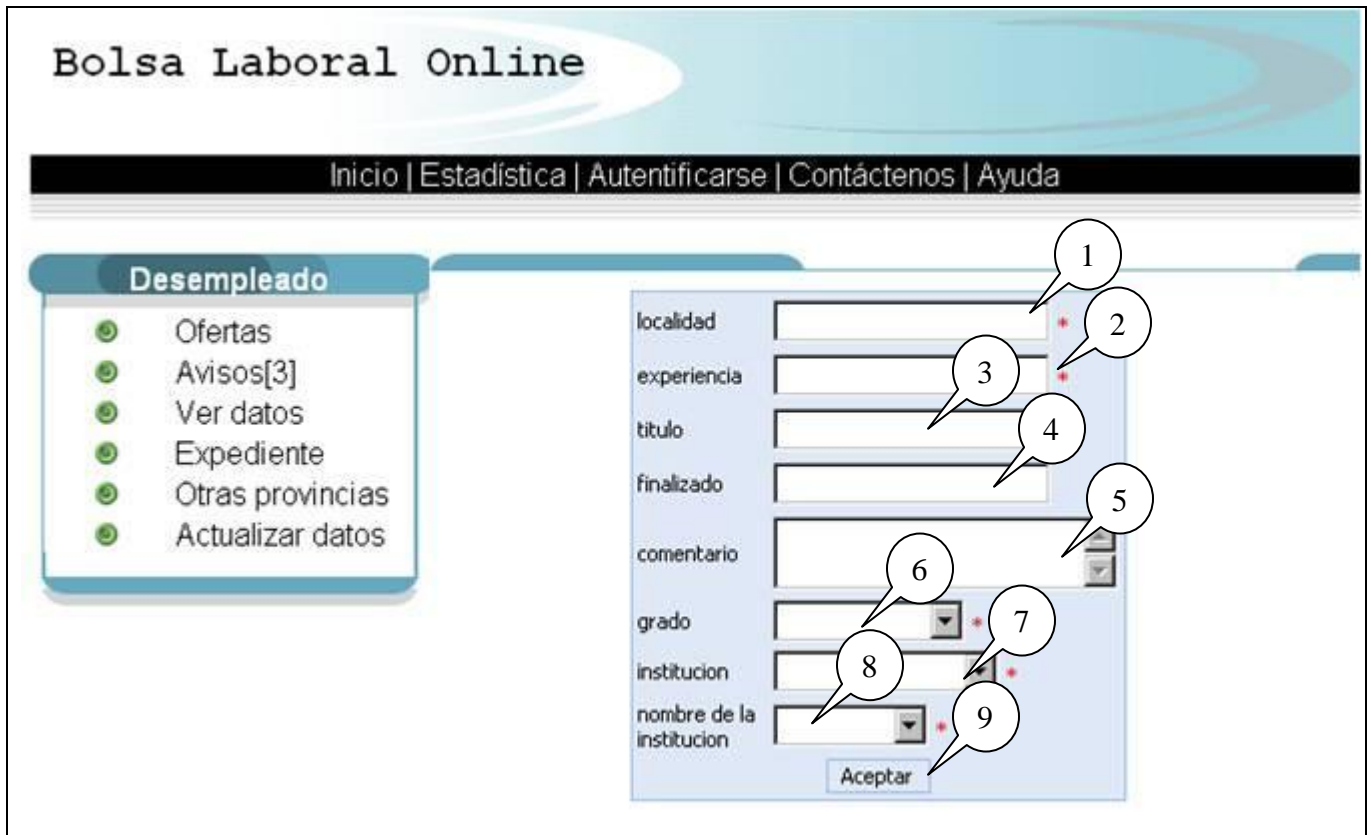
2.13- Descripción de los Casos de uso expandida

Caso de uso	Mostrar ofertas
Actor(es):	Desempleado(inicia)
Propósito:	Mostrar una lista con todas las plazas ofertadas para una provincia.
Resumen:	El caso de uso se inicia cuando el desempleado quiere ver todas las ofertas de trabajo publicadas por las distintas empresas en su provincia y el sistema genera la lista con estas plazas laborales.
Precondiciones:	El desempleado debe de estar registrado y autenticado en el sitio.
Pos-condiciones:	El sistema debe de mostrar en una lista todas las plazas laborales ofertadas en esa provincia.
Tipo:	Real y expandido.
Responsabilidades:	RF1.
Casos de usos relacionados:	Solicitar plaza
Interfaz I	



(1) Vínculo el cual muestra las ofertas laborales.	
Curso normal de eventos para el caso de uso	
Acción del actor	Respuesta del sistema
1- El desempleado selecciona la opción "Ofertas".(1)	2- El sistema realiza las consultas a las tablas de la base de datos.
	3- El sistema muestra una lista con todas las ofertas laborales y sus descripciones para la provincia donde reside el desempleado.
Cursos alternos	

Caso de uso	Gestionar expediente
Actor(es):	Desempleado.
Propósito:	Actualizar el currículum laboral.
Resumen:	El caso de uso se inicia cuando el desempleado quiera introducir algún dato de su currículum laboral o cuando quiera modificar el existente, guardando así estos valores en la base de datos.
Precondiciones:	El desempleado debe de estar registrado en el sitio.
Pos-condiciones:	
Tipo:	Real y expandido.
Responsabilidades:	RF2.
Casos de usos relacionados:	
Interfaz I	



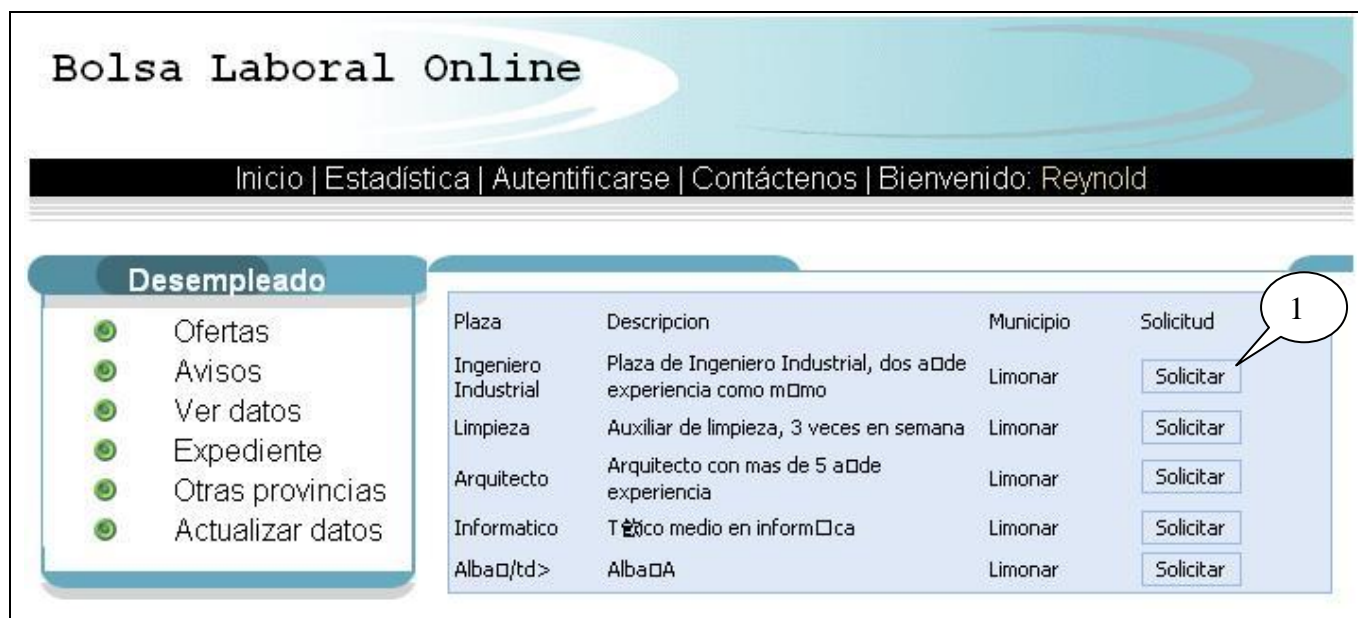
Interfaz II



Interfaz III	
<p>(1) Localidad en la que estudió. (2) Años que lleva de experiencia laboral. (3) Título que obtuvo al graduarse. (4) Fecha en la que finalizó sus estudios. (5) Algún comentario extra que desee realizar. (6) Grado de escolaridad que presenta. (7) Tipo de institución en la que se graduó (8) Nombre de la institución anterior. (10) Vínculo que hacia la interfaz de actualizar expediente.</p>	
Curso normal de eventos para el caso de uso	
Acción del actor	Respuesta del sistema
Sección Insertar expediente	
	1- El sistema muestra una serie de campos, insertar (1), experiencia (2), título (3), finalizado (4), comentario (5), grado (6), institución (7),

	nombre de la institución (8).
2- El desempleado llena los campos.	
3- El desempleado presiona el botón “Aceptar” (9).	4- El sistema crea el expediente guardándolo en la base de datos.
Sección Actualizar expediente	
1- El desempleado va a la opción de “Expediente” (10).	2- El sistema realiza las consultas a las tablas de la base de datos donde obtiene todo lo relacionado con el currículum del desempleado.
	3- El sistema muestra todos los datos del currículum del desempleado.
4- El desempleado modifica los datos que desee.	
5- El desempleado presiona el botón “Actualizar” (11).	6- Con los datos introducidos el sistema actualiza la base de datos.
Cursos alternos	

Caso de uso	Solicitar plaza
Actor(es):	Desempleado.
Propósito:	Optar por una plaza laboral.
Resumen:	El caso de uso se inicia cuando el desempleado opta por una plaza laboral haciéndole el pedido al sistema.
Precondiciones:	El sistema debe de haber listado todas las plazas laborales ofertadas para esa provincia.
Pos-condiciones:	
Tipo:	Real y expandido.
Responsabilidades:	RF3.
Casos de usos relacionados:	
Interfaz I	



(1) Solicitar la plaza

Curso normal de eventos para el caso de uso

Acción del actor	Respuesta del sistema
1- El desempleado va a la opción de "Solicitar plaza".	2- El sistema realiza las consultas a las tablas de la base de datos.
	3- El sistema muestra una lista con todas las ofertas laborales y sus descripciones para la provincia donde reside el desempleado.
4- El desempleado opta por alguna de las plazas ofertadas presionando el botón "Solicitar" (1).	5- El sistema guarda en la base de datos el desempleado y la plaza que solicitó.

Cursos alternos

Caso de uso	Ver datos
Actor(es):	Desempleado.
Propósito:	Ver por parte del desempleado todos sus datos personales y su currículum.
Resumen:	El caso de uso se inicia cuando el desempleado le interesa ver sus datos personales y laborales y el sistema le muestra esos datos.
Precondiciones:	

Pos-condiciones:	
Tipo:	Real y expandido.
Responsabilidades:	RF4.
Casos de usos relacionados:	

Interfaz I



(1) Vínculo el cual muestra los datos del desempleado.

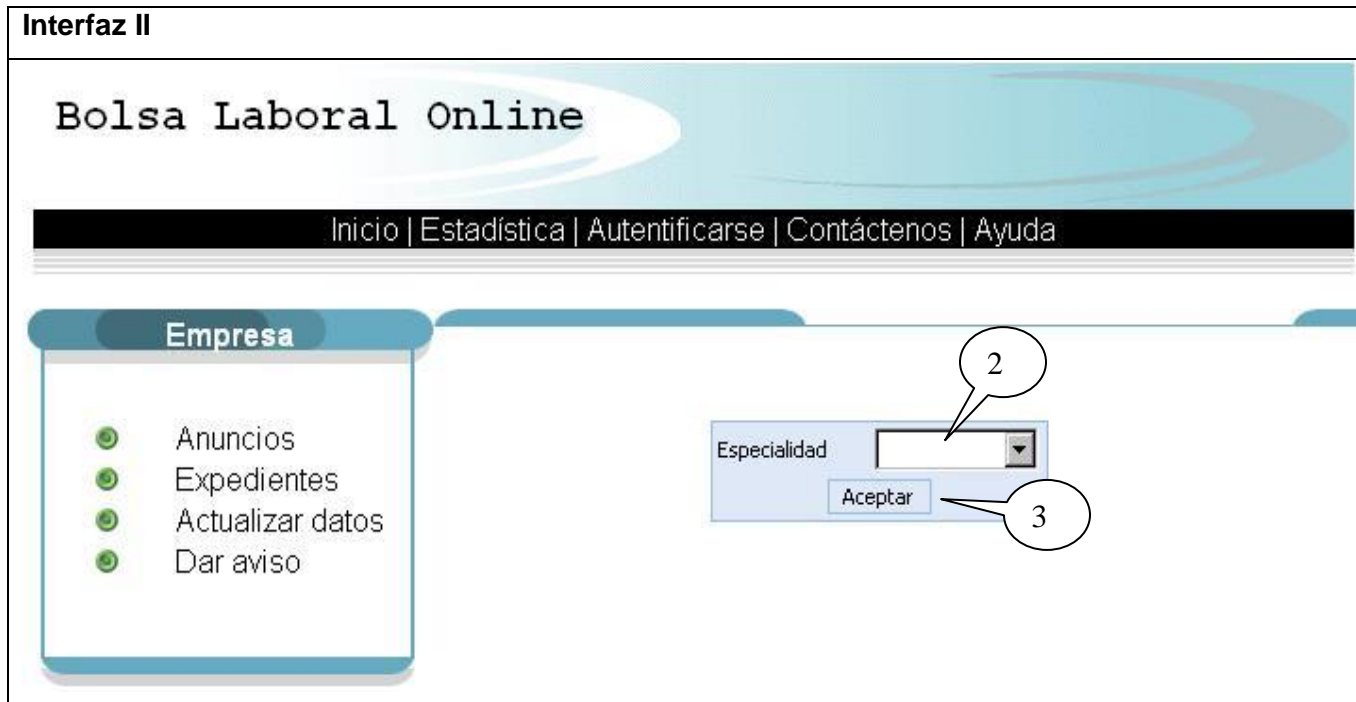
Curso normal de eventos para el caso de uso

Acción del actor	Respuesta del sistema
1- El desempleado va a la opción "Ver datos" (1).	2- El sistema realiza las consultas a las tablas de la base de datos.
	3- El sistema muestra todos los datos del usuario, organizados en datos personales y datos laborales.

Cursos alternos

Caso de uso	Revisar expedientes
Actor(es):	Empresa
Propósito:	Revisar los expedientes de los usuarios.
Resumen:	El caso de uso se inicia cuando la empresa se interesa por revisar algún expediente, donde la aplicación le muestra el o los expediente(s) solicitado(s).
Precondiciones:	
Pos-condiciones:	
Tipo:	Real y expandido.
Responsabilidades:	RF5
Casos de usos relacionados:	





(1) Vínculo el cual muestra los datos del desempleado.

(2) Especialidad a buscar

(4) Ver expediente

Curso normal de eventos para el caso de uso

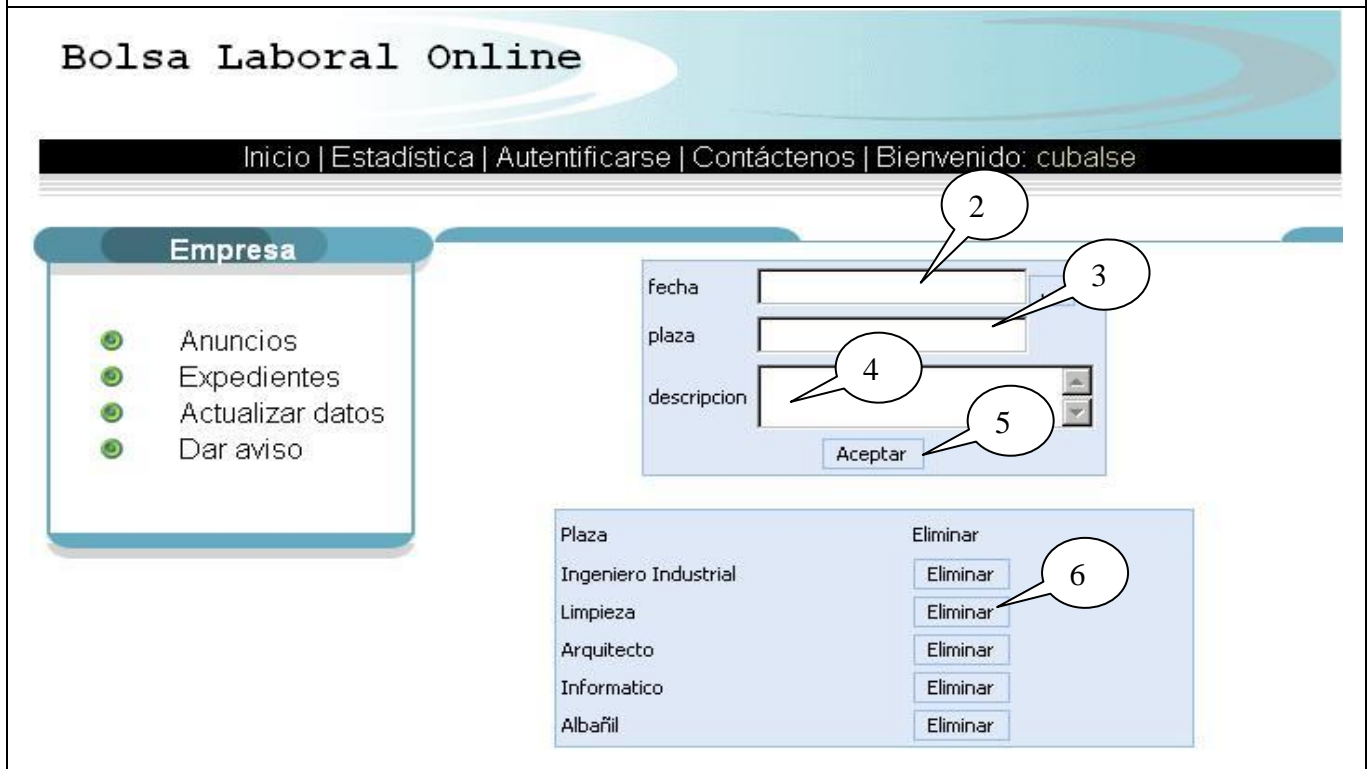
Acción del actor	Respuesta del sistema
------------------	-----------------------

1- La empresa va a la opción “Expedientes” (1).	2- El sistema le muestra a la empresa una lista con las especialidades existentes.
3- La empresa selecciona la especialidad deseada (2).	
4- La empresa presiona el botón “Aceptar” (3).	5- El sistema realiza las consultas a las tablas de la base de datos.
	6- El sistema muestra una lista con todos los desempleados que son de esa especialidad.
7- La empresa selecciona al desempleado al cual le quiera ver el expediente presionando el botón “Ver” (4).	8- El sistema muestra todos los datos de ese desempleado.
Cursos alternos	

Caso de uso	Gestionar anuncio
Actor(es):	Empresa
Propósito:	Publicar una oferta laboral.
Resumen:	El caso de uso se inicia cuando la empresa desea ofertar o eliminar una plaza laboral, donde la empresa publica su anuncio en la aplicación o elimina alguno que ya realizó.
Precondiciones:	
Pos-condiciones:	
Tipo:	Real y expandido.
Responsabilidades:	RF6
Casos de usos relacionados:	
Interfaz I	



Interfaz II



(1) Vínculo a la página anuncios. (2) Fecha del anuncio (3) Plaza ofertada (4) Descripción de la plaza (6) Eliminar la plaza	
Curso normal de eventos para el caso de uso	
Acción del actor	Respuesta del sistema
1- La empresa va a la opción de “Anuncios” (1).	
Sección Insertar anuncio	
	1- El sistema le muestra una serie de campos (fecha (2), plaza (3), descripción (4)) que la empresa debe de llenar con los datos del anuncio.
2- La empresa llena los campos solicitados.	
3- La empresa presiona el botón “Aceptar” (5).	5- El sistema actualiza la base de datos con el nuevo anuncio creado.
Sección Eliminar anuncio	
	1- El sistema muestra una lista con todos los anuncios realizados por la empresa.
2- La empresa selecciona el anuncio que va a eliminar presionando el botón “Eliminar” (6).	3- El sistema elimina de la base de datos el anuncio solicitado.
Cursos alternos	

Caso de uso	Generar XML
Actor(es):	Desempleado
Propósito:	Generar un reporte en formato RRHH-XML.
Resumen:	El caso de uso se inicia cuando el desempleado desea generar sus datos en un formato RRHH-XML para su fácil manipulación donde la aplicación lo genera y le da la posibilidad de guardarlo en la computadora como un archivo XML.

Precondiciones:	
Pos-condiciones:	
Tipo:	Real y expandido.
Responsabilidades:	RF7
Casos de usos relacionados:	

Interfaz I

Bolsa Laboral Online

Inicio | Estadística | Autenticarse | Contáctenos | Bienvenido: Reynold

Desempleado

- Ofertas
- Avisos
- Ver datos
- Expediente
- Otras provincias
- Actualizar datos

Nombre	Reynold	Edad	23
Apellidos	Elias Hernandez	Estado	soltero
Fecha nacimiento	1985-01-12		
Sexo	masculino		
Cárnnet de Identidad	85011210421		
Correo electrónico	asd		
Teléfono	123		
Dirección	qwq weqwe	Provincia	Matanzas
Municipio	Limonar	Especialidad	diseñador
Cod. Postal	1234	Nombre de la institución	UCI
Grado	Universitario		
Título	ingeniero	Años experiencia	2
Institucion	Universidad		
localidad	matanzas		
Finalizado	2008-06-05		
Comentarios	esto es prueba		

Generar XML

(1) Genera el XML.

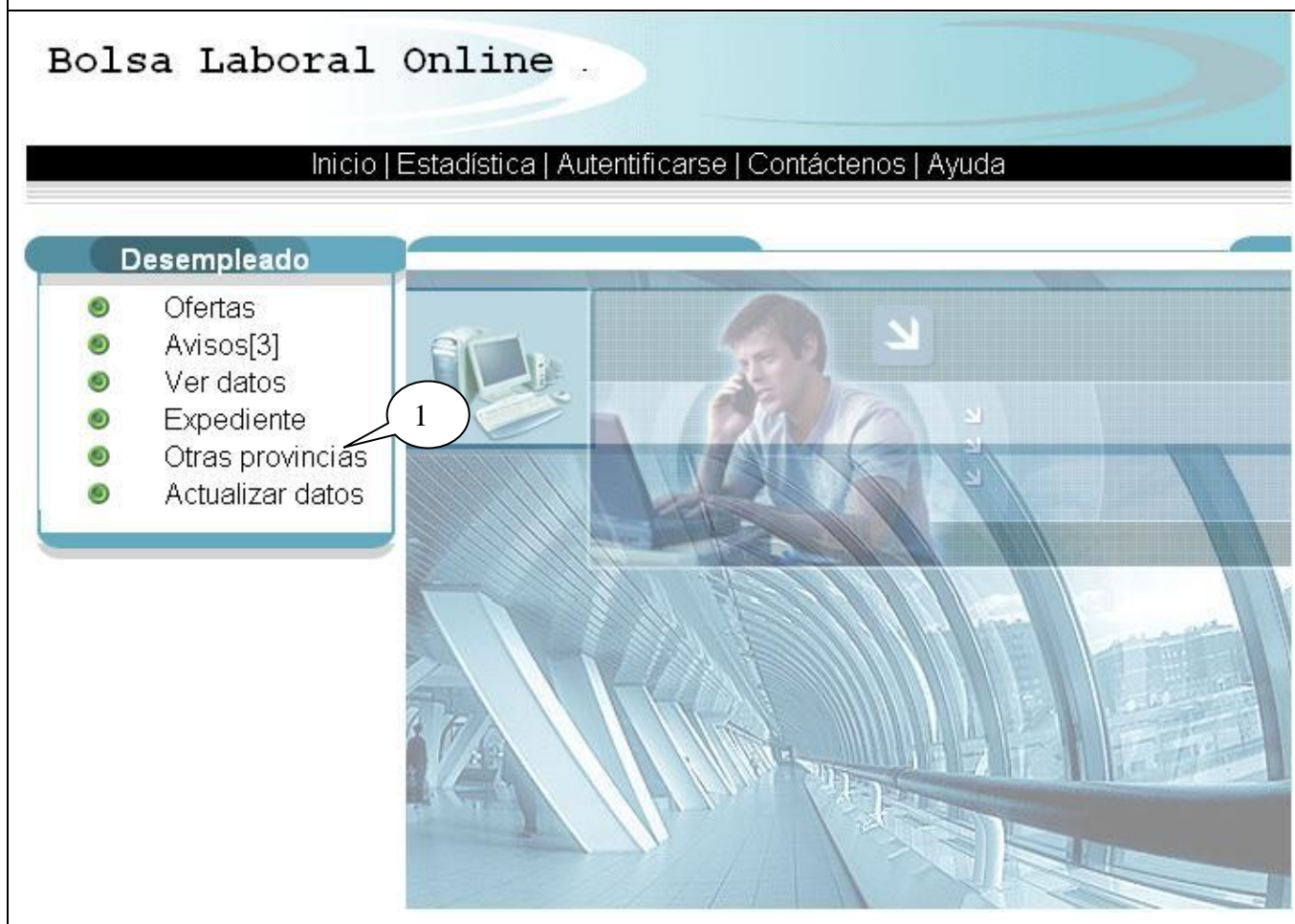
Curso normal de eventos para el caso de uso

Acción del actor	Respuesta del sistema
1- El caso de uso se inicia cuando el usuario ya mostró sus datos personales y laborales y presiona el botón "Generar XML".	2- El sistema genera y muestra los datos en un formato XML.
	4- El sistema almacena los datos del XML en el escritorio del usuario.

Cursos alternos

Caso de uso	Buscar solicitud en otra provincia
Actor(es):	Desempleado
Propósito:	Buscar ofertas de trabajos en otras provincias.
Resumen:	El caso de uso se inicia cuando el desempleado le interesa determinar saber las ofertas laborales ofertadas en otras provincias, mostrándole así el sistema una lista con estas plazas.
Precondiciones:	
Pos-condiciones:	
Tipo:	Real y expandido.
Responsabilidades:	RF9
Casos de usos relacionados:	Solicitar

Interfaz I



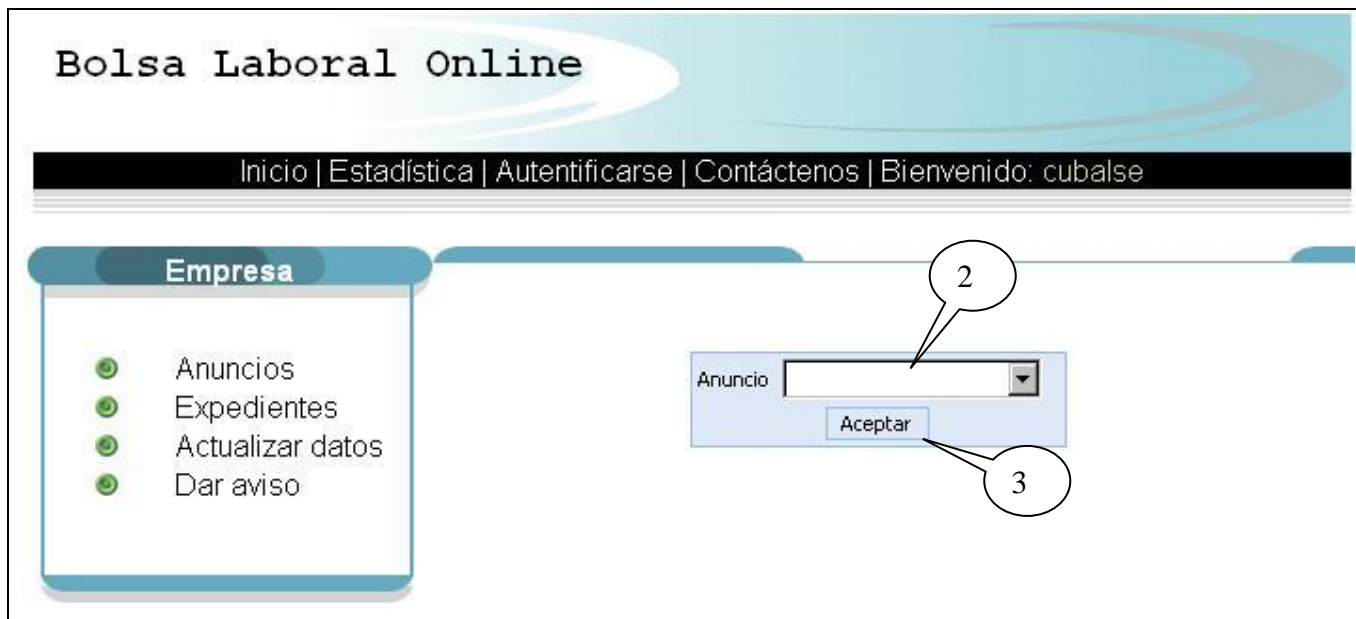
Interfaz II	
<p>(1) Vínculo a la página seleccionar provincia. (2) Seleccionar la provincia.</p>	
Curso normal de eventos para el caso de uso	
Acción del actor	Respuesta del sistema
1- El desempleado va a la opción de "Otras provincias" (1).	2- La aplicación muestra un listado con todas las provincias.
3- El usuario selecciona la provincia en la cual le interesa saber las ofertas existentes (2).	
4- El desempleado presiona el botón "Aceptar" (3).	5- El sistema muestra una lista con todas las plazas en la provincia seleccionada por el usuario.
Cursos alternos	

Caso de uso	Avisar
Actor(es):	Empresa
Propósito:	Avisar al desempleado que ha sido seleccionado para la plaza laboral.
Resumen:	El caso de uso se inicia cuando la empresa necesita avisar al desempleado que ha sido seleccionado para el puesto de trabajo.
Precondiciones:	
Pos-condiciones:	
Tipo:	Real y expandido.
Responsabilidades:	RF10
Casos de usos relacionados:	

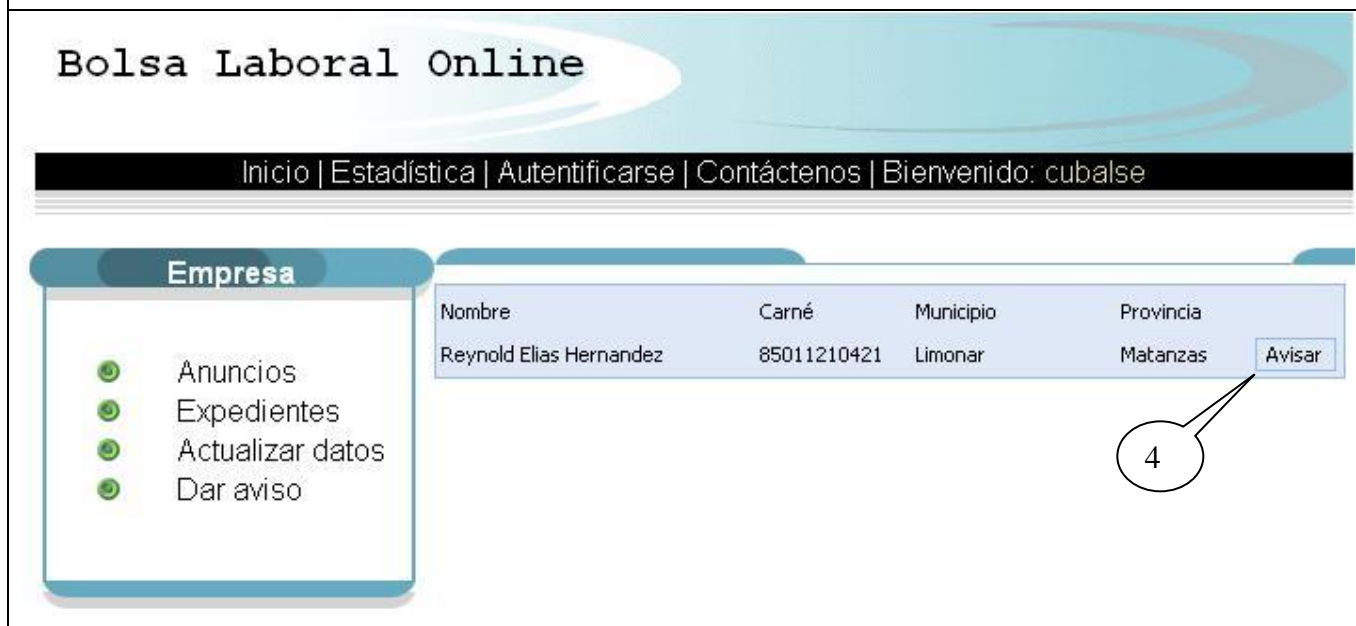
Interfaz I



Interfaz II



Interfaz III



(2) Selección del anuncio

(4) Avisar

Curso normal de eventos para el caso de uso

Acción del actor	Respuesta del sistema
1- La empresa va a la opción "Dar aviso" (1).	2- El sistema muestra una lista con todos los anuncios realizados por la empresa.
3- La empresa selecciona un anuncio (2).	

4- La empresa presiona el botón "Aceptar" (3).	5- El sistema muestra una lista con todos los desempleados que optaron por solicitar la plaza de este anuncio.
6- La empresa presiona el botón "Avisar" (4).	
Cursos alternos	

Ver más: Anexo 1.

Conclusiones

A partir de un detallado estudio acerca de la actualidad de los procesos del negocio referentes a la selección de un personal para un puesto de trabajo determinado se llegó a la conclusión de que se podía automatizar algunos de estos procesos, surgiendo así una lista de requisitos funcionales representada en un diagrama de casos de uso del sistema, además se describen de forma detallada estos casos de uso.

Capítulo III: Análisis y diseño del sistema

Introducción

Un proyecto software no consiste sólo en programar, es muy importante analizar y diseñar una posible solución. El objetivo de este capítulo es traducir los requisitos funcionales a una especificación que describe cómo implementar el sistema. El análisis consiste en obtener una visión del sistema que se preocupa de ver QUÉ hace, de modo que sólo se interesa por los requisitos funcionales. Por otro lado, el diseño es un refinamiento del análisis que tiene en cuenta los requisitos no funcionales, en definitiva CÓMO cumple el sistema sus objetivos.

3.1- Análisis

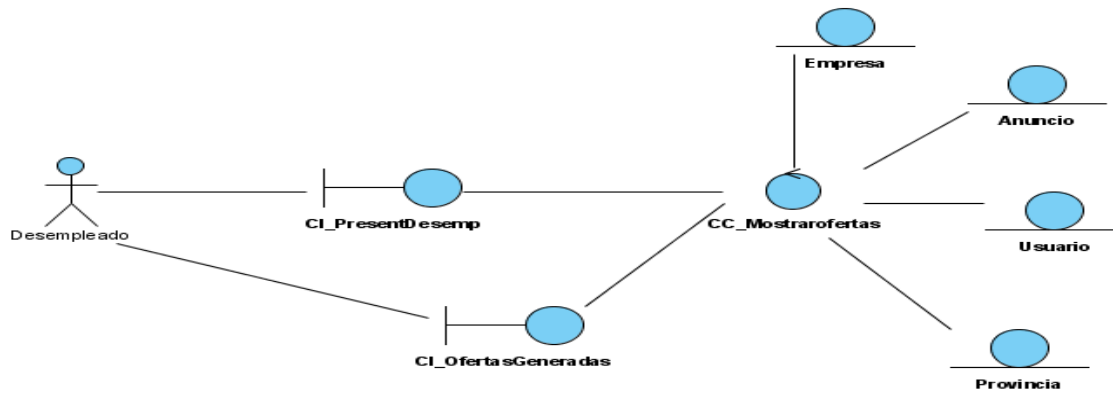
3.1.1- ¿Qué es el Análisis?

Los investigadores han identificado los problemas del análisis y sus causas y han desarrollado varias notaciones de modelado y sus correspondientes conjuntos de heurísticas para solucionarlos. Cada método de análisis tiene su punto de vista. Sin embargo, todos los métodos de análisis se relacionan por un conjunto de principios operativos.

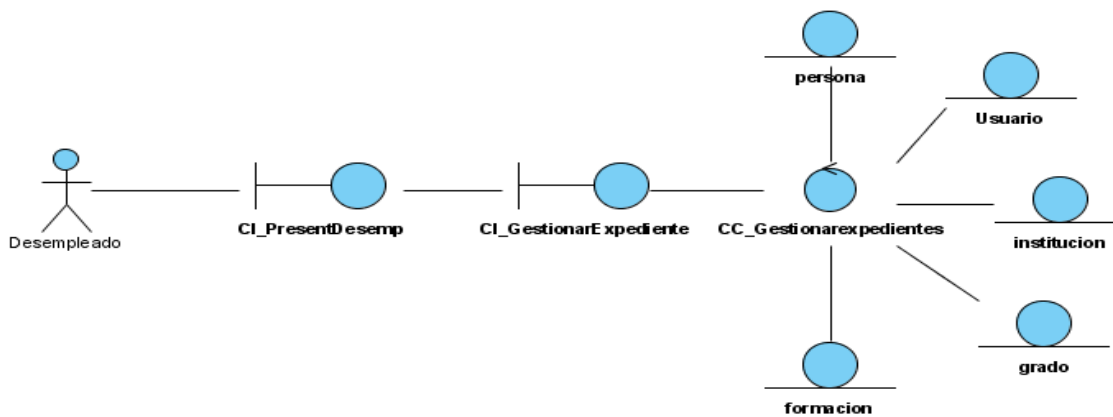
- Análisis es la investigación de requerimientos de información
- Descubrir y documentar los objetos del dominio del problema
- Determinar la forma en que los objetos del dominio del problema se relacionan entre sí
- Deben definirse las funciones que debe realizar el software.
- Debe representarse el comportamiento del software (como consecuencia de acontecimientos externos).
- Deben dividirse los modelos que representan información, función y comportamiento de manera que se descubran los detalles por capas (o jerárquicamente).
- El proceso de análisis debería ir desde la información esencial hasta el detalle de la implementación.

3.1.2- Diagramas de clases del Análisis

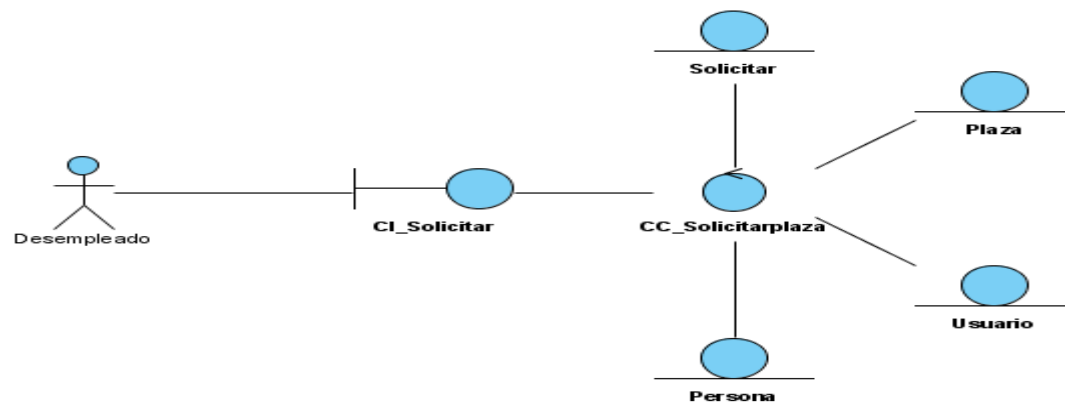
Generar ofertas



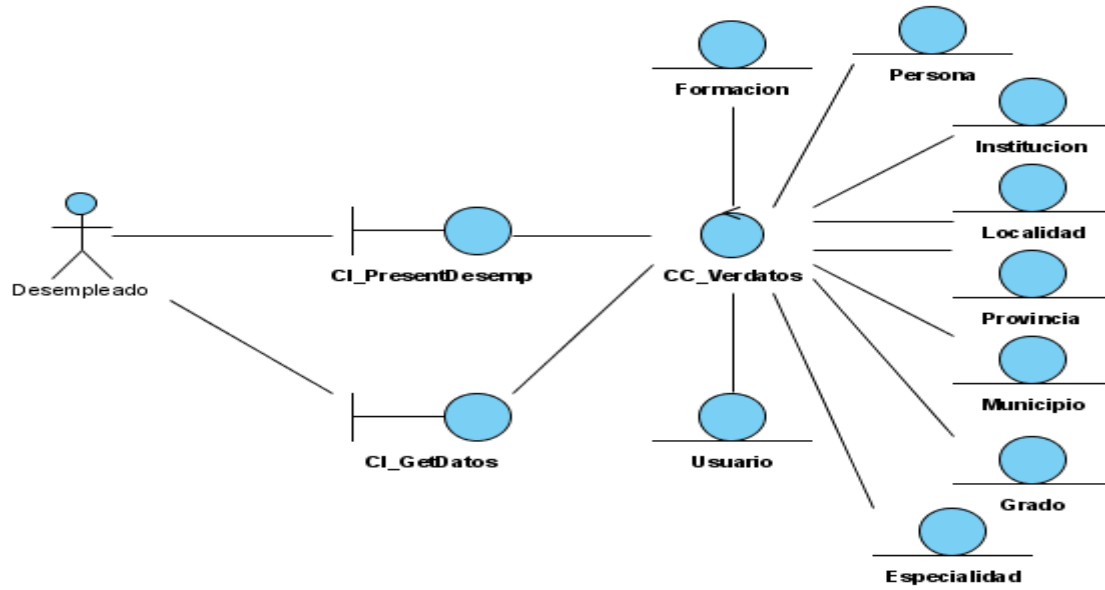
Gestionar expediente



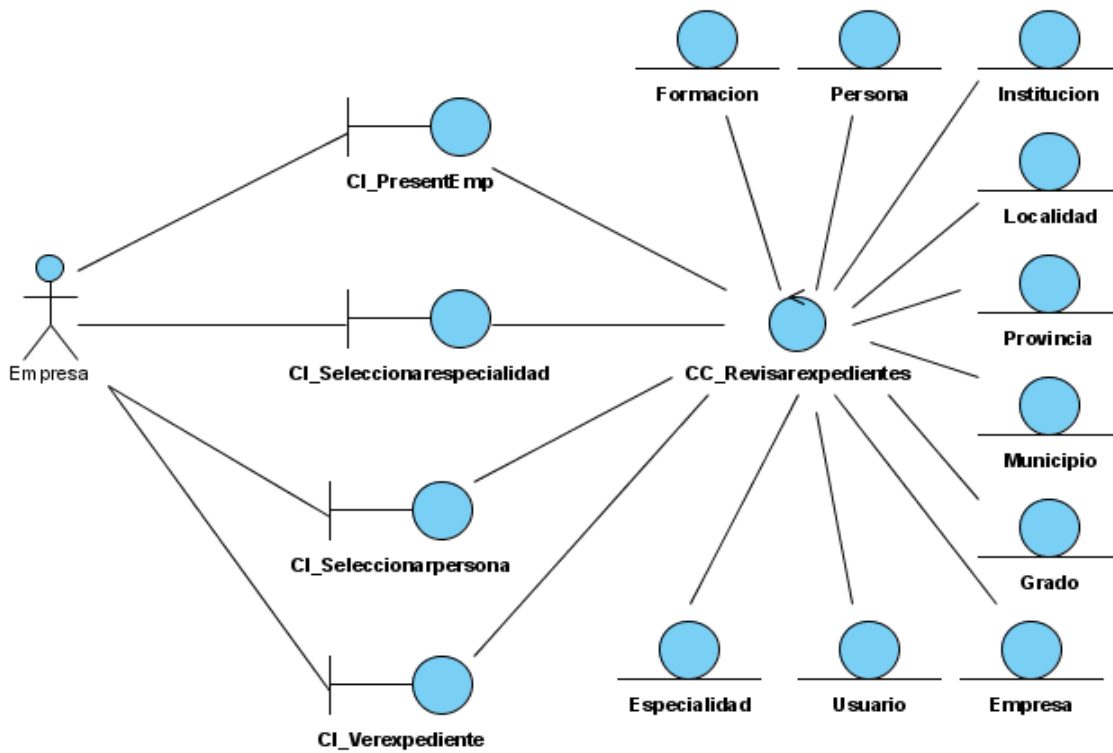
Solicitar plaza



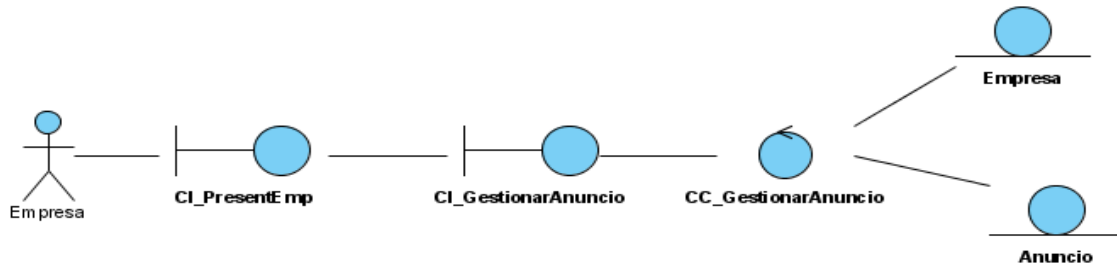
Ver datos



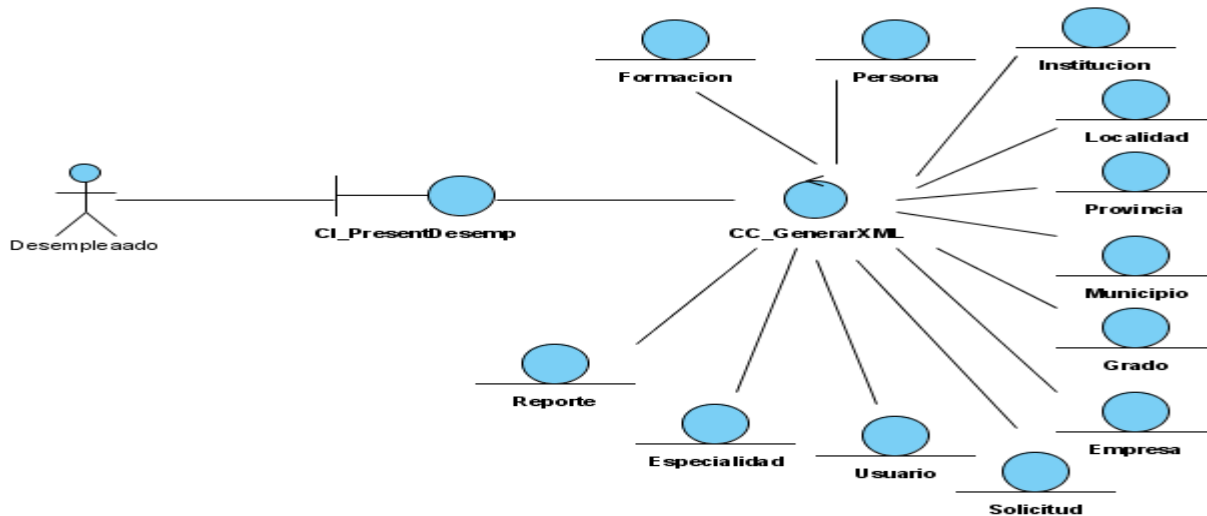
Revisar expedientes



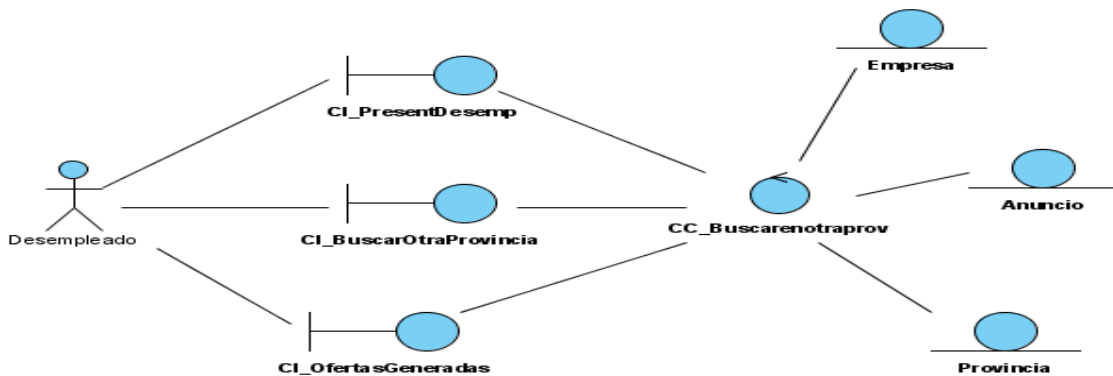
Gestionar anuncio



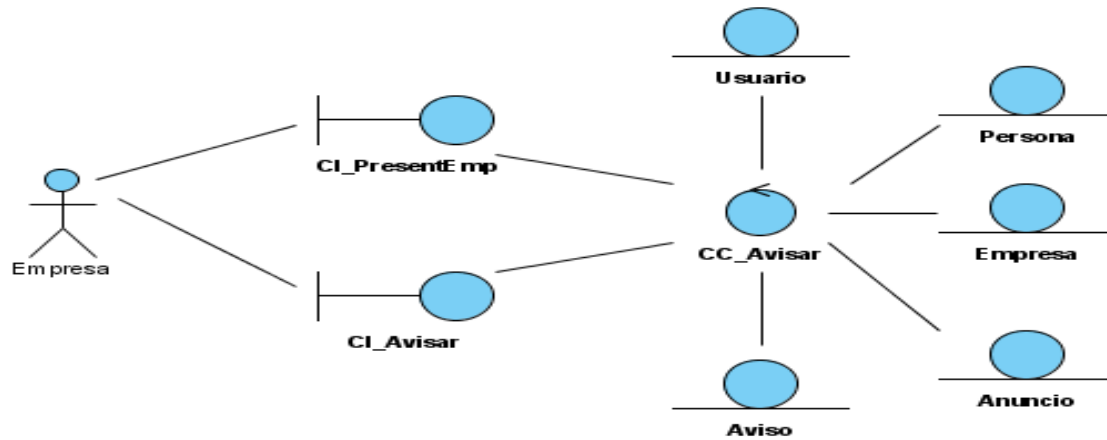
Generar XML



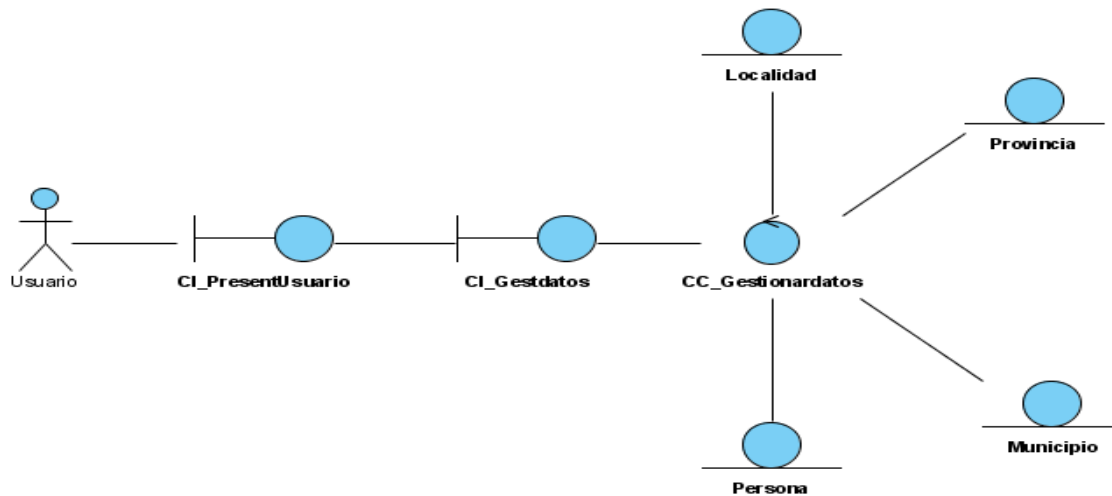
Buscar solicitud en otra provincia



Avisar



Gestionar datos del usuario



Ver más: Anexo 2.

3.2- Diseño

3.2.1- ¿Qué es el diseño?

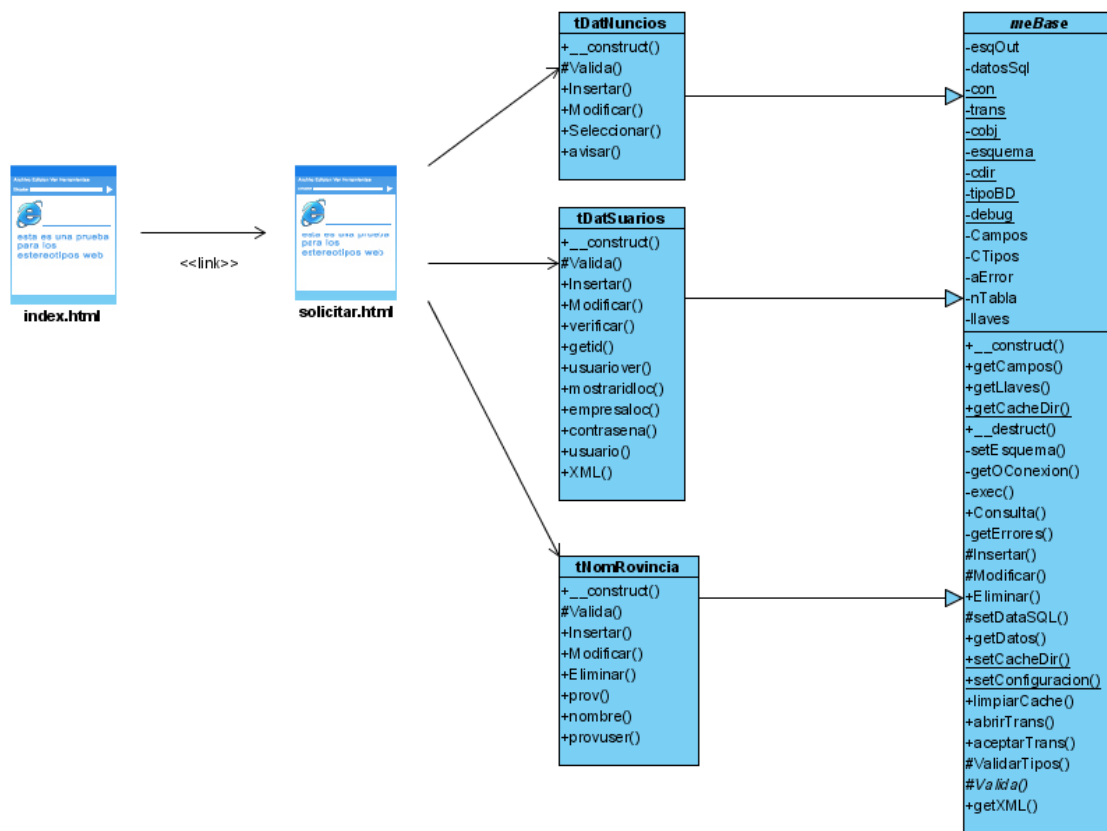
El diseño del software se encuentra en el núcleo técnico de la ingeniería del software y se aplica independientemente del modelo de diseño de software que se utilice. El diseño actúa como un puente entre las actividades de análisis que describen los requisitos o propiedades que el sistema debe tener, y las actividades de implementación que describen la plataforma de programación y condiciones

propias del lenguaje en que se crearan los componentes físicos de dicho sistema. El objetivo de la fase de diseño es la generación de una descripción sobre como sintetizar los objetos extraídos del dominio de la solución al problema, cuyo comportamiento este de acuerdo con el modelo de análisis y con el resto de los requisitos del sistema asociados a la fase de implementación de dichos objetos como componentes de software, en definitiva el principal resultado del diseño es el modelo de diseño que se esfuerza en conservar la estructura del sistema impuesta por el modelo de análisis y que sirve como esquema para la implementación.

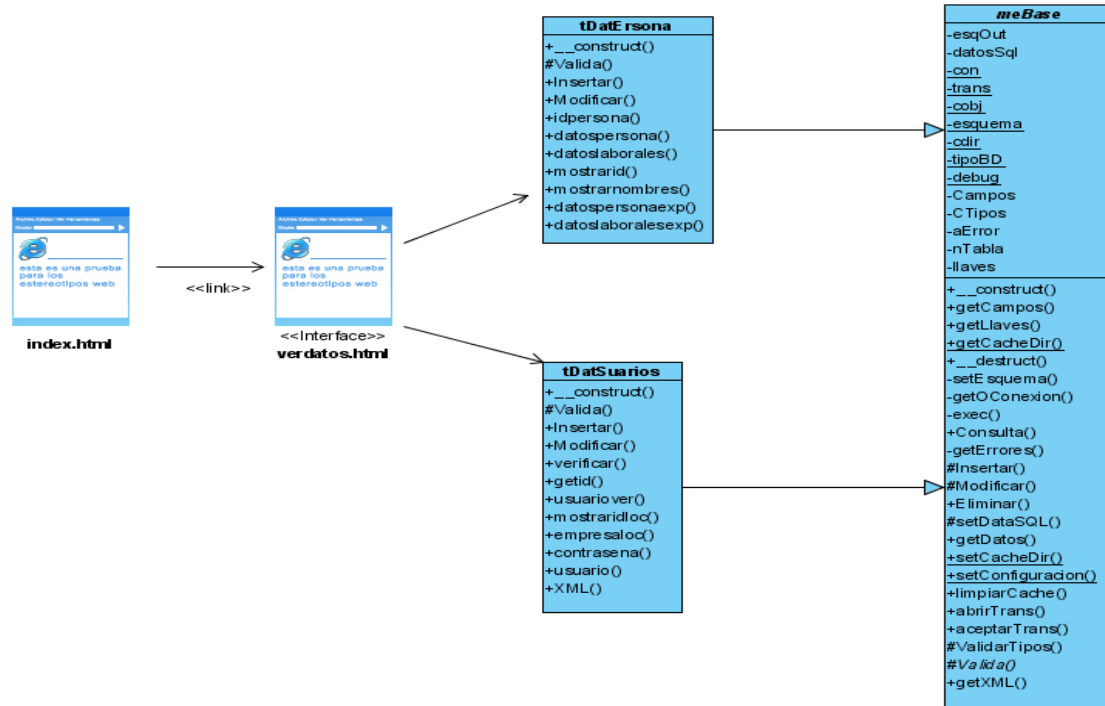
A continuación se representan diagramas de diseño de algunos casos de usos.

3.2.2- Diagramas de clases de diseño

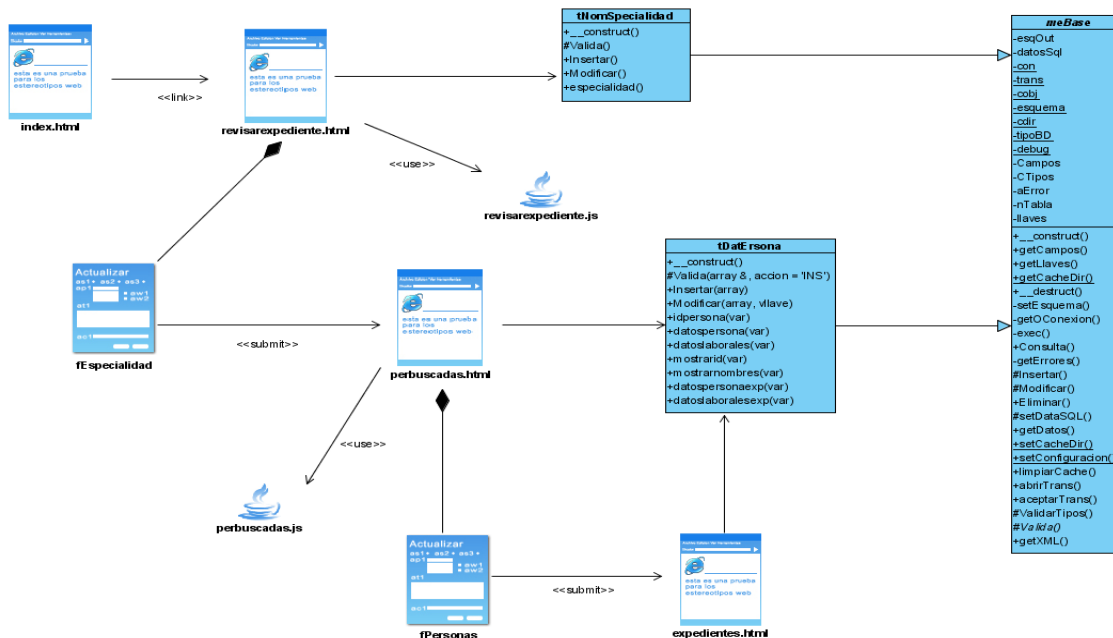
Mostrar ofertas



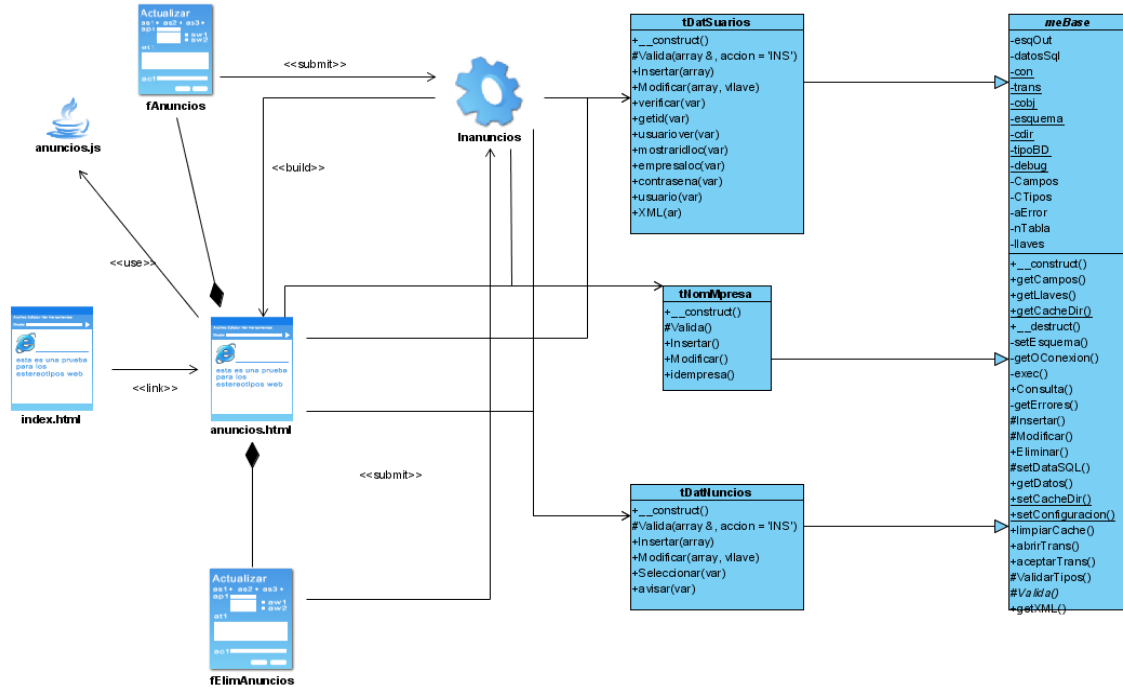
Ver datos



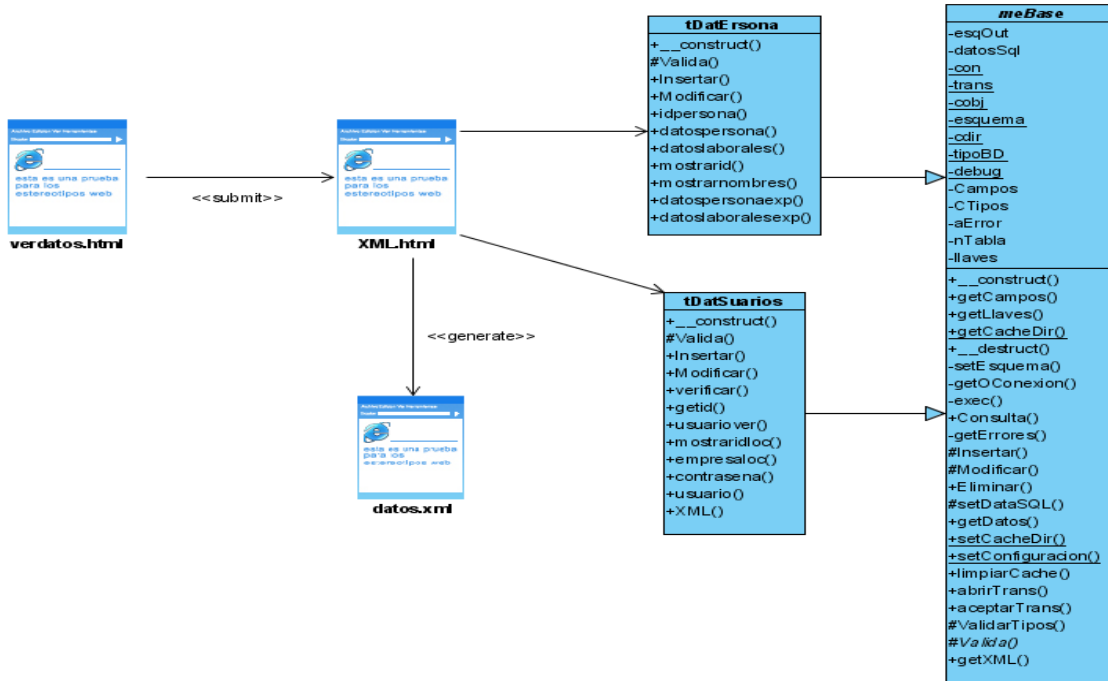
Revisar expedientes



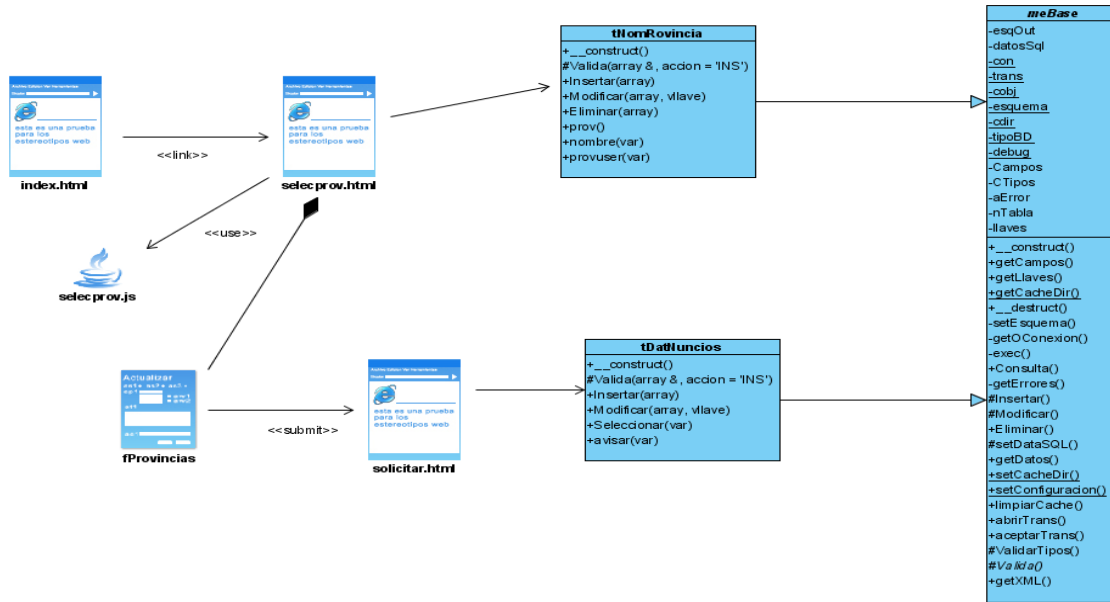
Gestionar anuncio



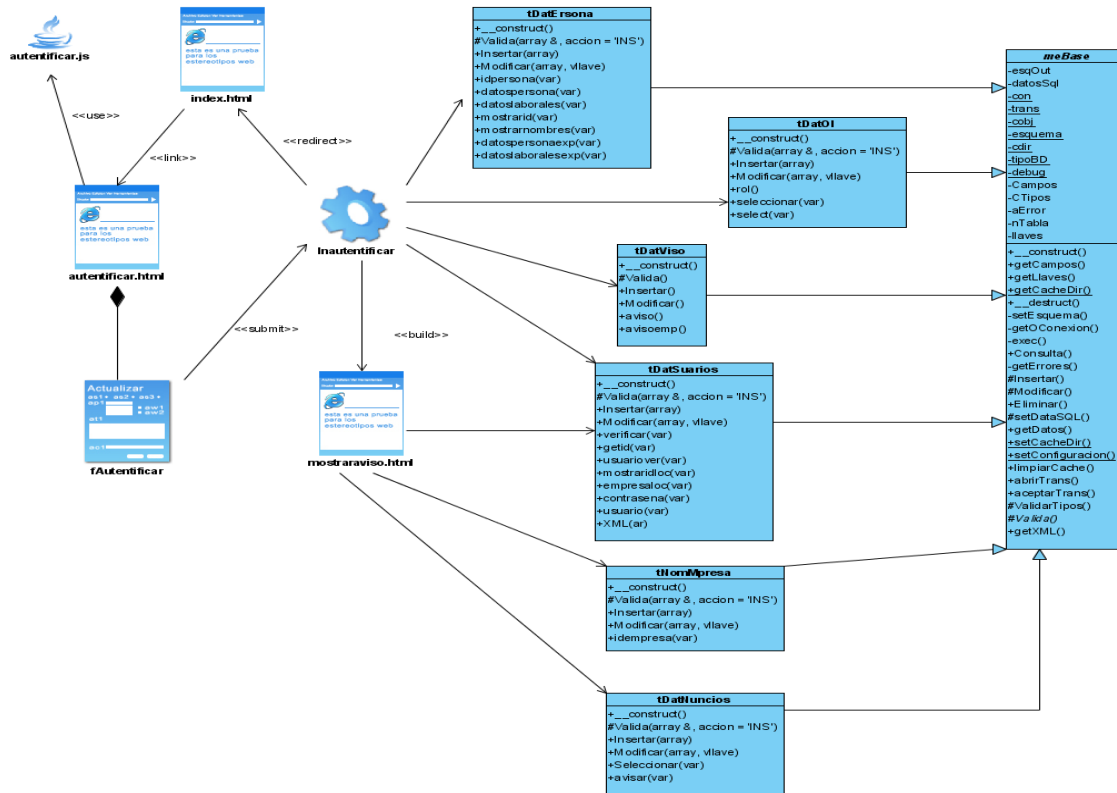
Generar XML



Buscar solicitud en otra provincia



Avisar

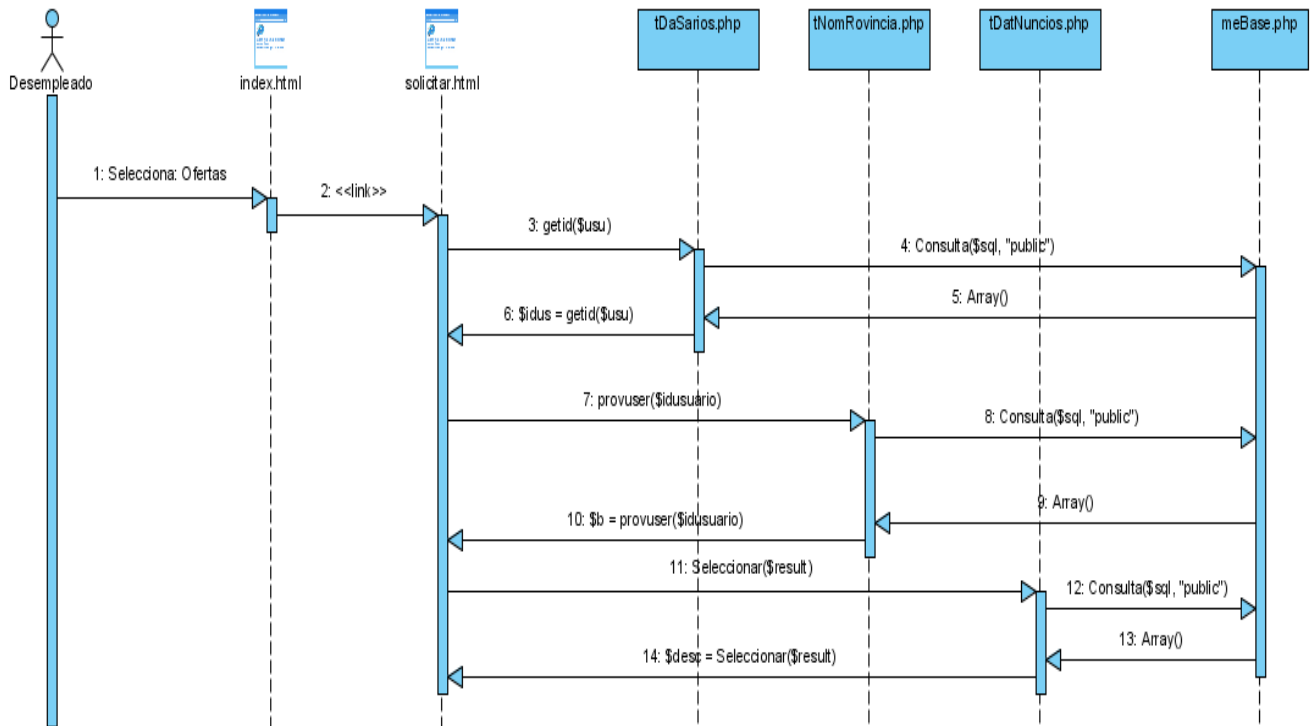


Típicamente uno examina la descripción de un caso de uso para determinar qué objetos son necesarios para la implementación del escenario. Si tienes modelada la descripción de cada caso de uso como una secuencia de varios pasos, entonces puedes "caminar sobre" esos pasos para descubrir qué objetos son necesarios para que se puedan seguir los pasos.

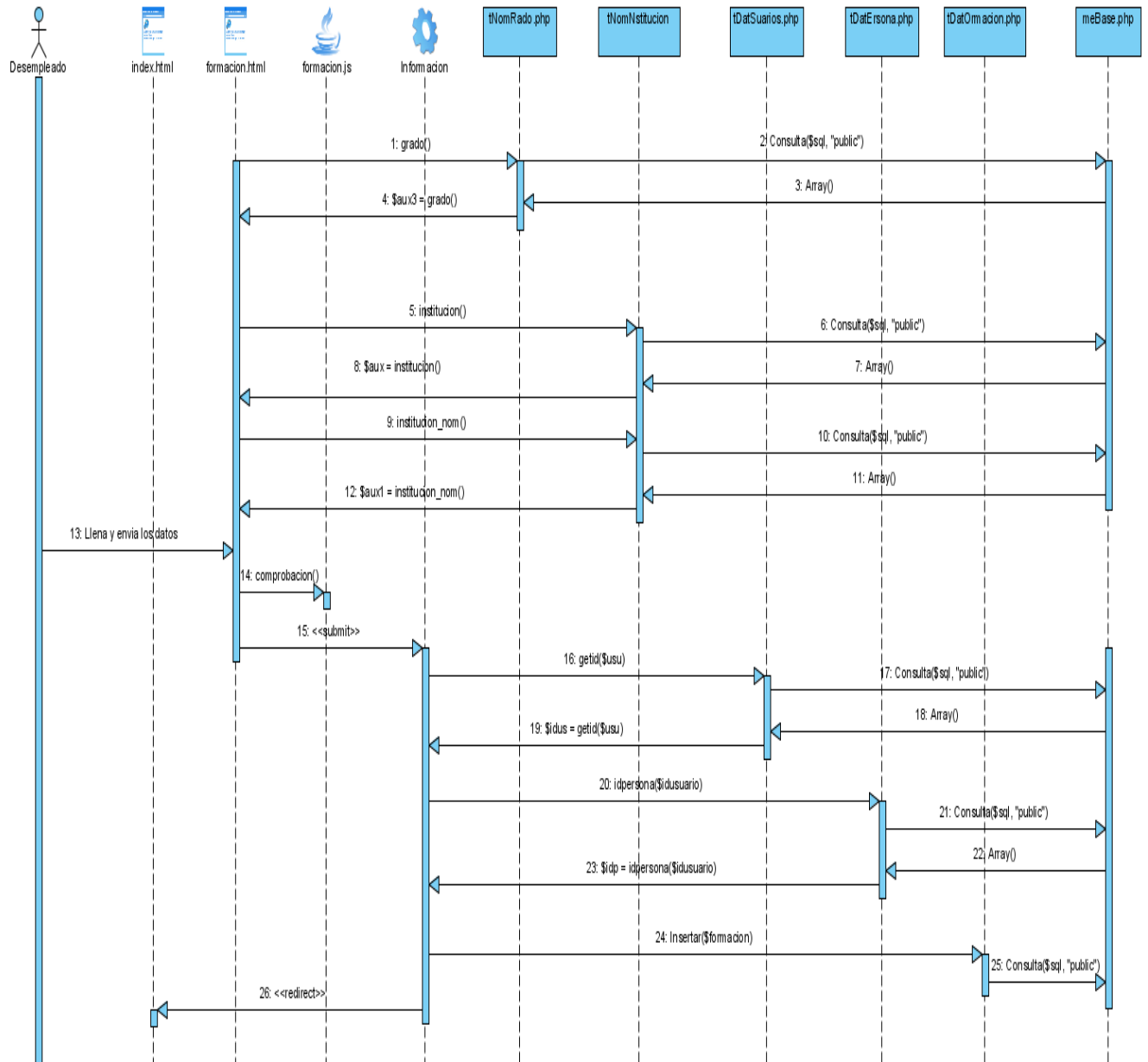
Un diagrama de secuencia muestra los objetos que intervienen en el escenario con líneas discontinuas verticales, y los mensajes pasados entre los objetos como vectores horizontales. Los mensajes se dibujan cronológicamente desde la parte superior del diagrama a la parte inferior; la distribución horizontal de los objetos es arbitraria.

En los diagramas presentados a continuación se muestra solo el flujo normal de los eventos dentro de las realizaciones de cada caso de uso.

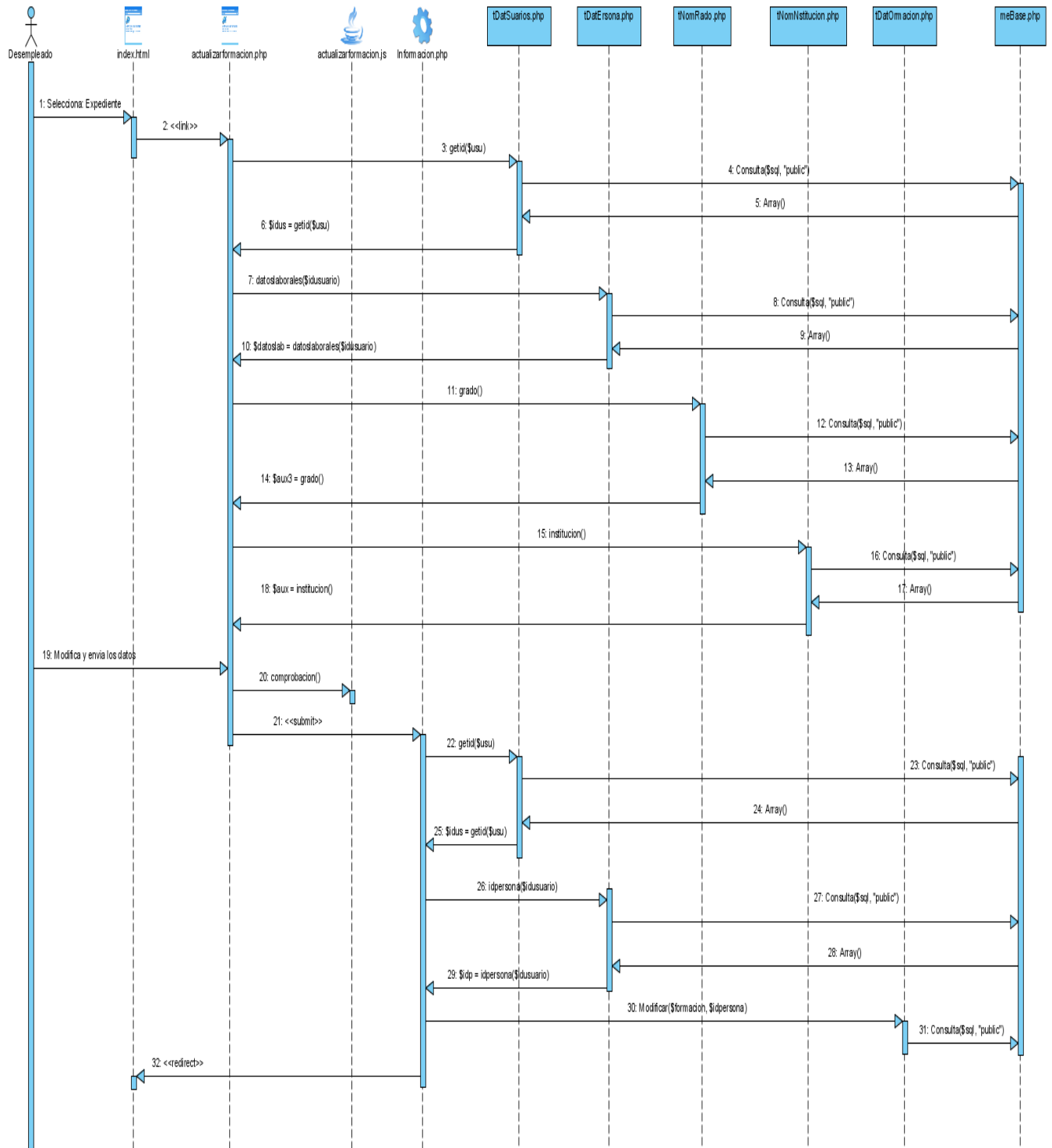
Mostrar ofertas



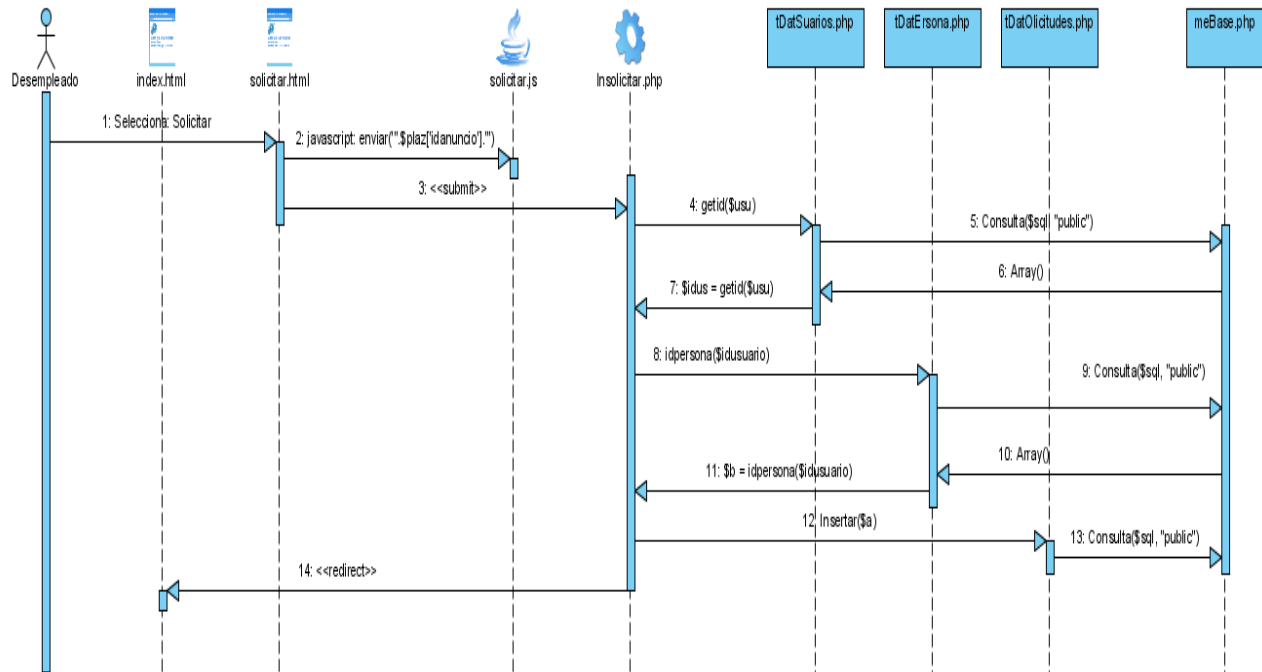
Gestionar Expediente (Insertar)



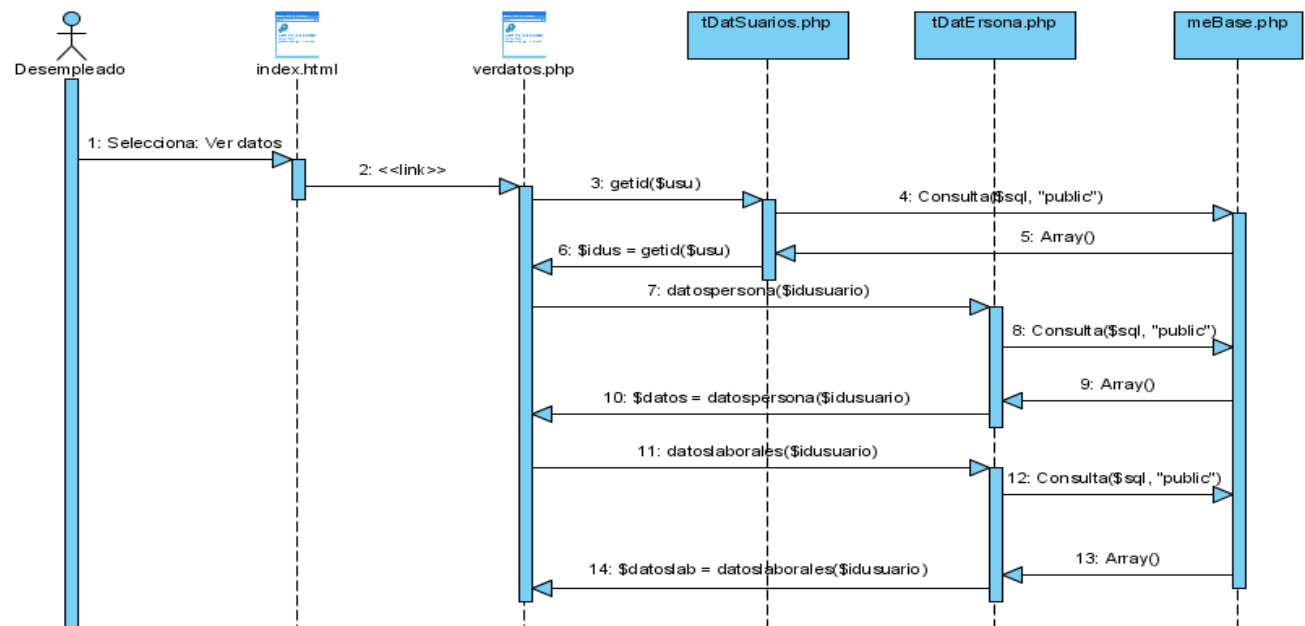
Gestionar Expediente (Modificar)



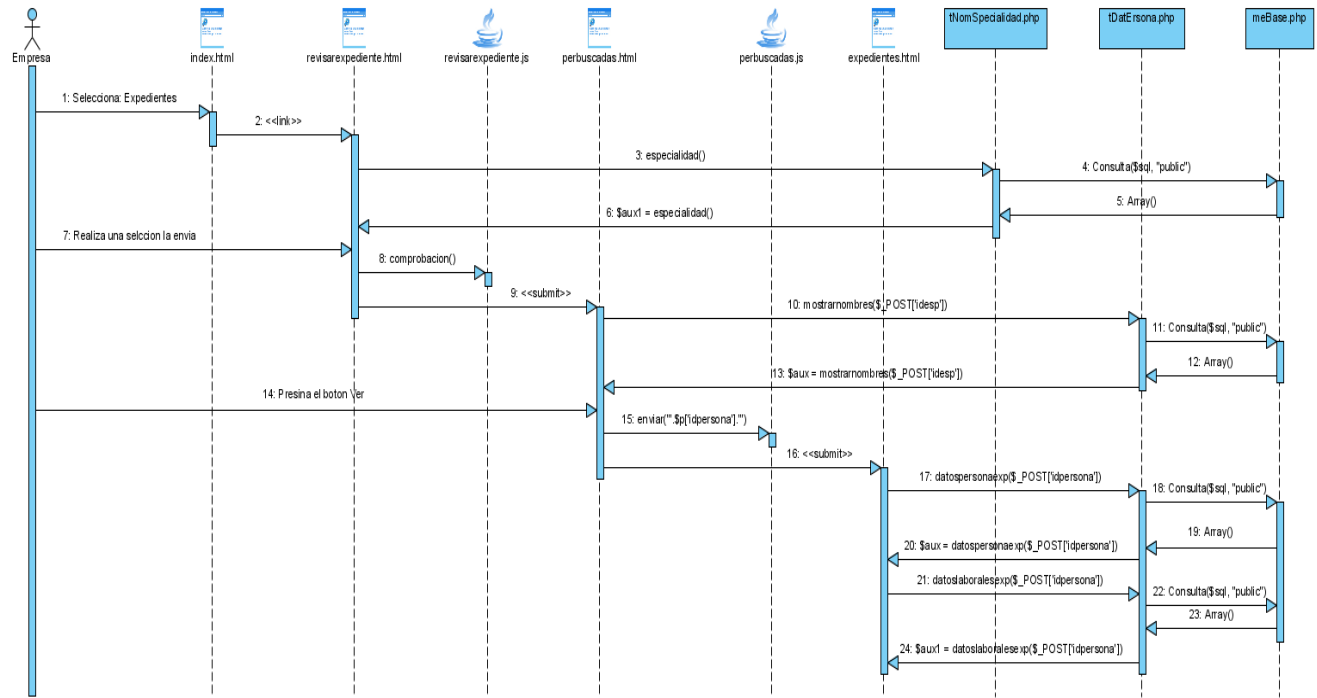
Solicitar plaza



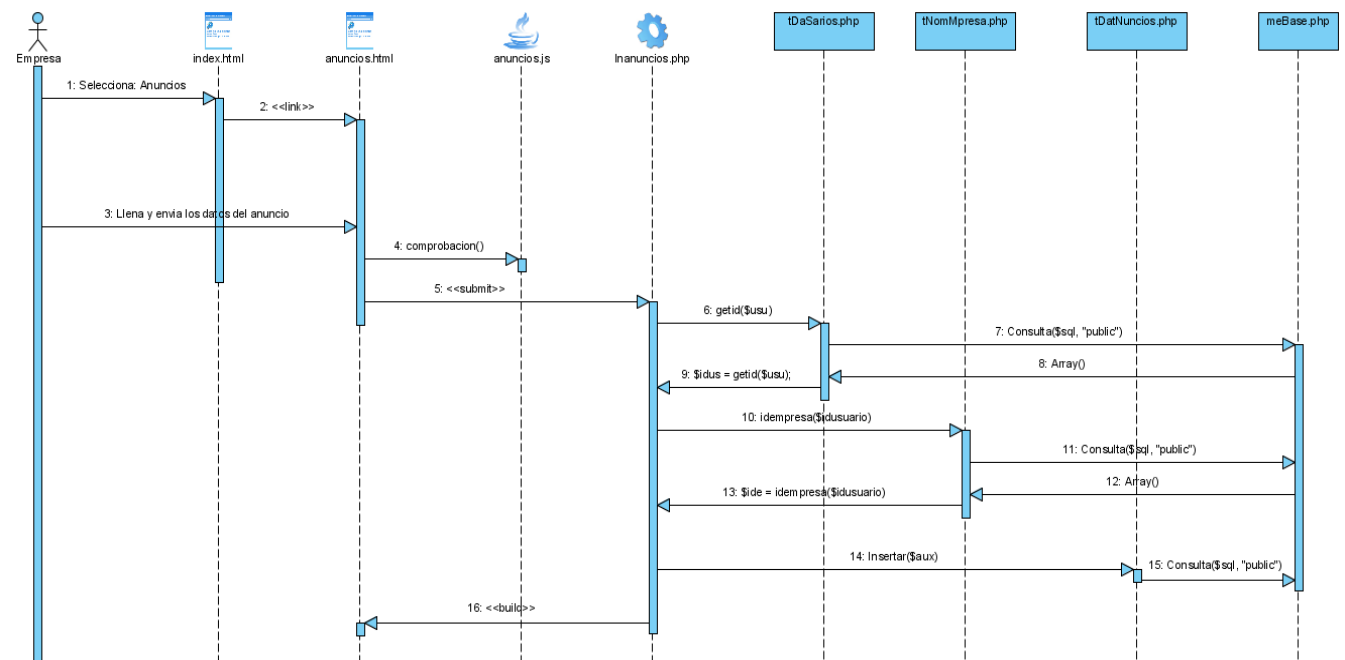
Ver datos



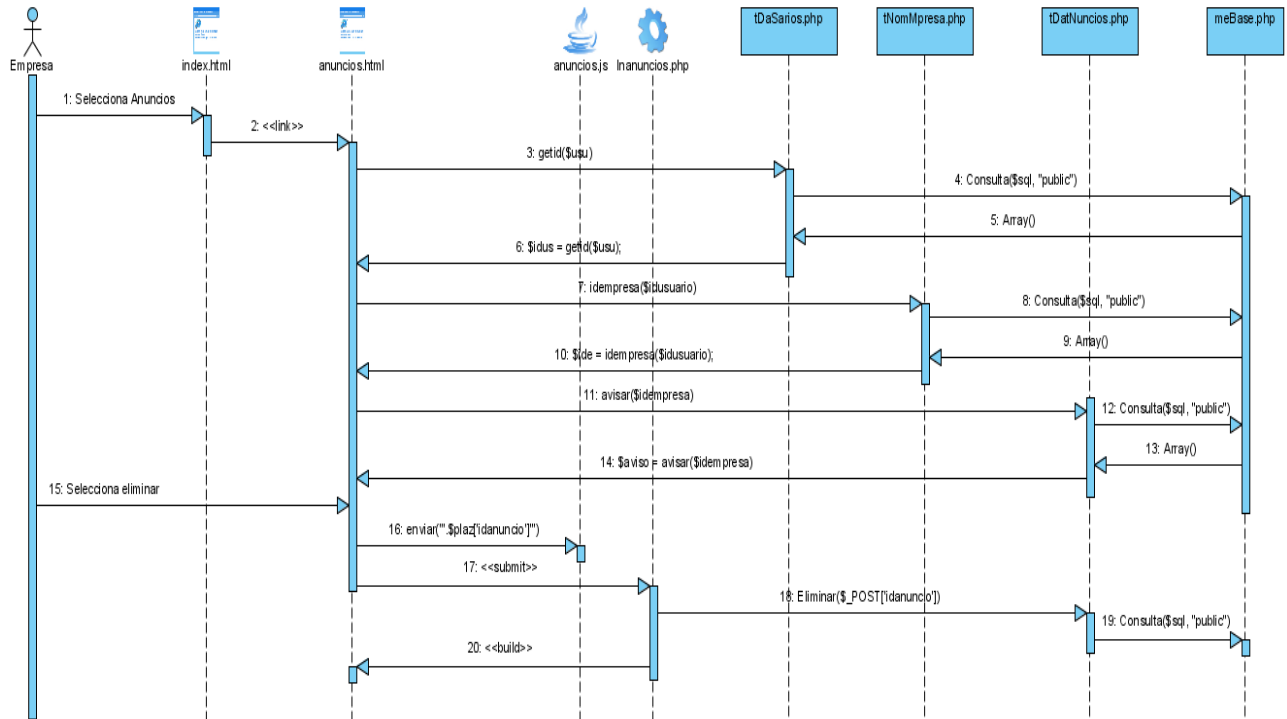
Revisar expedientes



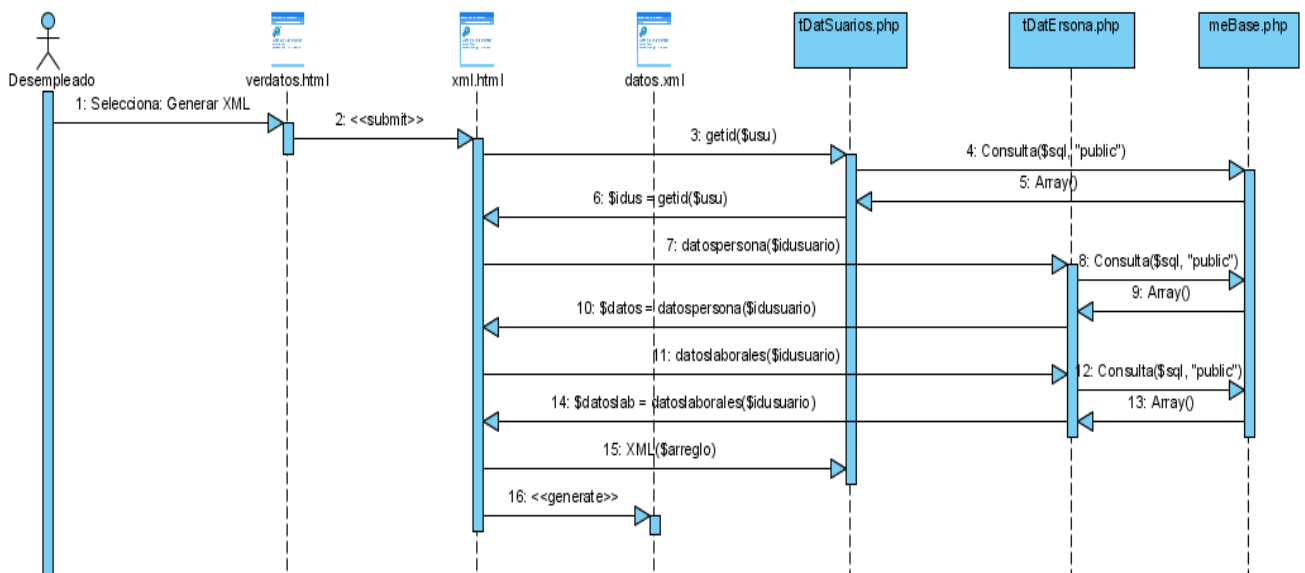
Gestionar anuncio (Insertar)



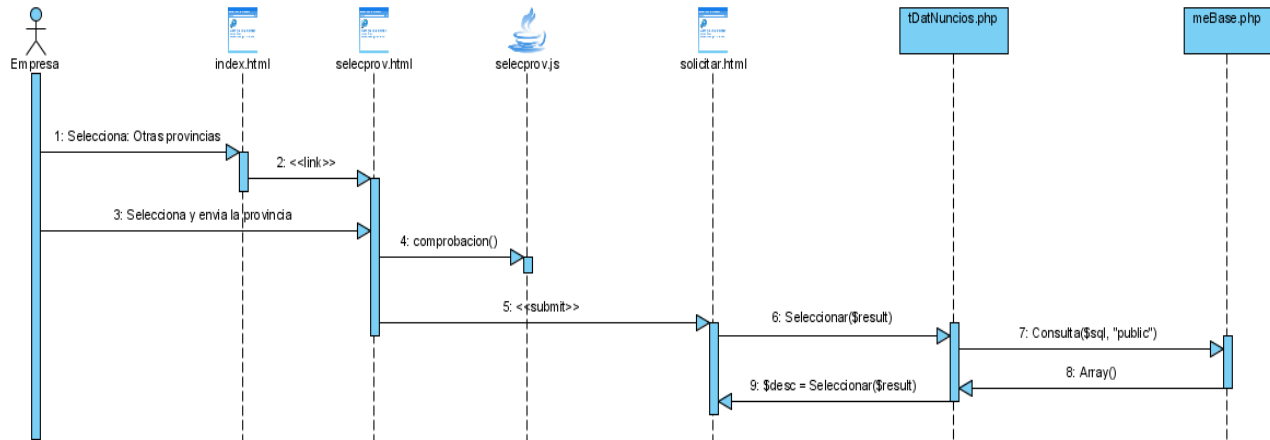
Gestionar anuncio (Eliminar)



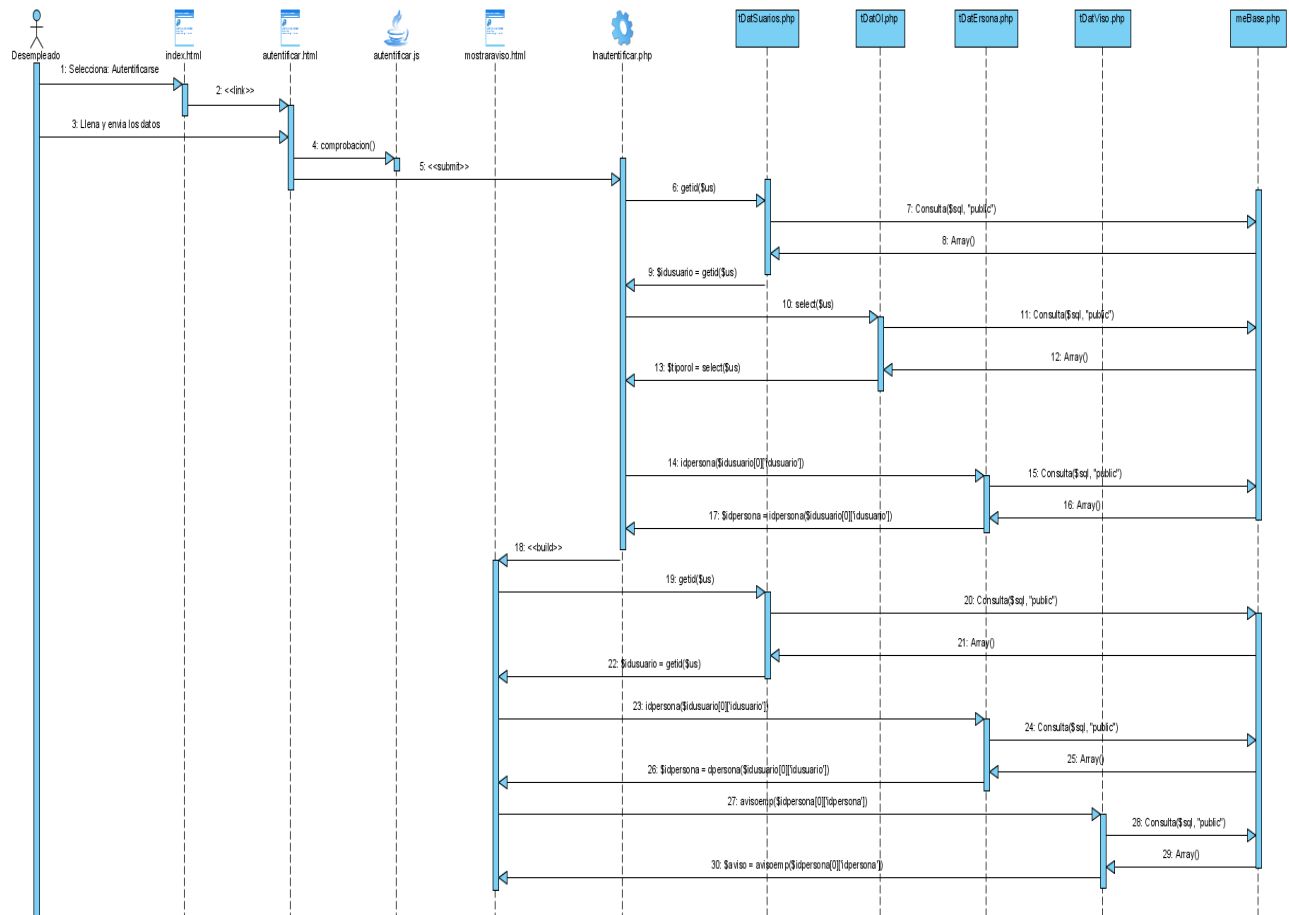
Generar XML



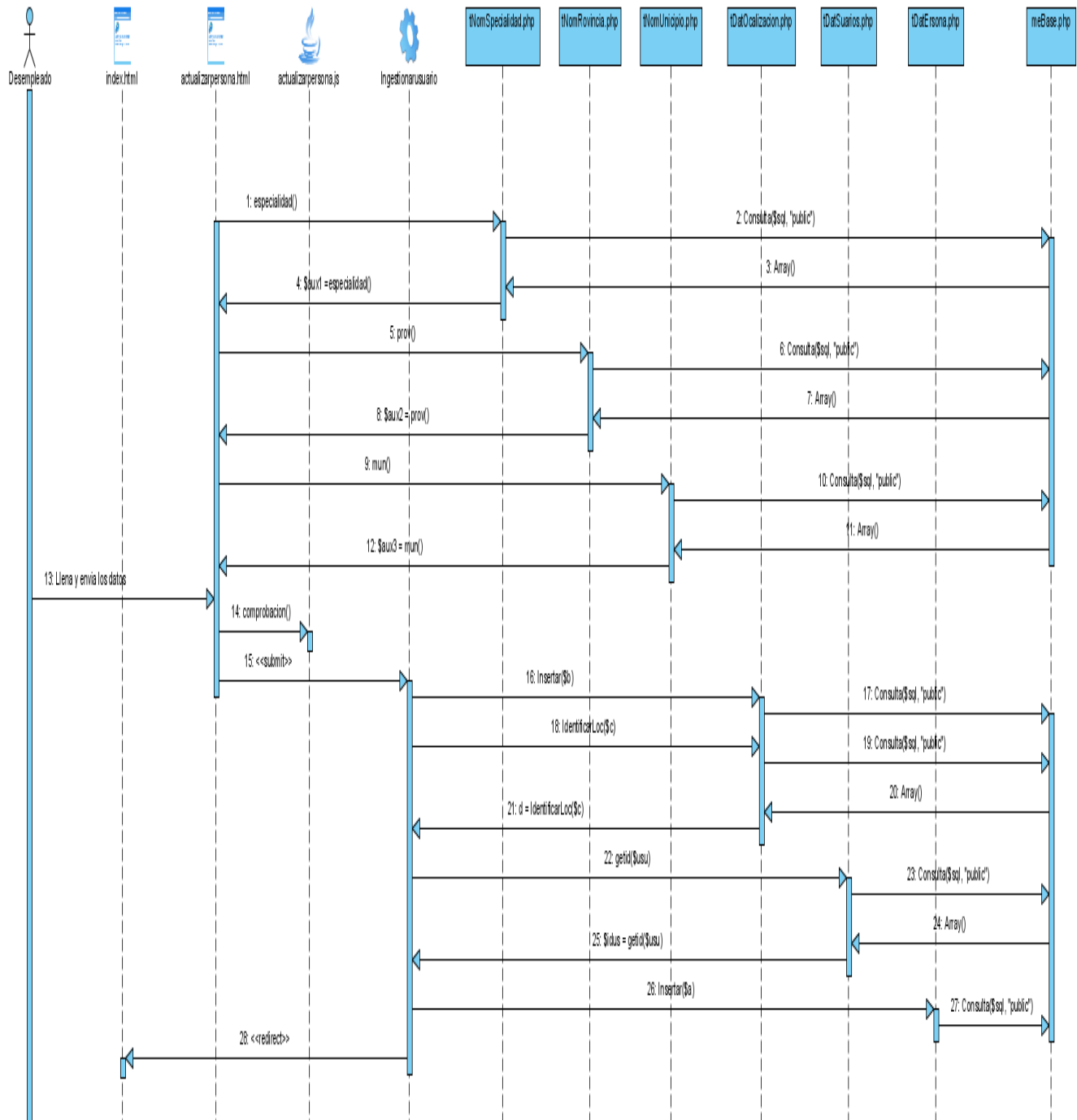
Buscar solicitud en otra provincia



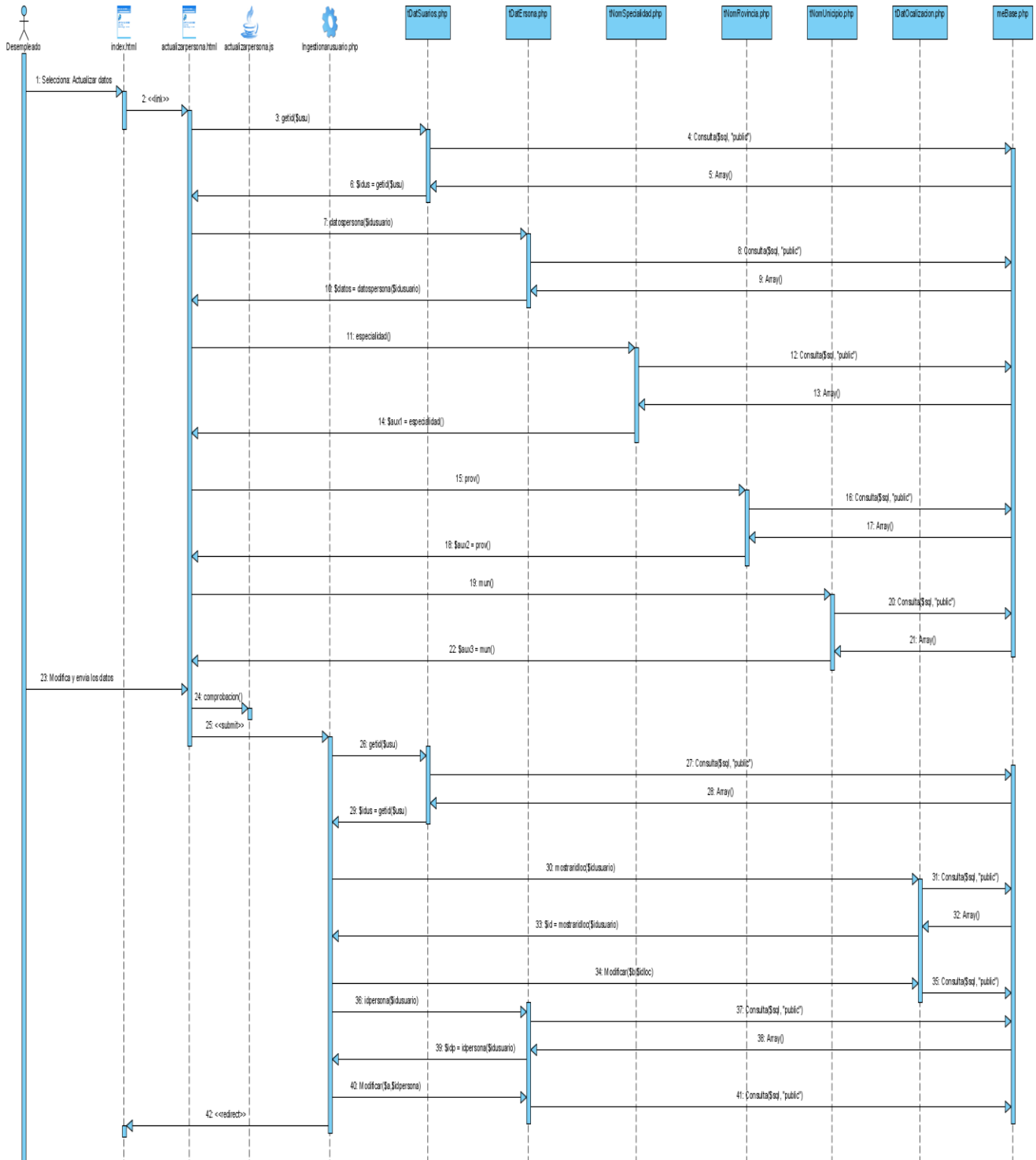
Avisar



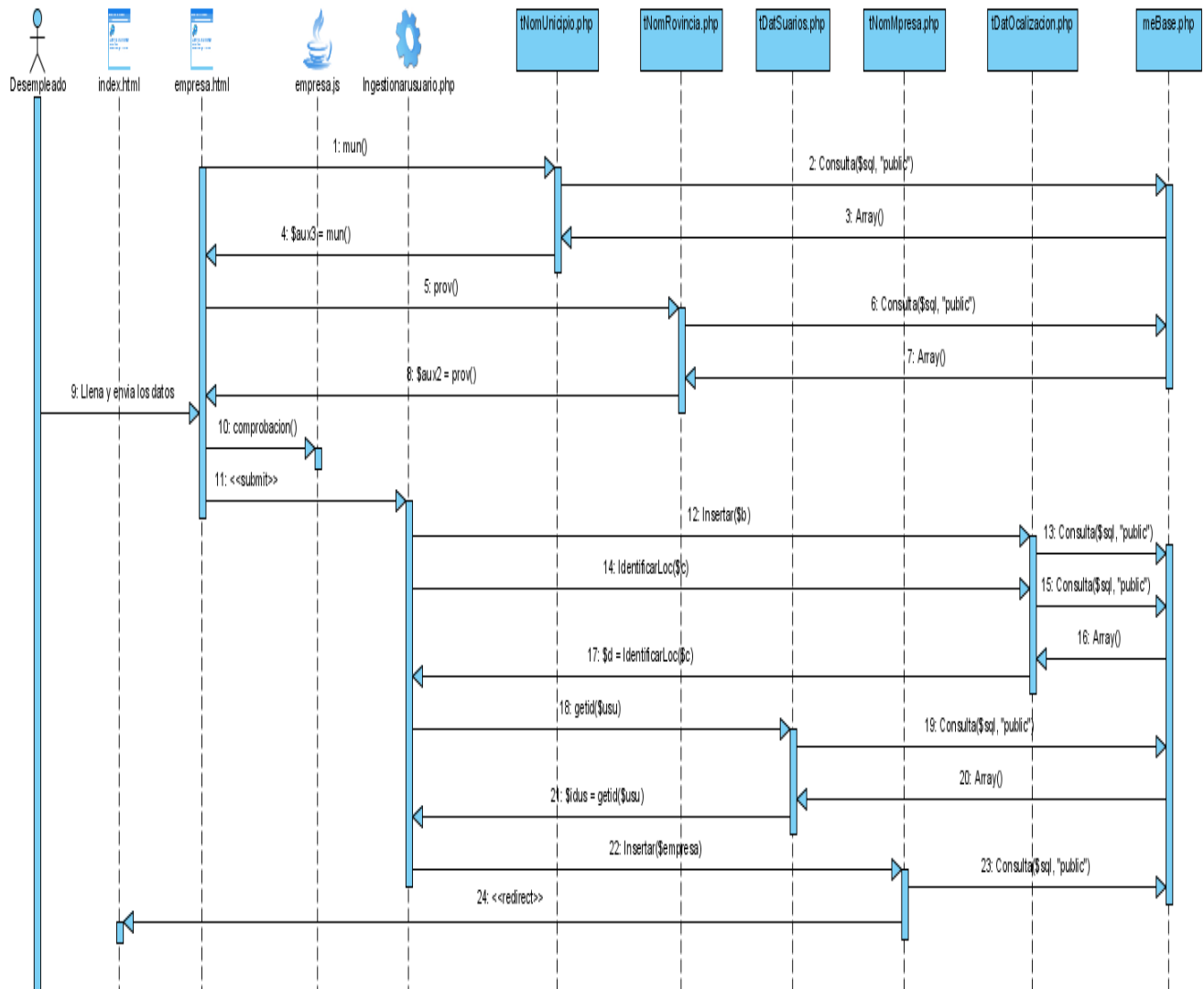
Gestionar datos del usuario (Insertar, desempleado)



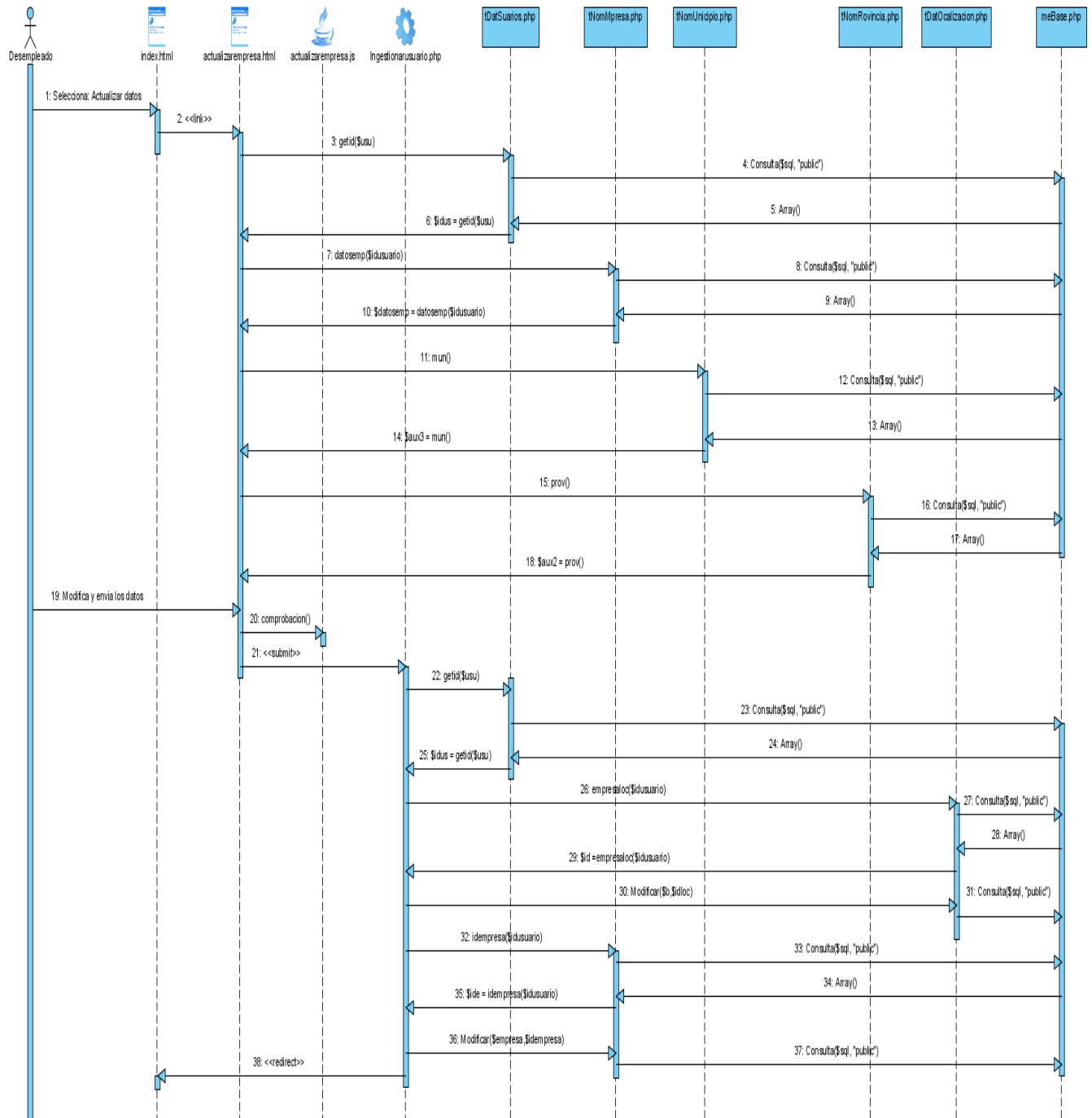
Gestionar datos del usuario (Modificar, desempleado)



Gestionar datos del usuario (Insertar, empresa)

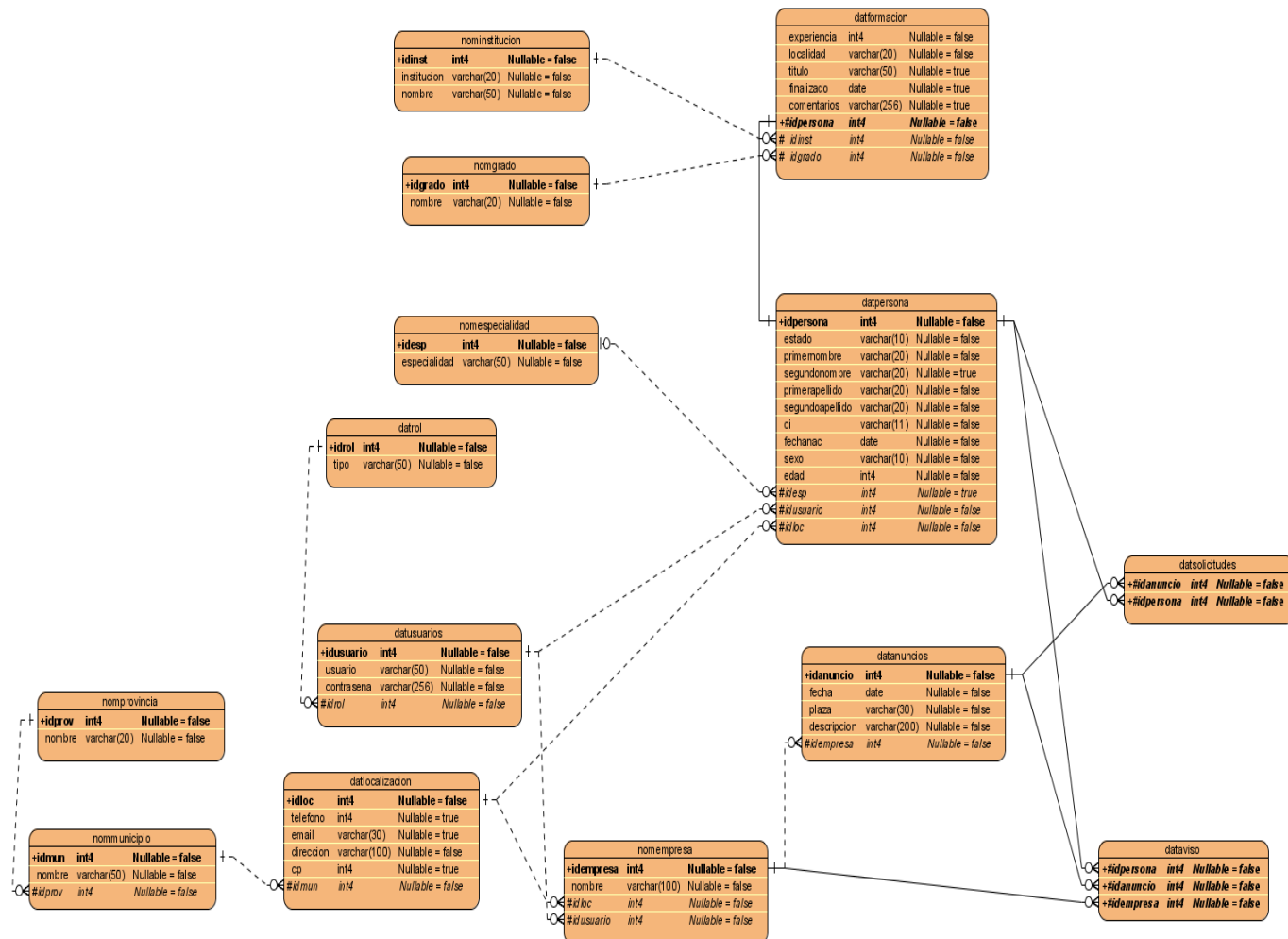


Gestionar datos del usuario (Modificar, empresa)



Ver más: Anexo 4.

3.4.2- Modelo físico de datos



3.5- Descripción de las tablas más importantes de la base de datos.

Nombre: datpersona		
Descripción: Datos personales		
Atributo	Tipo	Descripción
idpersona	integer	Identificador de la persona.
estado	varchar(10)	Estado marital de la persona.
primernombre	varchar(20)	Primer nombre de la persona.
segundonombre	varchar(20)	Segundo nombre de la persona.
primerapellido	varchar(20)	Primer apellido de la persona.
segundoapellido	varchar(20)	Segundo apellido de la persona.

ci	varchar(11)	Carné de identidad de la persona.
fechanac	date	Fecha de nacimiento de la persona.
sexo	Varchar(10)	Sexo de la persona.
idespecialidad	integer	Identificador de la especialidad.
idusuario	integer	Identificador del usuario que representa a la persona.
idloc	integer	Identificador de la localidad (dirección).

Nombre: datformacion		
Descripción: Datos de formación profesional		
Atributo	Tipo	Descripción
experiencia	integer	Años de experiencia.
localidad	varchar(20)	Lugar donde se formó.
titulo	varchar(50)	Título que obtuvo.
finalizado	date	Fecha en que se graduó.
comentarios	varchar(256)	Algún comentario extra.
idpersona	integer	Identificador de la persona.
idinst	integer	Identificador de la institución donde se graduó.
idgrado	integer	Identificador del grado de escolaridad.

Nombre: datlocalizacion		
Descripción: Dirección de la persona o empresa.		
Atributo	Tipo	Descripción
idloc	integer	Identificador de la dirección.
telefono	integer	Teléfono particular o de la empresa.
email	varchar(30)	Correo electrónico.
direccion	varchar(100)	Dirección particular o de la empresa.
cp	integer	Código postal de la localidad.
idmun	integer	Identificador del municipio al cual pertenece.

Nombre: datanuncios		
Descripción: Datos del anuncio.		
Atributo	Tipo	Descripción
idanuncio	integer	Identificador del anuncio.
fecha	date	Fecha en que se publicó el anuncio de la plaza.
plaza	varchar(30)	Plaza ofertada.
descripcion	varchar(200)	Descripción de la plaza ofertada.
idempresa	integer	Identificador de la empresa que oferta la plaza.

Nombre: datsolicitudes		
Descripción: Solicitud de una persona para optar por una plaza.		
Atributo	Tipo	Descripción
idanuncio	integer	Identificador del anuncio.
idpersona	integer	Identificador de la persona que opta por el anuncio.

Nombre: dataviso		
Descripción: Aviso a alguna persona de que fue seleccionado para la plaza.		
Atributo	Tipo	Descripción
idpersona	integer	Identificador de la persona a la que se le va a avisar.
idanuncio	integer	Identificador del anuncio.
idempresa	integer	Identificador de la empresa que publicó el anuncio.

Ver más: Anexo 5.

3.6- Descripción de clases:

Nombre: tDatErsona	
Tipo de clase: Acceso a datos	
Atributo	Tipo
-	-
Responsabilidades:	
Nombre:	__construct()
Descripción:	Método que crea un objeto de tipo persona.
Nombre:	Valida(& \$array, \$accion='INS')
Descripción:	Valida que la entrada de datos a una consulta sea la correcta.
Nombre:	Insertar(\$array)
Descripción:	Envía el arreglo entrado por parámetro a la clase meBase para insertar los elementos de este en la base de datos.
Nombre:	Modificar(\$array, \$llave)
Descripción:	Envía el arreglo y la llave primaria de la tabla donde se quieren actualizar los datos allí existentes a la clase neBase.
Nombre:	idpersona(\$var)
Descripción:	Obtiene el identificador de la persona pasándole por parámetro el identificador del usuario.
Nombre:	datospersona(\$var)
Descripción:	Obtiene los datos personales de la persona pasándole como parámetro el identificador su usuario.
Nombre:	datoslaborales(\$var)
Descripción:	Obtiene los datos que conforman el expediente laboral de una persona pasándole como parámetro el identificador su usuario.
Nombre:	mostrarid(\$var)
Descripción:	Obtiene el identificador de persona pasándole por parámetro el carné de identidad.
Nombre:	mostrarnombres(\$var)
Descripción:	Obtiene una lista con el identificador, el nombre y los apellidos de todas las personas que tienen una especialidad determinada pasada como un identificador por parámetro.

Nombre:	datospersonaexp(\$var)
Descripción:	Obtiene los datos personales de una persona pasándole como parámetro el identificador de esa persona.
Nombre:	datoslaboralesexp(\$var)
Descripción:	Obtiene los datos que conforman el expediente laboral de una persona pasándole como parámetro el identificador de esa persona.

Nombre: tDatNuncios	
Tipo de clase: Acceso a datos	
Atributo	Tipo
Responsabilidades:	
Nombre:	__construct()
Descripción:	Método que crea un objeto de tipo anuncios.
Nombre:	Valida(& \$array, \$accion='INS')
Descripción:	Valida que la entrada de datos a una consulta sea la correcta.
Nombre:	Insertar(\$array)
Descripción:	Envía el arreglo entrado por parámetro a la clase meBase para insertar los elementos de este en la base de datos.
Nombre:	Seleccionar(\$var)
Descripción:	Obtiene el anuncio, el identificador de ese anuncio, la descripción y el municipio donde se encuentra la empresa que publica ese anuncio, pasándole como parámetro el identificado de la provincia.
Nombre:	avisar(\$var)
Descripción:	Obtiene la plaza y el identificador del anuncio pasándole como parámetro el identificador de una empresa.

Nombre: tDatOcalizacion	
Tipo de clase: Acceso a datos	
Atributo	Tipo
Responsabilidades:	

Nombre:	__construct()
Descripción:	Método que crea un objeto de tipo localización.
Nombre:	Valida(& \$array, \$accion='INS')
Descripción:	Valida que la entrada de datos a una consulta sea la correcta.
Nombre:	Insertar(\$array)
Descripción:	Envía el arreglo entrado por parámetro a la clase meBase para insertar los elementos de este en la base de datos.
Nombre:	IdentificarLoc(\$var)
Descripción:	Selecciona el identificador de la localidad pasándole como parámetro la dirección de la localidad.

Nombre: tDatOlicitudes	
Tipo de clase: Acceso a datos	
Atributo	Tipo
Responsabilidades:	
Nombre:	__construct()
Descripción:	Método que crea un objeto de tipo solicitudes.
Nombre:	Valida(& \$array, \$accion='INS')
Descripción:	Valida que la entrada de datos a una consulta sea la correcta.
Nombre:	Insertar(\$array)
Descripción:	Envía el arreglo entrado por parámetro a la clase meBase para insertar los elementos de este en la base de datos.

Nombre: tDatOrmacion	
Tipo de clase: Acceso a datos	
Atributo	Tipo
Responsabilidades:	
Nombre:	__construct()
Descripción:	Método que crea un objeto de tipo formación.
Nombre:	Valida(& \$array, \$accion='INS')

Descripción:	Valida que la entrada de datos a una consulta sea la correcta.
Nombre:	Insertar(\$array)
Descripción:	Envía el arreglo entrado por parámetro a la clase meBase para insertar los elementos de este en la base de datos.

Nombre: tDatViso	
Tipo de clase: Acceso a datos	
Atributo	Tipo
Responsabilidades:	
Nombre:	__construct()
Descripción:	Método que crea un objeto de tipo aviso.
Nombre:	Valida(& \$array, \$accion='INS')
Descripción:	Valida que la entrada de datos a una consulta sea la correcta.
Nombre:	Insertar(\$array)
Descripción:	Envía el arreglo entrado por parámetro a la clase meBase para insertar los elementos de este en la base de datos.
Nombre:	aviso(\$var)
Descripción:	Obtiene el identificador del anuncio pasándole como parámetro el identificador de la empresa.
Nombre:	avisoemp(\$var)
Descripción:	Obtiene el identificador del anuncio, la plaza, la empresa, el municipio y la provincia de esa empresa, pasándole como parámetro el identificador de una persona.

Ver más: Anexo 6.

Conclusiones

En este capítulo, se profundizó en la modelación y la realización de los diagramas de clases de análisis y de diseño analizándose el dominio del problema, así como los modelos de análisis para crear especificaciones adicionales que nutren con detalles una futura implementación.

CAPITULO IV: IMPLEMENTACIÓN Y PRUEBA

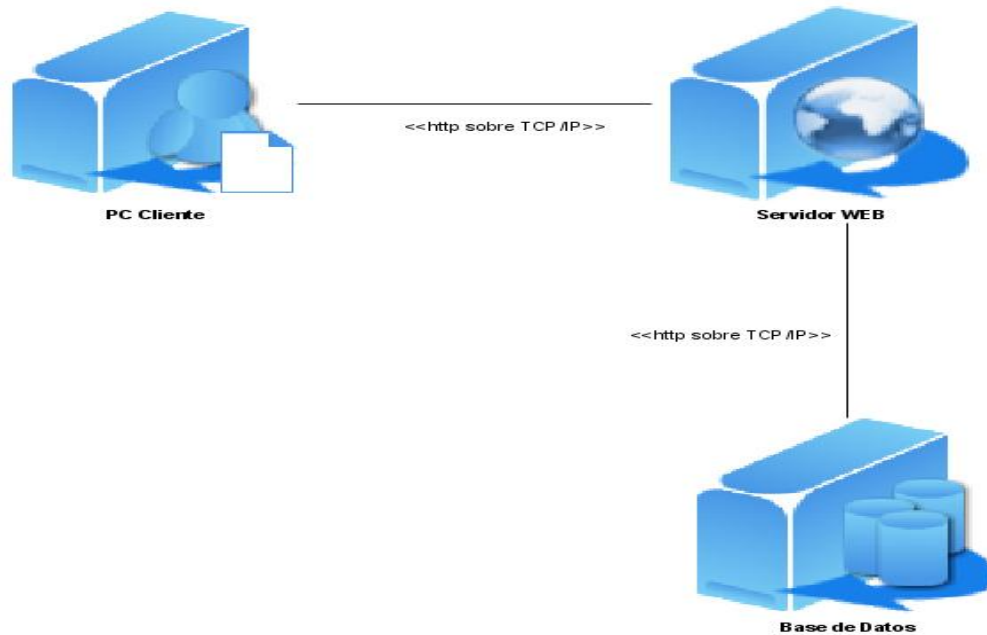
Introducción:

En la implementación se empezó con el resultado del diseño e implementamos el sistema en términos de componentes, es decir, ficheros de código fuente, scripts, ficheros de código binario, ejecutables y similares.

En este capítulo se aborda la parte de implementación donde se muestra las dependencias entre las partes de código del sistema (diagrama de componentes) o la estructura del sistema en ejecución (diagrama de despliegue): los diagramas de componentes se utilizan para modelar la vista de implementación estática de un sistema, mientras que los diagramas de despliegue se utilizan para modelar la vista de despliegue estática. Un componente no es más que una parte física reemplazable de un sistema que empaqueta su implementación y es conforme a un conjunto de interfaces a las que proporciona su realización. Algunos componentes tienen identidad y pueden poseer entidades físicas, que incluyen objetos en tiempo de ejecución, documentos, bases de datos, etcétera. También se le aplicará y documentará unas series de pruebas al producto para garantizar su calidad final.

4.1- Diagrama de despliegue

El diagrama de despliegue muestra la asignación de artefactos concretos de software, como ejecutables, a nodos computacionales, o sea, el despliegue de los mismos en la arquitectura física y la comunicación entre los elementos de los nodos. Un nodo es un recurso de ejecución, una unidad de computación de algún tipo, desde un sensor a un mainframe.



4.2- Diagrama de componentes:

Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones.

Los componentes representan todos los tipos de elementos software que entran en la fabricación de aplicaciones informáticas. Pueden ser simples archivos, paquetes de Ada, bibliotecas cargadas dinámicamente, etc. Las relaciones de dependencia se utilizan en los diagramas de componentes para indicar que un componente utiliza los servicios ofrecidos por otro componente. El diagrama de componente hace parte de la vista física de un sistema, la cual modela la estructura de implementación de la aplicación por sí misma, su organización en componentes y su despliegue en nodos de ejecución.

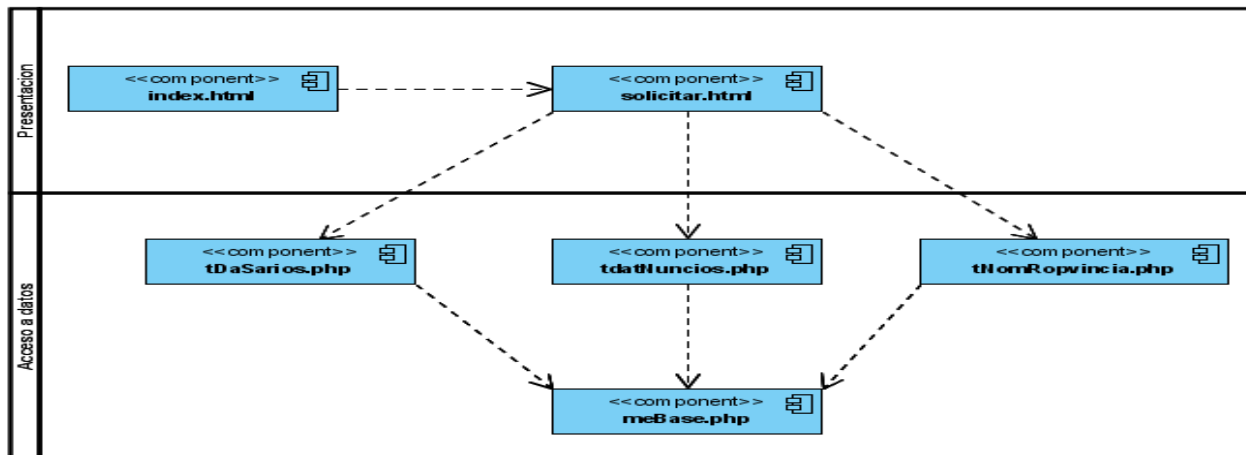
Esta vista proporciona la oportunidad de establecer correspondencias entre las clases y los componentes de implementación y nodos. La vista de implementación se representa con los diagramas de componentes.

4.2.1- ¿Qué es Componente?

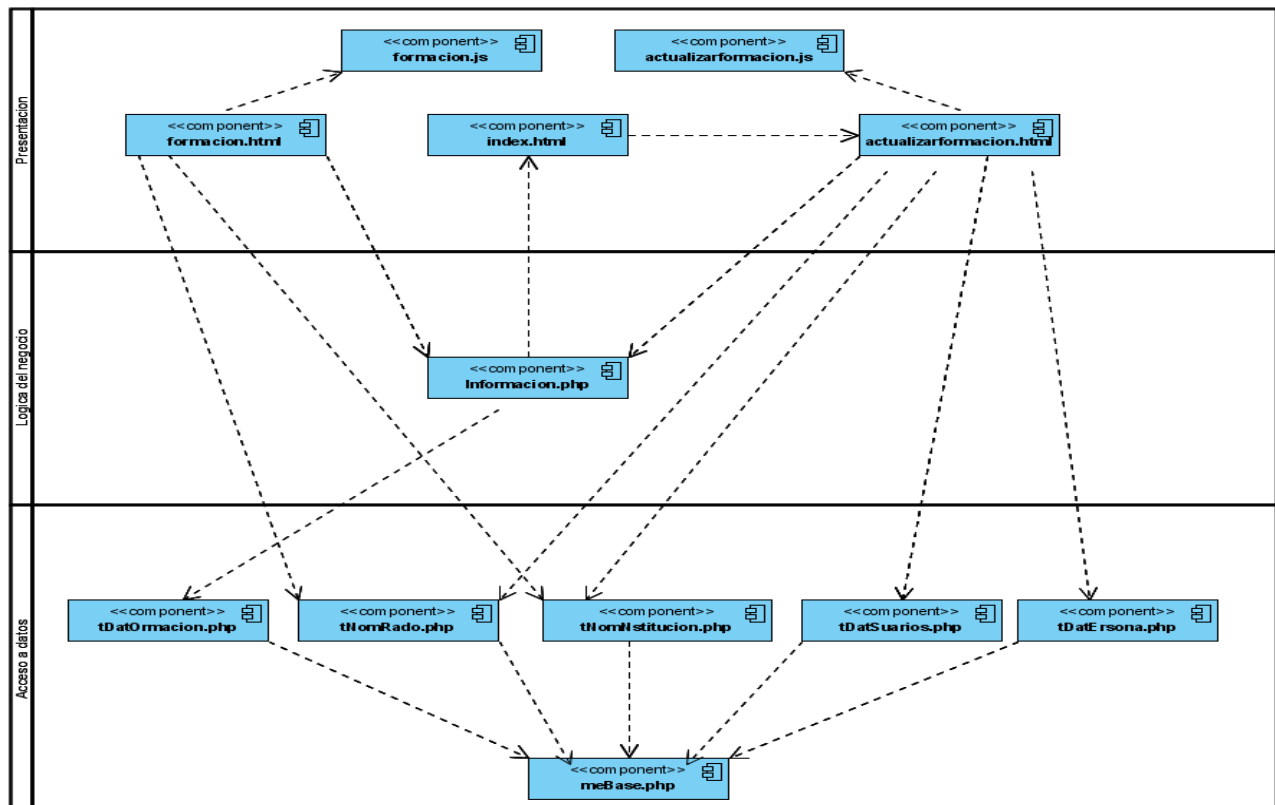
Es una parte física reemplazable de un sistema que empaqueta su implementación y es conforme a un conjunto de interfaces a las que proporciona su realización. Algunos componentes tienen identidad y pueden poseer entidades físicas, que incluyen objetos en tiempo de ejecución, documentos, bases de datos, etc. Los componentes existentes en el dominio de la implementación son unidades físicas en los computadores que se pueden conectar con otros componentes, sustituir, trasladar, archivar, etc. Los componentes tienen dos características: Empaquetan el código que implementa la funcionalidad de un

sistema, y algunas de sus propias instancias de objetos que constituyen el estado del sistema. Los llamados últimos componentes de la identidad, porque sus instancias poseen identidad y estado.[20]

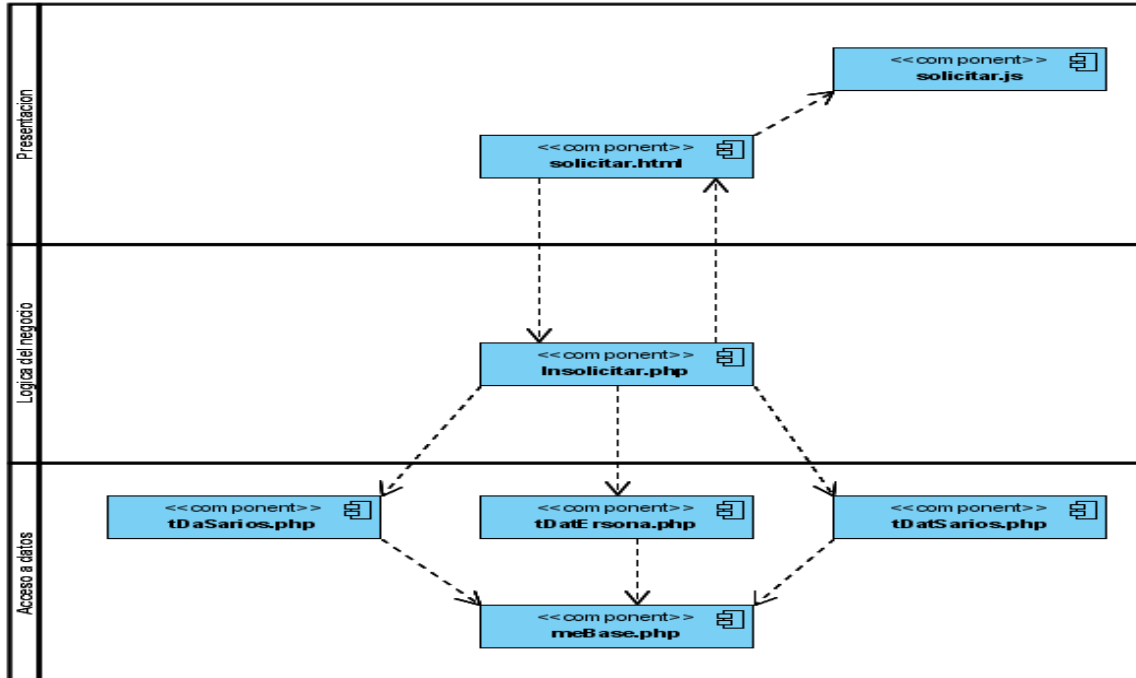
Mostrar ofertas



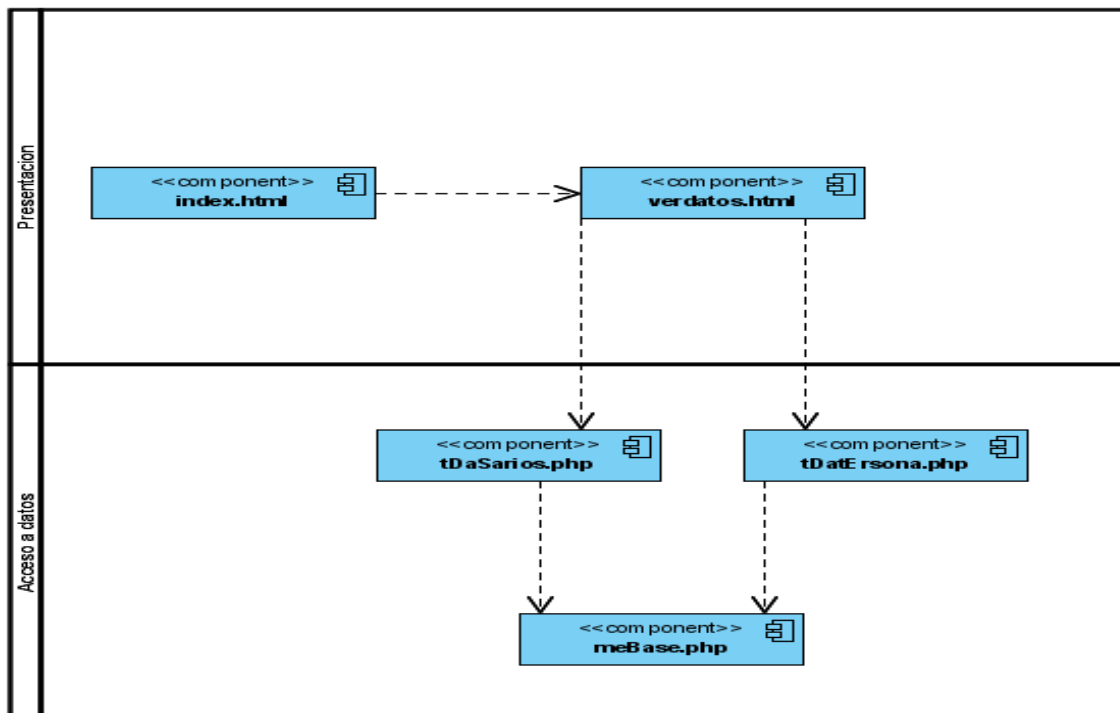
Gestionar expedientes



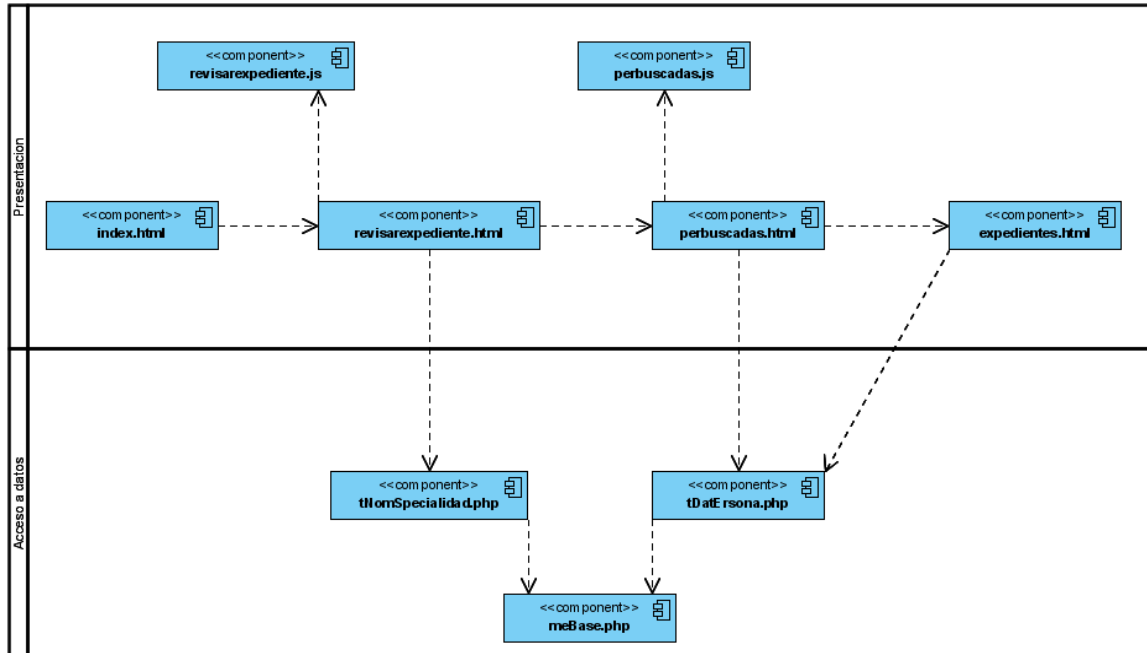
Solicitar plaza



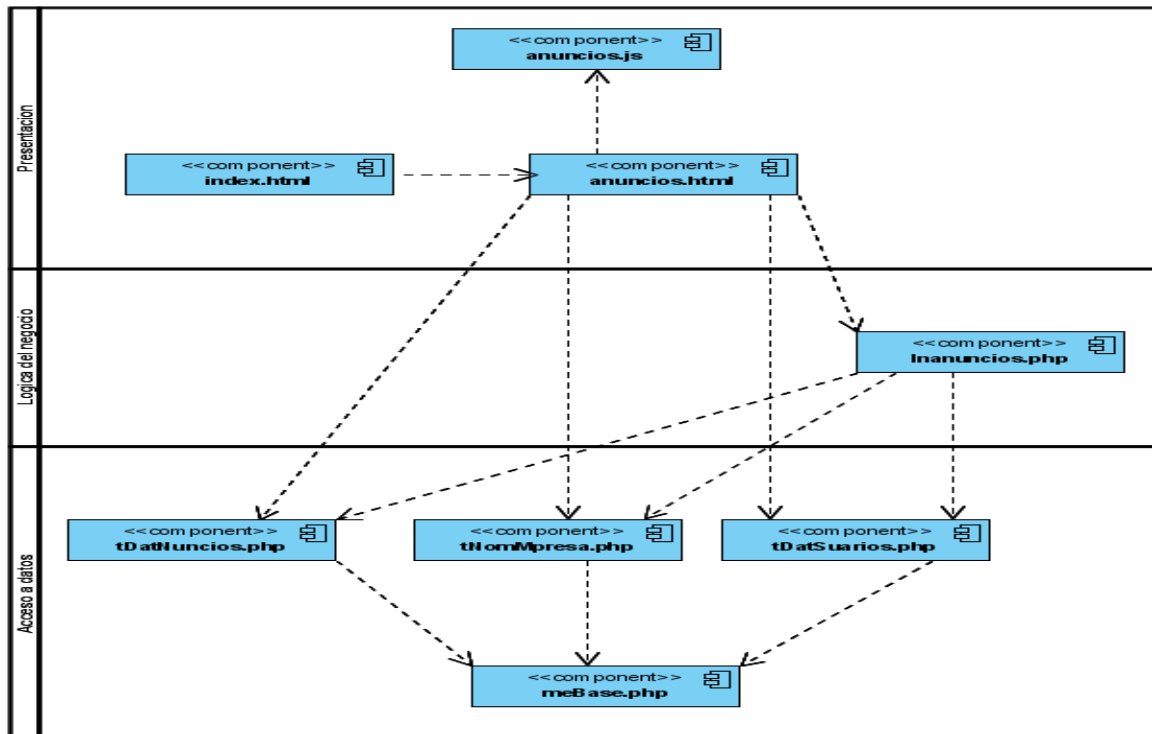
Ver datos



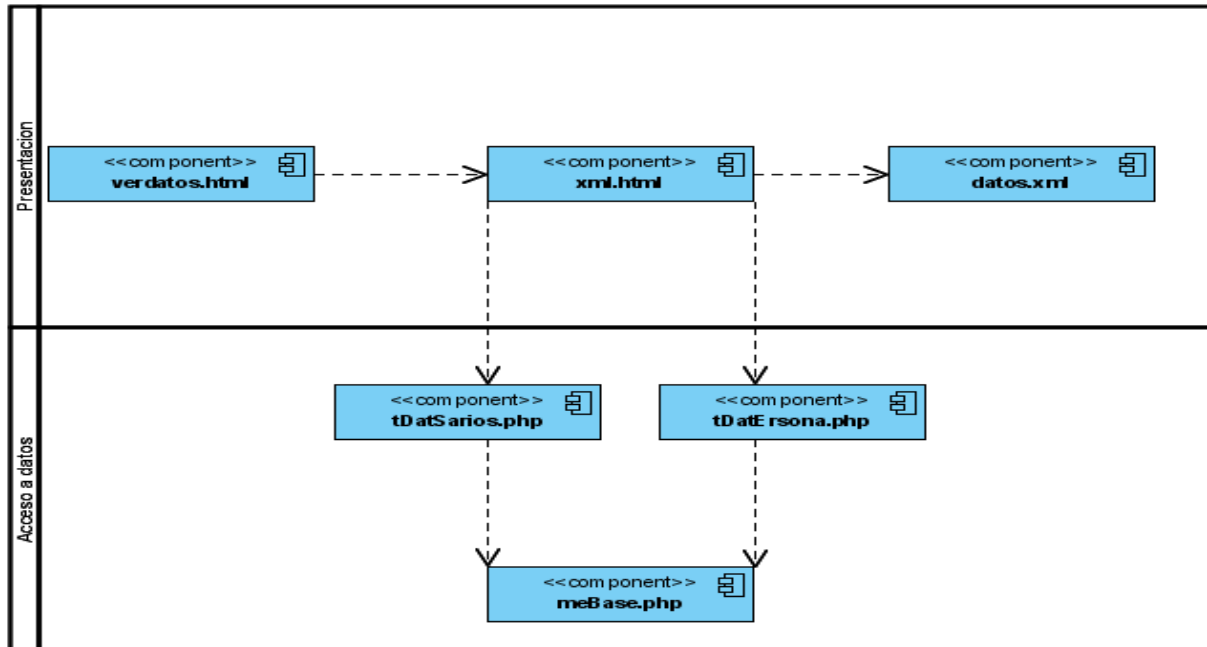
Revisar expedientes



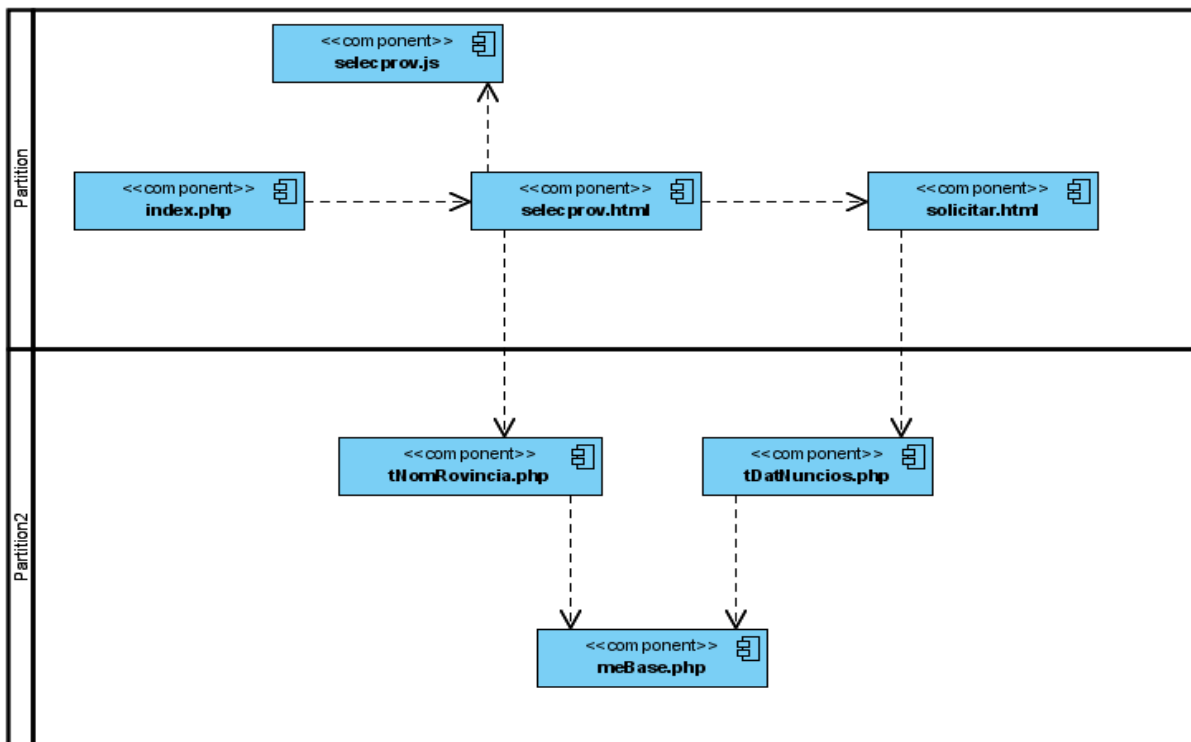
Gestionar anuncio



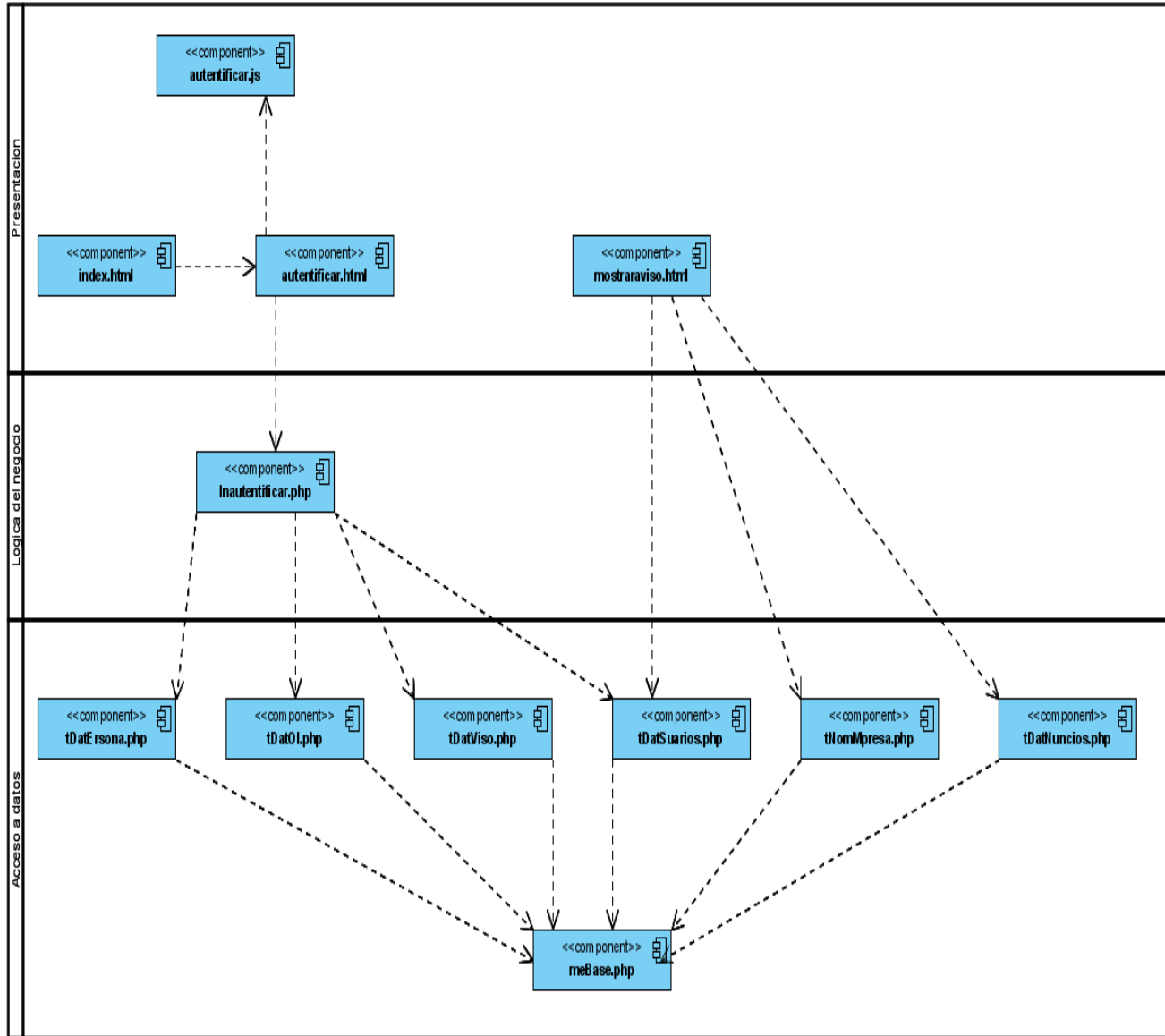
Generar XML



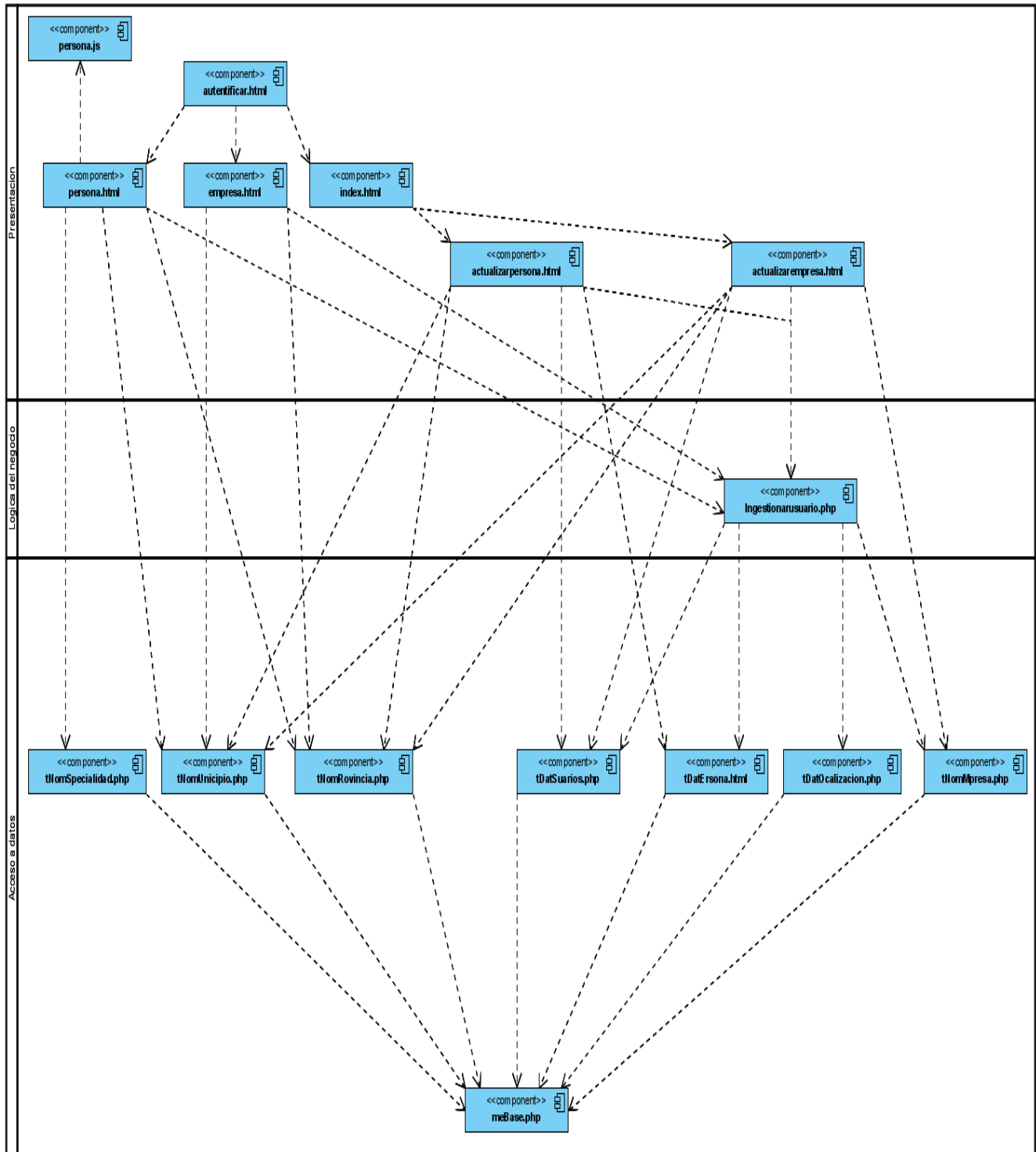
Buscar solicitud en otra provincia



Avisar



Gestionar datos del usuario



Ver más: Anexo 7.

4.3- Prueba:

La etapa de pruebas tiene una gran importancia para garantizar la calidad final del software, aunque la realización de las mismas no elimina la posibilidad de existencia de errores. Las pruebas son actividades en las cuales un sistema o componente es ejecutado bajo unas condiciones o requerimientos especificados, los resultados son observados y registrados, y se hace una evaluación de algún aspecto del sistema o componente.

Cualquier proceso de ingeniería puede ser probado de una de dos formas:

- Se pueden llevar a cabo pruebas que demuestren que cada función es completamente operativa.
- Se pueden desarrollar pruebas que aseguren que la operación interna se ajusta a las especificaciones y que todos los componentes internos se han comprobado de forma adecuada.

La primera aproximación se denomina prueba de la caja negra y la segunda prueba de la caja blanca.

Prueba de caja negra: Las pruebas se llevan a cabo sobre la interfaz del software, y es completamente indiferente el comportamiento interno y la estructura del programa. Los casos de prueba de la caja negra pretende demostrar que:

- Las funciones del software son operativas.
- La entrada se acepta de forma adecuada.
- Se produce una salida correcta.
- La integridad de la información externa se mantiene.

Se derivan conjuntos de condiciones de entrada que ejerciten completamente todos los requerimientos funcionales del programa. La prueba de la caja negra intenta encontrar errores de las siguientes categorías:

- Funciones incorrectas o ausente.
- Errores de interfaz.
- Errores en estructuras de datos o en accesos a bases de datos externas.
- Errores de rendimiento.
- Errores de inicialización y de terminación.

Los casos de prueba deben satisfacer los siguientes criterios:

- Reducir, en un coeficiente que es mayor que uno, el número de casos de prueba adicionales.
- Que digan algo sobre la presencia o ausencia de clases de errores.

4.3.1- Modelo de pruebas

CU Mostrar ofertas.		
Entrada	Resultados	Condiciones
		El desempleado debe de estar registrado y autenticado en el sitio.
	El sistema muestra una lista de todos los anuncios existentes realizados por las empresas de la misma provincia de residencia del desempleado.	

CU Ver datos.		
Entrada	Resultados	Condiciones
		El desempleado debe de estar registrado y autenticado en el sitio.
	El sistema muestra una lista de todos los datos del desempleado, ya sean laborales o personales.	

CU Gestionar expediente.		
Entrada	Resultados	Condiciones
		El campo “Experiencia” solo acepta números con un máximo de tres cifras.
Localidad: “matanzas” Experiencia: “3” Título: “ingeniero”	El sistema creo con éxito el expediente, archivándolo en la base de datos.	

<p>Finalizado: "2007/12/12"</p> <p>Comentario: "esto es una prueba"</p> <p>Grado: "Universitario"</p> <p>Institución: "Universidad"</p> <p>Nombre de la institución: "UCI"</p>		
---	--	--

CU Buscar solicitud en otra provincia.		
Entrada	Resultados	Condiciones
		El desempleado debe de estar registrado y autenticado en el sitio.
Seleccione la provincia: "Matanzas"	El sistema muestra una lista con todas las plazas ofertadas en la provincia la cual seleccionó el usuario.	

CU Solicitar.		
Entrada	Resultados	Condiciones
		El sistema debe de haber generado una lista con los laborales.
Botón "Solicitar"	El sistema actualiza la base de datos con la solicitud del optante.	

CU Gestionar datos del usuario (Insertar, persona).		
Entrada	Resultados	Condiciones
		El campo " Carné Ident " solo acepta números con 11 dígitos.

		<p>El campo “Edad” solo acepta números de hasta tres dígitos</p> <p>El campo “E-mail” solo acepta direcciones de correos válidas.</p> <p>El campo “Telef” solo aceptas números.</p> <p>El campo “Cod Postal” solo acepta números.</p>
<p>1er Nombre: “Reynold”</p> <p>2do Nombre: “Luis”</p> <p>1er Apellido: “Elías”</p> <p>2do Apellido: “Hernández”</p> <p>Carné Ident: “85011210421”</p> <p>Estado: “soltero”</p> <p>Fecha Nac: “1985/01/12”</p> <p>Edad: “23”</p> <p>Sexo: “masculino”</p> <p>Especialidad: “obrero”</p> <p>Dirección: “calle 3ra, #801, e/ 8 y 10”</p> <p>Provincia: “Matanzas”</p> <p>Municipio:</p>	<p>El sistema introdujo con éxito los datos creando así los datos personales de este desempleado.</p>	

<p>“Limonar”</p> <p>E-mail: “relias@estudiantes.uci.cu”</p> <p>Telef: “280054”</p> <p>Cod Postal: “40100”</p>		
--	--	--

CU Gestionar datos del usuario (Insertar, empresa).		
Entrada	Resultados	Condiciones
		<p>El campo “E-mail” solo acepta direcciones de correos válidas.</p> <p>El campo “Telef” solo aceptas números.</p> <p>El campo “Cod Postal” solo acepta números.</p>
<p>Nombre: “CUBALSE”</p> <p>Dirección: “Aquí va una dirección”</p> <p>Provincia: “Matanzas”</p> <p>Municipio: “Matanzas”</p> <p>E-mail: “empresa@cubemp.mtz.cu”</p> <p>Telef: “262398”</p> <p>Cod Postal: “40100”</p>	<p>El sistema introdujo con éxito los datos creando así los datos institucionales de la empresa.</p>	

CU Gestionar datos del usuario (Modificar, persona).		
Entrada	Resultados	Condiciones

<p>1er Nombre: "Pepe"</p> <p>2do Nombre:</p> <p>1er Apellido: "Rivero"</p> <p>2do Apellido: "Hernández"</p> <p>Carné Ident: "81071230421"</p> <p>Estado: "casado"</p> <p>Fecha Nac: "1985/01/12"</p> <p>Edad: "23"</p> <p>Sexo: "masculino"</p> <p>Especialidad: "diseñador"</p> <p>Dirección: "calle 4ta, #23, e/ 3 y 7"</p> <p>Provincia: "Villa Clara"</p> <p>Municipio: "Santa Clara"</p> <p>E-mail: "relias@estudiantes.uci.cu"</p> <p>Telef: "370054"</p> <p>Cod Postal: "50600"</p>	<p>El sistema modificó correctamente los datos del desempleado, actualizando así esta información.</p>	<p>El desempleado debe de estar registrado y autenticado en el sitio.</p>
--	--	---

CU Gestionar datos del usuario (Modificar, empresa).		
Entrada	Resultados	Condiciones
		<p>El desempleado debe de estar registrado y autenticado en el sitio.</p> <p>El campo “Carné Ident” solo acepta números con 11 dígitos.</p> <p>El campo “Edad” solo acepta números de hasta tres dígitos</p> <p>El campo “E-mail” solo acepta direcciones de correos válidas.</p> <p>El campo “Telef” solo aceptas números.</p> <p>El campo “Cod Postal” solo acepta números.</p>
<p>Nombre: “CARACOL”</p> <p>Dirección: “Esto es una prueba”</p> <p>Provincia: “Matanzas”</p> <p>Municipio: “Varadero”</p> <p>E-mail: “caracol@cubemp.mtz.cu”</p> <p>Telef: “512354”</p> <p>Cod Postal: “43100”</p>	<p>El sistema modificó correctamente los datos de la empresa, actualizando así esta información.</p>	

CU Revisar expedientes.		
Entrada	Resultados	Condiciones
		<p>La empresa debe de estar registrada y autenticada en el sitio.</p>

Especialidad: "Obrero"	El sistema muestra una lista de todos los desempleados que son del tipo seleccionado y que optaron por alguna de sus plazas ofertadas.	
Botón: "Ver"	El sistema muestra, todos los datos del desempleado seleccionado por la empresa.	

CU Gestionar anuncio (Insertar).		
Entrada	Resultados	Condiciones
		La empresa debe de estar registrada y autenticada en el sitio.
Fecha: "2008/05/27" Plaza: "Chofer" Descripción: "Chofer de camión con más de 2 años de experiencia"	El sistema insertó con éxito el anuncio deseado.	

CU Gestionar anuncio (Eliminar).		
Entrada	Resultados	Condiciones
		La empresa debe de haber publicado al menos un anuncio. El sistema muestra una lista con todos los anuncios realizados por la empresa.
Botón: "Eliminar"	El sistema eliminó con éxito el anuncio seleccionado.	

CU Generar XML.		
Entrada	Resultados	Condiciones
		El desempleado debe de haber mostrados sus datos personales.
Botón "Generar XML"	Se generó en formato XML todos los datos del desempleado.	

CU Avisar.		
Entrada	Resultados	Condiciones
		La empresa debe de estar registrada y autenticada en el sitio.
Anuncio: "mecánico"	El sistema muestra una lista de todos los desempleados que optaron por el anuncio seleccionado.	
Botón: "Avisar"	El sistema guarda en la base de datos el o los aviso(s) a lo(s) desempleado(s).	

Ver más: Anexo 8.

Conclusiones

En este capitulo quedo analizado de forma integra todos los diagramas de despliegue y componentes, lo cual contribuyo de forma detallada al diseño del sistema, para llevar una implementación mas segura y entendible.

CONCLUSIONES

Durante la realización del presente trabajo de diploma, se ha seguido la metodología RUP, donde divide el ciclo de vida de la producción del software en 4 fases. Cada fase de está representada por diferentes flujos, los cuales juegan un papel insustituible.

Como parte de este proceso:

- Se realizó un estudio profundo de las herramientas a emplear para la construcción del la aplicación.

- Se analizaron las características de aplicaciones web que realizan la misma función y se encuentran en la red (Internet).

- Se estudió y modeló la actualidad de los procesos

- Se diseño una solución a la situación problémica existente.

- Se implementó una aplicación web como colofón al diseño creado capaz de gestionar todo lo referente a la captación de un personal para un puesto de trabajo determinado.

RECOMENDACIONES

Publicar este trabajo.

Realizar una prueba piloto para ver la aceptación que tiene entre la población y las empresas.

Trabajar en nuevas mejoras para el diseño del sitio, teniendo en cuenta la opinión de los usuarios.

BIBLIOGRAFÍA**Trabajos Referenciados**

1. **JAÉN, MIVIA.** <http://www.monografias.com>. [Online] [Cited: 01, 16, 2008.] <http://www.monografias.com/trabajos14/bolsa-trabajo/bolsa-trabajo.shtml#IMPLI>.
2. www.protean.es. [Online] [Cited: 01, 16, 2008.] <http://www.protean.es/content/view/239/465/>.
3. **Zamitz, Carlos Alberto Román.** www.profesores.fi-b.unam.mx. [Online] [Cited: 01, 16, 2008.] <http://profesores.fi-b.unam.mx/carlos/aydoo/usecase.html>.
4. **Arenas, María Isabel García.** www.geneura.ugr.es. [Online] [Cited: 01, 16, 2008.] <http://geneura.ugr.es/~maribel/xml/introduccion/index.shtml#15>.
5. **Reagle, Joseph.** www.html.conclase.net. [Online] [Cited: 01, 16, 2008.] <http://html.conclase.net/w3c/html401-es/intro/intro.html>.
6. **Guzmán, Mario Alberto Arredondo.** www.manuales.dgsca.unam.mx. [Online] [Cited: 01, 17, 2008.] http://manuales.dgsca.unam.mx/xml/qu%E9_es.html.
7. www.pgadmin.org. [Online] [Cited: 01, 17, 2008.] <http://www.pgadmin.org/features.php>.
8. www.dcc.uchile.cl. [Online] [Cited: 01, 17, 2008.] <http://www.dcc.uchile.cl/~cc61j/rup/sld001.htm>.
9. **Taylor, Jennifer.** www.adobe.com/es/devnet/dreamweaver. [Online] [Cited: 01, 20, 2008.] http://www.adobe.com/es/devnet/dreamweaver/articles/dw8_newfeatures.html.
10. **Lago, Ramiro.** www.proactiva-calidad.com. [Online] [Cited: 01, 20, 2008.] <http://www.proactiva-calidad.com/java/patrones/mvc.html>.
11. www.httpd.apache.org. [Online] [Cited: 01, 20, 2008.] http://httpd.apache.org/docs/2.0/es/new_features_2_0.html.
12. php.uci.cu. [Online] [Cited: 01, 20, 2008.] <http://php.uci.cu/?q=node/9>.
13. www.sobl.org. [Online] [Cited: 01, 20, 2008.] <http://www.sobl.org/traduccion/practical-postgres/node19.html>.
14. **Pelloutier, Fernand.** http://www.antorcha.net/biblioteca_virtual/. [Online] [Cited: 01, 23, 2008.] http://www.antorcha.net/biblioteca_virtual/historia/bolsas/indice.html.
15. **Azara, Felix de.** www.e-estrategia.com.ar. [Online] [Cited: 02, 08, 2008.] <http://www.e-estrategia.com.ar/ediciones/edicion0052/presente%20y%20futuro%20de%20los%20si.pdf>.

Trabajos Consultados

Pressman, R. *Software Engineering. A Practitioner's Approach*. Fourth Edition. McGraw – Hill. USA, 1999.

Booch, G., Rumbaugh, J., Jacobson, I. *El Lenguaje Unificado de Modelado*. Addison-Wesley. 1999.

Larman, C. *UML Y PATRONES, Introducción al análisis y diseño orientado a objetos*. La Habana. Cuba 2004.

Booch, G., Rumbaugh, J., Jacobson, I. *El Proceso Unificado de Desarrollo de Software*. La Habana. Cuba 2004.

GLOSARIO

Nodo: Por definición punto donde convergen más de dos líneas. A veces se refiere a una única máquina en Internet. Normalmente se refiere a un punto de confluencia en una red.

PERL: Lenguaje para manipular textos, ficheros y procesos. Con estructura de scrip. Desarrollado por Larry Wall, es multiplataforma ya que funciona en *Unix*.

HTML: HyperText Markup Language. Lenguaje de Marcas de Hypertexto. Lenguaje para elaborar páginas Web actualmente se encuentra en su versión 3. Fue desarrollado en el *CERN*.

Http: HyperText Transfer Protocol. Protocolo de Transferencia de Hipertexto. Protocolo usado en *www*.

JAVASCRIPT: Programa escrito en el lenguaje script de Java que es interpretado por la aplicación cliente, normalmente un navegador (Browser).

SQL: Structured Query Language. Lenguaje de Petición Estructurada. Lenguaje para base de datos.

Arquitectura: Consiste en un conjunto de patrones y abstracciones coherentes que proporcionan el marco de referencia necesario para guiar la construcción del software para un sistema de información. La arquitectura software establece los fundamentos para que analistas, diseñadores y programadores trabajen en una línea común que permita alcanzar los objetivos y necesidades del sistema de información.

Patrón: Es una solución a un problema no trivial que es efectiva (ya se resolvió el problema satisfactoriamente en ocasiones anteriores) y reusable (se puede aplicar a diferentes problemas en distintas circunstancias).

RUP: El Rational Unified Process (RUP) es una metodología formal, a veces también llamada proceso. El RUP describe a gran detalle todas las actividades, roles, responsabilidades, productos de trabajo y herramientas para definir quién hace qué y en qué momento en un proyecto de desarrollo de software

Autenticación: (Griego: *αυθεντικός* = verdadero o genuino, de ' los authentes' = el autor) es el acto de establecimiento o confirmación de algo (o alguien) como auténtico, es decir que reclama hecho por o sobre la cosa son verdadero. La autenticación de un objeto puede significar (pensar) la confirmación de su procedencia, mientras que la autenticación de una persona a menudo consiste en verificar su identidad. La autenticación depende de uno o varios factores de autenticación.


XML: Sigla en inglés de eXtensible Markup Language («lenguaje de marcas extensible»), es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C).

Arquitectura MVC: Es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. El patrón MVC se ve frecuentemente en aplicaciones web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página, el controlador es el Sistema de Gestión de Base de Datos y el modelo es el modelo de datos.

RRHH: Referente a la parte empresarial relacionada con recursos humanos.

ANEXOS

Anexo 1

Caso de uso	Gestionar registro del usuario
Actor(es):	Usuario
Propósito:	Crear un nuevo usuario, ya sea desempleado o empresa.
Resumen:	El caso de uso se inicia cuando un usuario no registrado quiere registrarse, para así gozar de todas las facilidades que brinda la aplicación, o si ya esta registrado y desea modificar la contraseña, el usuario o el rol.
Precondiciones:	
Poscondiciones:	
Tipo:	Real y expandido.
Responsabilidades:	RF8
Casos de usos relacionados:	
Interfaz I	
	
Interfaz II	

Bolsa Laboral Online

Interfaz III

Bolsa Laboral Online

Inicio | Estadística | Autenticarse | Contáctenos | Bienvenido: Reynold

(1) Registrar nuevo usuario

(2) Usuario

(3) Contraseña

(4) Seleccionar rol

(6) Usuario

(7) Contraseña

(8) Nueva contraseña

(9) Repetir la nueva contraseña	
(10) Seleccionar rol	
Curso normal de eventos para el caso de uso	
Acción del actor	Respuesta del sistema
Sección Insertar usuario	
1- El usuario va a la opción de “Registrarse” (1).	2- La aplicación le muestra un formulario al usuario (usuario (2), contraseña (3), rol (4)).
3- El usuario llena los campos solicitados.	
4- El usuario presiona el botón “Aceptar” (5).	5- El sistema actualiza la base de datos con el nuevo usuario creado.
Sección Modificar usuario	
	1- La empresa le muestra una serie de campos (usuario (6), contraseña (7), nueva contraseña (8), Rep. Contraseña (9), rol (10)).
2- El usuario llena o modifica los campos	
3- El usuario presiona el botón “Modificar” (11).	4- El sistema actualiza la base de datos modificando los datos antiguos con estos nuevos.
Cursos alternos	

Caso de uso	Generar estadística
Actor(es):	Usuario
Propósito:	Generar un gráfico con las estadísticas de las plazas ofertadas por cada empresa en un periodo de tiempo determinado.
Resumen:	El caso de uso se inicia cuando un usuario esta interesado en saber con que frecuencia se oferta alguna plaza laboral., mostrándole el sistema una gráfica con esta información.
Precondiciones:	
Poscondiciones:	
Tipo:	Real y expandido.
Responsabilidades:	RF11

<p>Casos de usos relacionados:</p>	
<p>Interfaz I</p>	
<p>Interfaz II</p>	

(1) Generar estadística.	
(2) Introducir año.	
Curso normal de eventos para el caso de uso	
Acción del actor	Respuesta del sistema
1- El usuario va a la opción "Estadística" (1).	2- El sistema muestra una aplicación para que el usuario seleccione el año (2).
3- El usuario introduce el año.	
4- El usuario presiona el botón "Aceptar" (3).	4- El sistema realiza la búsqueda mediante una consulta a la base de datos.
	5- El sistema muestra un gráfico con los resultados de la búsqueda.
Cursos alternos	

Caso de uso	Gestionar datos del usuario
Actor(es):	Usuario
Propósito:	Actualizar los datos personales del usuario en caso de sufrir algún cambio.
Resumen:	El caso de uso se inicia cuando el usuario quiera introducir algún dato personal o institucional dependiendo del tipo de usuario que sea, en caso de ya haberlo puede modificar estos, actualizando así la base de datos con los nuevos valores.
Precondiciones:	
Poscondiciones:	
Tipo:	Real y expandido.
Responsabilidades:	RF12
Casos de usos relacionados:	
Interfaz I	

Bolsa Laboral Online

Inicio | Estadística | Autentificarse | Contáctenos | Bienvenido:

Desempleado

- Ofertas
- Avisos
- Ver datos
- Expediente
- Otras provincias
- Actualizar datos

1er nombre <input type="text"/>	2do nombre <input type="text"/>
1er apellido <input type="text"/>	2do apellido <input type="text"/>
Carnet Ident. <input type="text"/>	estado <input type="text"/>
fecha nac <input type="text"/>	edad <input type="text"/>
sexo <input type="text"/>	especialidad <input type="text"/>
Direccion <input type="text"/>	
provincia <input type="text"/>	municipio <input type="text"/>
E-mail <input type="text"/>	telef <input type="text"/>
Cod Postal <input type="text"/>	
<input type="button" value="Aceptar"/>	

Interfaz II

Bolsa Laboral Online

Inicio | Estadística | Autentificarse | Contáctenos | Ayuda

Desempleado

- Ofertas
- Avisos[3]
- Ver datos
- Expediente
- Otras provincias
- Actualizar datos

Interfaz III																																														
<div data-bbox="186 336 722 388" style="text-align: center;"> <h2>Bolsa Laboral Online</h2> </div> <div data-bbox="365 451 1291 493" style="text-align: center; background-color: black; color: white; padding: 5px;"> Inicio Estadística Autenticarse Contáctenos Bienvenido: Reynold </div> <div style="display: flex; justify-content: space-between; margin-top: 20px;"> <div data-bbox="170 556 560 892" style="width: 25%; border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;"> <p style="text-align: center; font-weight: bold; color: #0070C0; margin-bottom: 10px;">Desempleado</p> <ul style="list-style-type: none"> ● Ofertas ● Avisos ● Ver datos ● Expediente ● Otras provincias ● Actualizar datos </div> <div data-bbox="560 588 1485 1123" style="width: 75%; padding: 10px;"> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">1er nombre</td> <td><input type="text" value="Reynold"/></td> <td style="width: 5%; text-align: center;">*</td> <td style="width: 50%;">2do nombre</td> <td><input type="text"/></td> </tr> <tr> <td>1er apellido</td> <td><input type="text" value="Elias"/></td> <td style="text-align: center;">*</td> <td>2do apellido</td> <td><input type="text" value="Hernandez"/></td> </tr> <tr> <td>Carnet Ident.</td> <td><input type="text" value="85011210421"/></td> <td style="text-align: center;">*</td> <td>estado</td> <td><input type="text" value=""/></td> </tr> <tr> <td>fecha nac</td> <td><input type="text" value="1985/01/12"/></td> <td style="text-align: center;">*</td> <td>edad</td> <td><input type="text" value="23"/></td> </tr> <tr> <td>sexo</td> <td><input type="text" value="masculino"/></td> <td style="text-align: center;">*</td> <td>especialidad</td> <td><input type="text" value=""/></td> </tr> <tr> <td colspan="5">Direccion <input type="text" value="calle 3ra. #801, e/ 8 y 10"/></td> </tr> <tr> <td>provincia</td> <td><input type="text" value=""/></td> <td style="text-align: center;">*</td> <td>municipio</td> <td><input type="text" value=""/></td> </tr> <tr> <td>E-mail</td> <td><input type="text" value="rey@estudiantes.uci.cu"/></td> <td></td> <td>Telef</td> <td><input type="text" value="19 23"/></td> </tr> <tr> <td>Cod Postal</td> <td><input type="text" value="40100"/></td> <td style="text-align: center;">*</td> <td colspan="2" style="text-align: center;"> <input type="button" value="Actualizar"/> </td> </tr> </table> </div> </div>		1er nombre	<input type="text" value="Reynold"/>	*	2do nombre	<input type="text"/>	1er apellido	<input type="text" value="Elias"/>	*	2do apellido	<input type="text" value="Hernandez"/>	Carnet Ident.	<input type="text" value="85011210421"/>	*	estado	<input type="text" value=""/>	fecha nac	<input type="text" value="1985/01/12"/>	*	edad	<input type="text" value="23"/>	sexo	<input type="text" value="masculino"/>	*	especialidad	<input type="text" value=""/>	Direccion <input type="text" value="calle 3ra. #801, e/ 8 y 10"/>					provincia	<input type="text" value=""/>	*	municipio	<input type="text" value=""/>	E-mail	<input type="text" value="rey@estudiantes.uci.cu"/>		Telef	<input type="text" value="19 23"/>	Cod Postal	<input type="text" value="40100"/>	*	<input type="button" value="Actualizar"/>	
1er nombre	<input type="text" value="Reynold"/>	*	2do nombre	<input type="text"/>																																										
1er apellido	<input type="text" value="Elias"/>	*	2do apellido	<input type="text" value="Hernandez"/>																																										
Carnet Ident.	<input type="text" value="85011210421"/>	*	estado	<input type="text" value=""/>																																										
fecha nac	<input type="text" value="1985/01/12"/>	*	edad	<input type="text" value="23"/>																																										
sexo	<input type="text" value="masculino"/>	*	especialidad	<input type="text" value=""/>																																										
Direccion <input type="text" value="calle 3ra. #801, e/ 8 y 10"/>																																														
provincia	<input type="text" value=""/>	*	municipio	<input type="text" value=""/>																																										
E-mail	<input type="text" value="rey@estudiantes.uci.cu"/>		Telef	<input type="text" value="19 23"/>																																										
Cod Postal	<input type="text" value="40100"/>	*	<input type="button" value="Actualizar"/>																																											
<div data-bbox="186 1228 722 1281" style="text-align: center;"> <h2>Bolsa Laboral Online</h2> </div> <div data-bbox="430 1344 1226 1386" style="text-align: center; background-color: black; color: white; padding: 5px;"> Inicio Estadística Autenticarse Contáctenos Bienvenido: </div> <div style="display: flex; justify-content: space-between; margin-top: 20px;"> <div data-bbox="170 1438 560 1774" style="width: 25%; border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;"> <p style="text-align: center; font-weight: bold; color: #0070C0; margin-bottom: 10px;">Empresa</p> <ul style="list-style-type: none"> ● Anuncios ● Expedientes ● Actualizar datos ● Dar aviso </div> <div data-bbox="560 1459 1388 1858" style="width: 75%; padding: 10px;"> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">Nombre</td> <td><input type="text"/></td> <td style="width: 5%; text-align: center;">*</td> </tr> <tr> <td>Direccion</td> <td><input type="text"/></td> <td style="text-align: center;">*</td> </tr> <tr> <td>Provincia</td> <td><input type="text" value=""/></td> <td style="text-align: center;">*</td> </tr> <tr> <td>Municipio</td> <td><input type="text" value=""/></td> <td style="text-align: center;">*</td> </tr> <tr> <td>E-mail</td> <td><input type="text"/></td> <td></td> </tr> <tr> <td>Telef</td> <td><input type="text"/></td> <td></td> </tr> <tr> <td>Cod. Postal</td> <td><input type="text"/></td> <td></td> </tr> <tr> <td colspan="3" style="text-align: center;"> <input type="button" value="Aceptar"/> </td> </tr> </table> </div> </div>		Nombre	<input type="text"/>	*	Direccion	<input type="text"/>	*	Provincia	<input type="text" value=""/>	*	Municipio	<input type="text" value=""/>	*	E-mail	<input type="text"/>		Telef	<input type="text"/>		Cod. Postal	<input type="text"/>		<input type="button" value="Aceptar"/>																							
Nombre	<input type="text"/>	*																																												
Direccion	<input type="text"/>	*																																												
Provincia	<input type="text" value=""/>	*																																												
Municipio	<input type="text" value=""/>	*																																												
E-mail	<input type="text"/>																																													
Telef	<input type="text"/>																																													
Cod. Postal	<input type="text"/>																																													
<input type="button" value="Aceptar"/>																																														

Interfaz V

Bolsa Laboral Online

Inicio | Estadística | Autenticarse | Contáctenos | Ayuda

Empresa

- Anuncios
- Expedientes
- Actualizar datos
- Dar aviso

22



Interfaz VI

Bolsa Laboral Online

Inicio | Estadística | Autenticarse | Contáctenos | Bienvenido: cubalse

Empresa

- Anuncios
- Expedientes
- Actualizar datos
- Dar aviso

Nombre *

Direccion *

Provincia *

Municipio *

E-mail

Telef 23

Cod. Postal

(1) Primer nombre del desempleado (2) Segundo nombre del desempleado (3) Primer Apellido del desempleado (4) Segundo Apellido del desempleado (5) Carnet de identidad (6) Estado marital (7) Fecha de nacimiento (8) Edad (9) Sexo (10) Especialidad (11) Dirección (12) Provincia (13) Municipio (14) Correo electrónico (15) Código postal (16) Teléfono (20) Nombre de la empresa	
Curso normal de eventos para el caso de uso	
Acción del actor	Respuesta del sistema
Sección Insertar datos(desempleado)	
	1- El sistema muestra una serie de campos que el desempleado debe de llenar (1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16).
2- El desempleado llena los campos solicitados.	
3- El desempleado presiona el botón "Aceptar" (17).	4- El sistema guarda en la base de datos estos valores.
Sección Modificar datos (desempleado)	
1- El desempleado va a la opción de "Actualizar datos" (18).	2- El sistema realiza una consulta a la base de datos donde obtienes todos los datos personales del desempleado.
	3- El sistema le muestra al desempleado en campos editables sus datos personales.

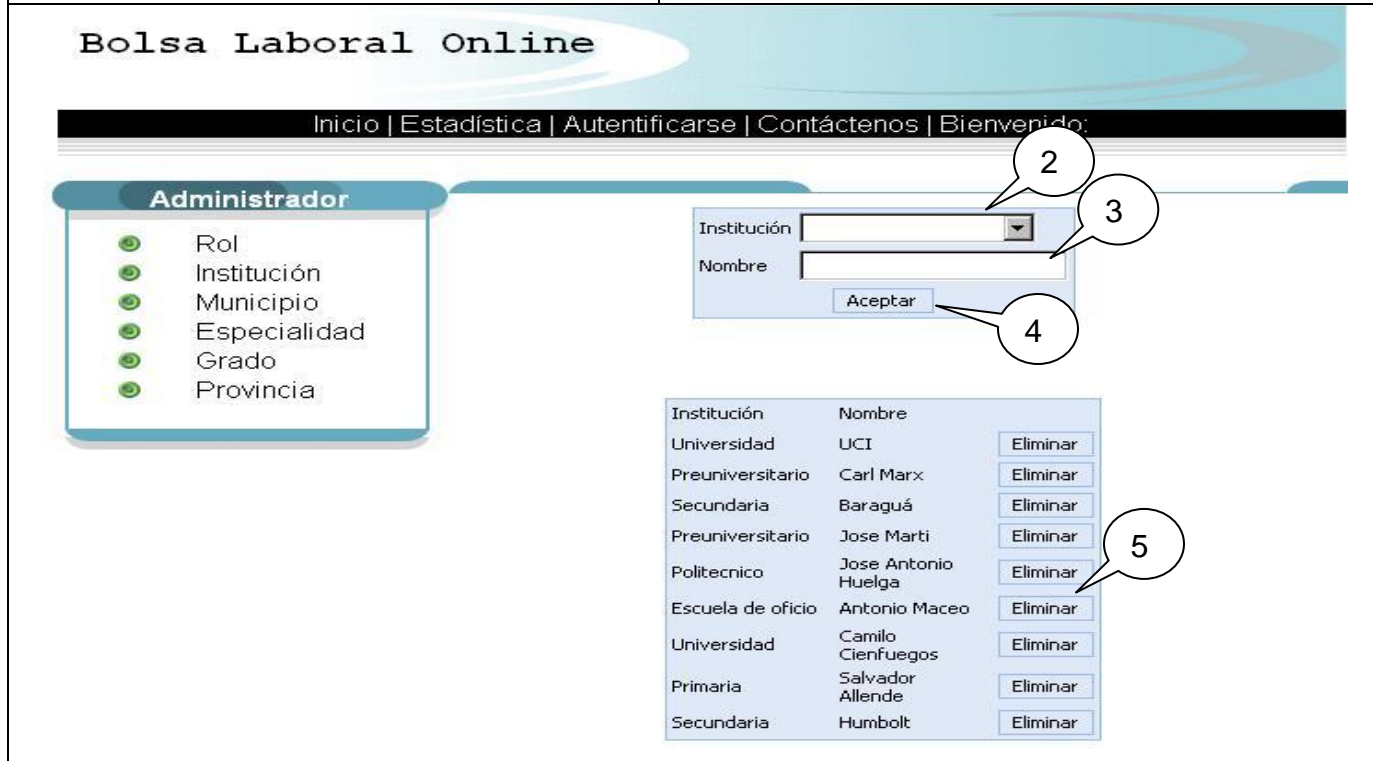
4- El usuario reescribe los datos personales que desea cambiar.	
5- El usuario presiona en el botón "Actualizar" (19).	6- El sistema realiza una consulta a la base de datos actualizando los datos del desempleado.
Sección Insertar datos (empresa)	
	1- El sistema muestra una serie de campos que la empresa debe de llenar (20, y la dirección que coincide en los mismo campos que la del desempleado).
2- La empresa llena los campos solicitados.	
3- La empresa presiona el botón "Aceptar" (21).	4- El sistema guarda en la base de datos estos valores.
Sección Modificar datos (empresa)	
1- La empresa va a la opción de "Actualizar datos" (22).	2- El sistema realiza una consulta a la base de datos donde obtienes todos los datos institucionales de la empresa.
	3- El sistema le muestra a la empresa en campos editables sus datos institucionales.
4- La empresa reescribe los datos personales que desea cambiar.	
5- La empresa presiona en el botón "Actualizar" (23).	6- El sistema realiza una consulta a la base de datos actualizando los datos de la empresa.
Cursos alternos	

Caso de uso	Actualizar nomenclador instituciones
Actor(es):	Administrador
Propósito:	Actualizar la lista de las instituciones.
Resumen:	Este caso de uso se inicia cuando el administrador del sistema va a introducir o eliminar una institución educativa, actualizando así la base de datos.
Precondiciones:	
Poscondiciones:	

Tipo:	Real y expandido.
Responsabilidades:	RF13
Casos de usos relacionados:	
Interfaz I	



Interfaz II	
--------------------	--

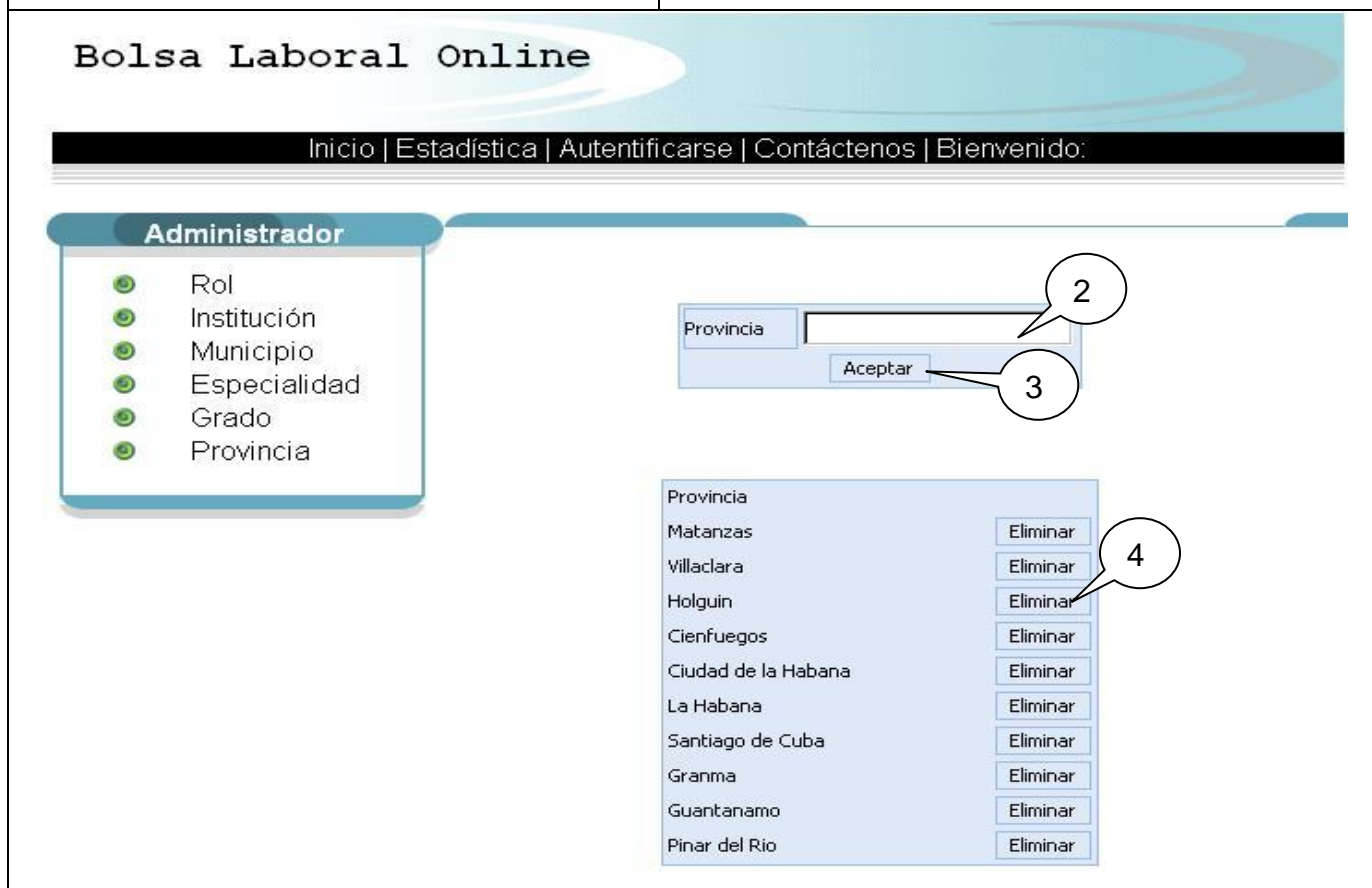


(2) Tipo de institución	
(3) Nombre de la institución	
Curso normal de eventos para el caso de uso	
Acción del actor	Respuesta del sistema
1- El administrador del sistema va a la opción "Institución" (1).	
Sección Insertar institución	
	1- El sistema muestra los campos a llenar para crear una nueva institución (2,3).
2- El administrador introduce los datos pedidos.	
3- El administrador presiona en el botón "Aceptar" (4).	4- El sistema actualiza la base de datos con la nueva institución creada.
Sección Eliminar institución	
	1- El sistema muestra una lista con todas las instituciones existentes en la base de datos.
2- El administrador presiona el botón "Eliminar" (5).	3- El sistema elimina de la base de datos la institución seleccionada.
Cursos alternos	

Caso de uso	Actualizar nomenclador provincia
Actor(es):	Administrador
Propósito:	Actualizar la lista con las provincias.
Resumen:	El caso de uso se inicia cuando el administrador desea insertar o eliminar una provincia, actualizando así la lista de las provincias.
Precondiciones:	
Poscondiciones:	
Tipo:	Real y expandido.
Responsabilidades:	RF14
Casos de usos relacionados:	
Interfaz I	

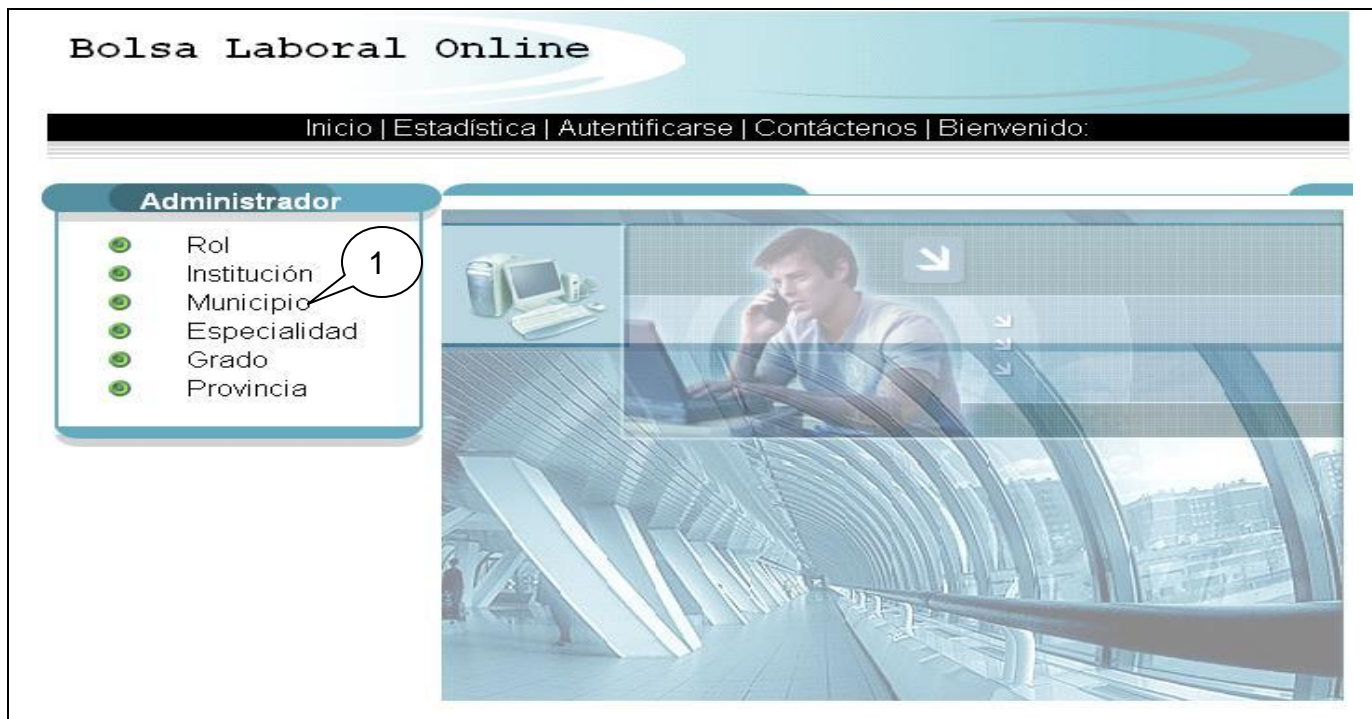


Interfaz II

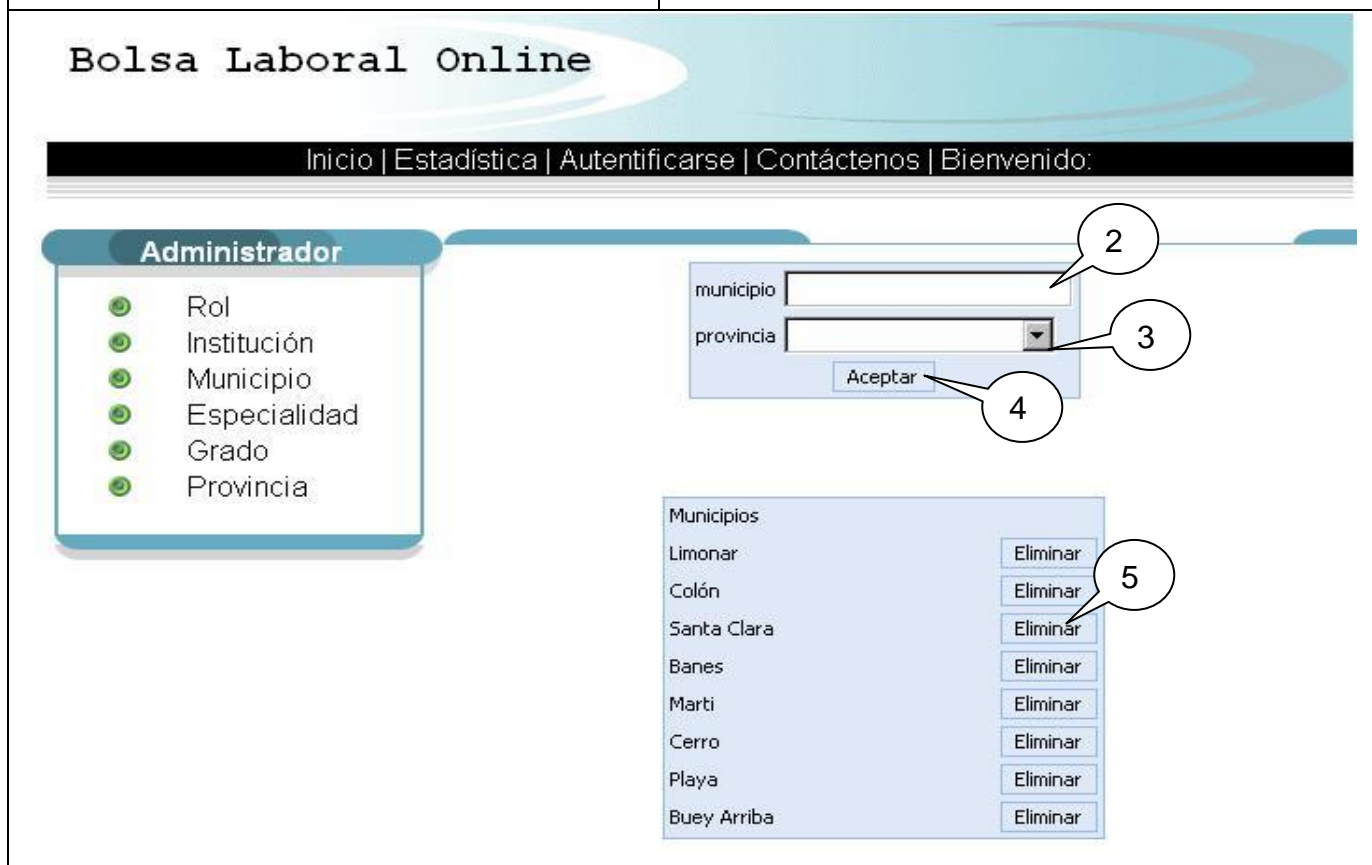


(2) Nombre de la provincia.	
Curso normal de eventos para el caso de uso	
Acción del actor	Respuesta del sistema
1- El administrador del sistema va a la opción "Provincia" (1).	
Sección Insertar provincia	
	1- El sistema muestra los campos a llenar para crear una nueva provincia (2).
2- El administrador introduce los datos pedidos.	
3- El administrador presiona el botón "Aceptar" (3).	4- El sistema actualiza la base de datos con la nueva provincia creada.
Sección Eliminar provincia	
	1- El sistema muestra una lista con todas las provincias existentes en la base de datos.
2- El administrador presiona el botón "Eliminar" (4).	3- El sistema elimina de la base de datos la provincia seleccionada.
Cursos alternos	

Caso de uso	Actualizar nomenclador municipios
Actor(es):	Administrador
Propósito:	Actualizar la lista con los municipios.
Resumen:	El caso de uso se inicia cuando el administrador insertar o eliminar un municipio, actualizando así la lista de municipio.
Precondiciones:	
Poscondiciones:	
Tipo:	Real y expandido.
Responsabilidades:	RF15
Casos de usos relacionados:	
Interfaz I	



Interfaz II

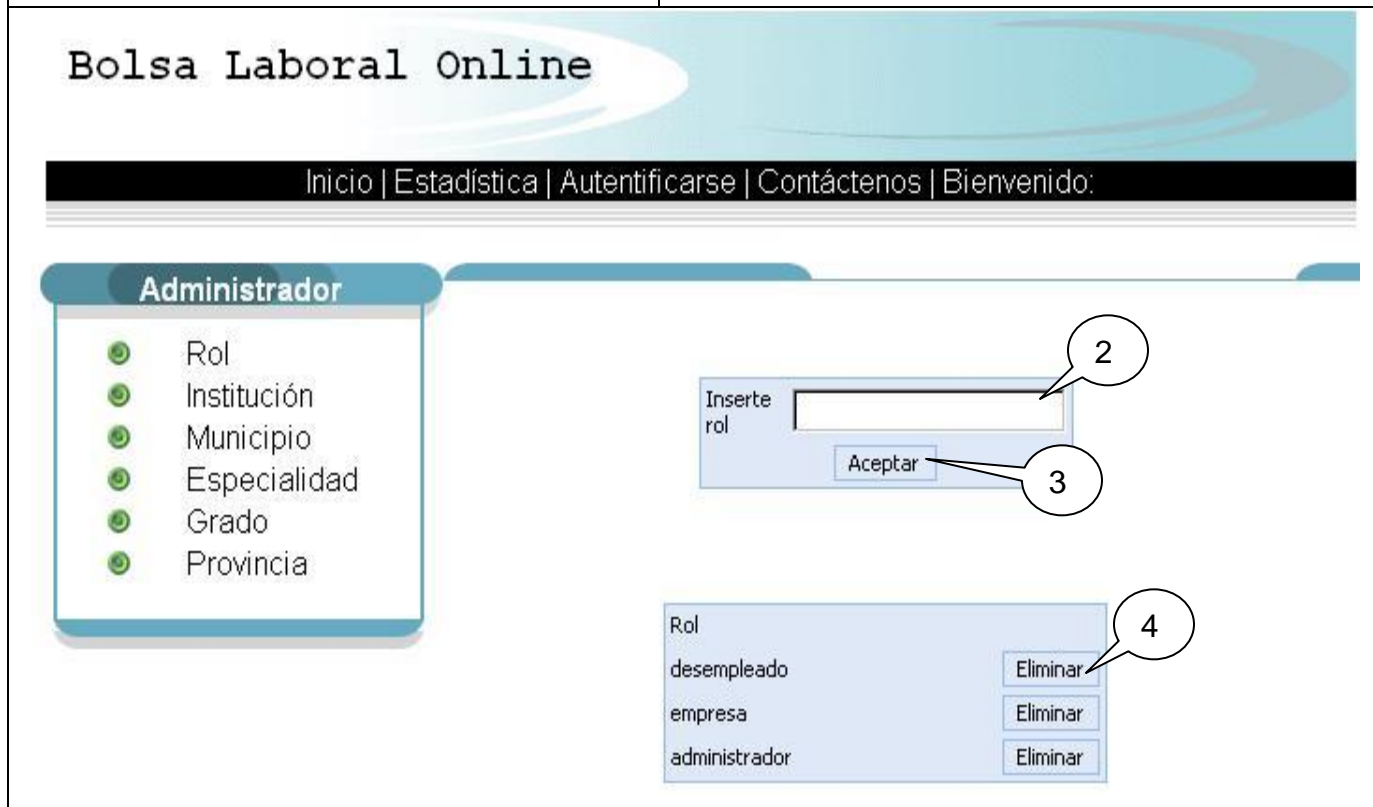


(2) Nombre del municipio	
(3) Nombre de la provincia	
Curso normal de eventos para el caso de uso	
Acción del actor	Respuesta del sistema
1- El administrador del sistema va a la opción "Municipios" (1).	
Sección Insertar municipio	
	1- El sistema muestra los campos a llenar para crear un nuevo municipio (2,3).
2- El administrador introduce los datos pedidos.	
3- El administrador presiona el botón "Aceptar" (4).	4- El sistema actualiza la base de datos con el nuevo municipio creado.
Sección Eliminar municipio	
	1- El sistema muestra una lista con todos los municipios existentes en la base de datos.
2- El administrador presiona el botón "Eliminar" (5).	3- El sistema elimina de la base de datos el municipio seleccionado.
Cursos alternos	

Caso de uso	Actualizar nomenclador rol
Actor(es):	Administrador
Propósito:	Actualizar la lista con los roles.
Resumen:	El caso de uso se inicia cuando el administrador desea insertar o eliminar un rol, actualizando así esta lista.
Precondiciones:	
Poscondiciones:	
Tipo:	Real y expandido.
Responsabilidades:	RF16
Casos de usos relacionados:	
Interfaz I	



Interfaz II

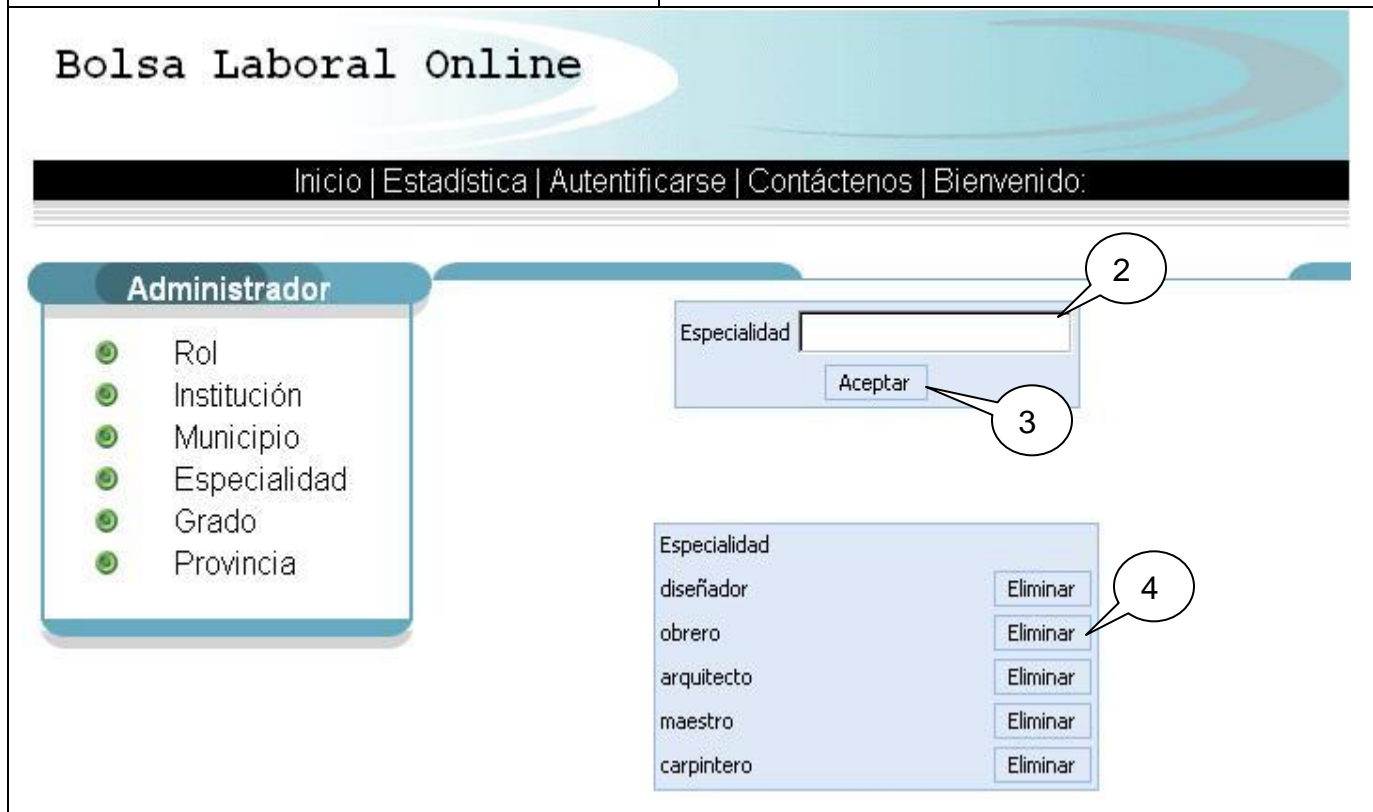


(2) Nombre del rol	
Curso normal de eventos para el caso de uso	
Acción del actor	Respuesta del sistema
1- El administrador del sistema va a la opción "Rol" (1).	
Sección Insertar Rol	
	1- El sistema muestra los campos a llenar para crear un nuevo rol (2).
2- El administrador introduce los datos pedidos.	
3- El administrador presiona el botón "Aceptar" (3).	4- El sistema actualiza la base de datos con el nuevo rol creado.
Sección Eliminar rol	
	1- El sistema muestra una lista con todos los roles existentes en la base de datos.
2- El administrador presiona el botón "Eliminar" (4).	3- El sistema elimina de la base de datos el rol seleccionado.
Cursos alternos	

Caso de uso	Actualizar nomenclador especialidad
Actor(es):	Administrador
Propósito:	Actualizar la lista con las especialidades laborales.
Resumen:	El caso de uso se inicia cuando el administrador desea introducir o eliminar una especialidad laboral, actualizando así esta lista.
Precondiciones:	
Poscondiciones:	
Tipo:	Real y expandido.
Responsabilidades:	RF17
Casos de usos relacionados:	
Interfaz I	

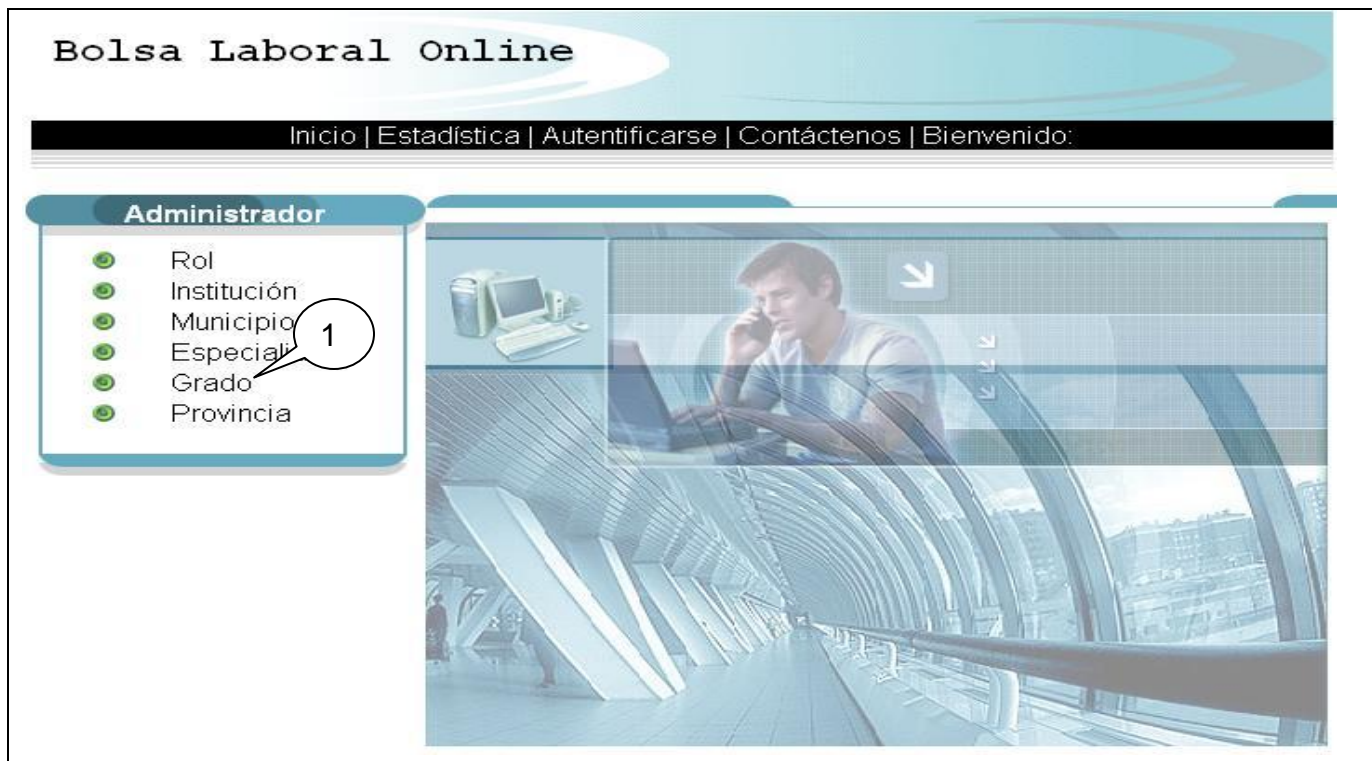


Interfaz II



(2) Nombre de la especialidad laboral	
Curso normal de eventos para el caso de uso	
Acción del actor	Respuesta del sistema
1- El administrador del sistema va a la opción "Especialidad" (1).	
Sección Insertar especialidad	
	1- El sistema muestra los campos a llenar para crear una nueva especialidad (2).
2- El administrador introduce los datos pedidos.	
3- El administrador presiona el botón "Aceptar" (3).	4- El sistema actualiza la base de datos con la nueva especialidad creada.
Sección Eliminar especialidad	
	1- El sistema muestra una lista con todas las especialidades laborales existentes en la base de datos.
2- El administrador presiona el botón "Eliminar" (4).	3- El sistema elimina de la base de datos la especialidad seleccionada.
Cursos alternos	

Caso de uso	Actualizar nomenclador grado
Actor(es):	Administrador
Propósito:	Actualizar la lista con los grados académicos.
Resumen:	El caso de uso se inicia cuando el administrador desea introducir o eliminar un grado académico, actualizando así esta lista.
Precondiciones:	
Poscondiciones:	
Tipo:	Real y expandido.
Responsabilidades:	RF18
Casos de usos relacionados:	
Interfaz I	



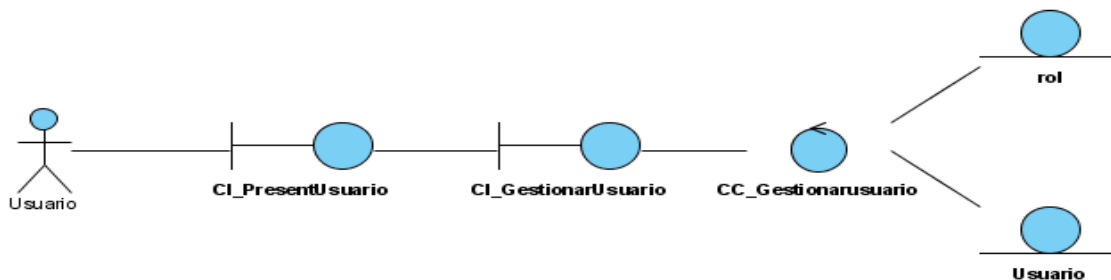
Interfaz II



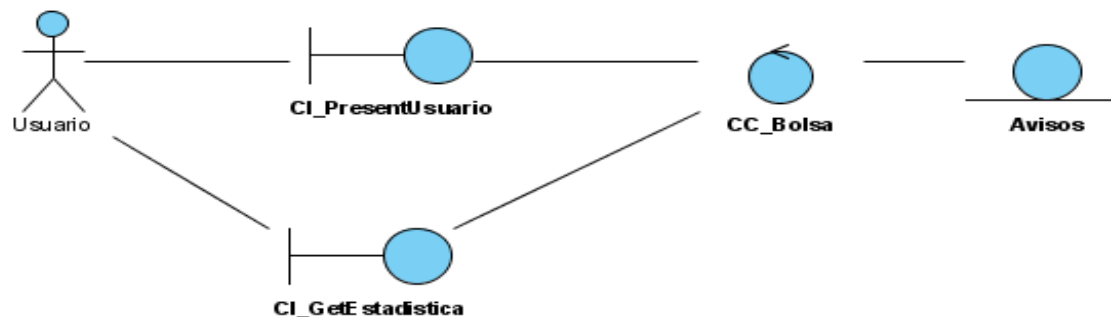
(2) Nombre del grado de escolaridad	
Curso normal de eventos para el caso de uso	
Acción del actor	Respuesta del sistema
1- El administrador del sistema va a la opción "Grado" (1).	
Sección Insertar especialidad	
	1- El sistema muestra los campos a llenar para crear un nuevo grado (2).
2- El administrador introduce los datos pedidos.	
3- El administrador presiona el botón "Aceptar" (3).	4- El sistema actualiza la base de datos con el nuevo grado creado.
Sección Eliminar especialidad	
	1- El sistema muestra una lista con todos los grados académicos existentes en la base de datos.
2- El administrador presiona el botón "Eliminar" (4).	3- El sistema elimina de la base de datos el grado académico seleccionado.
Cursos alternos	

Anexo 2

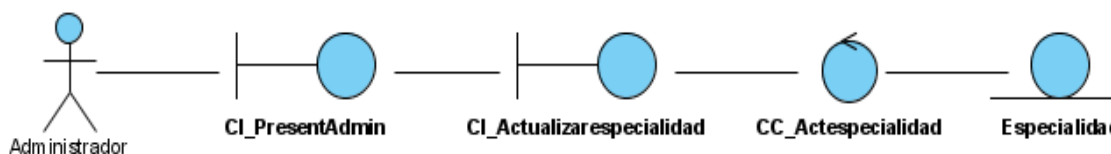
Gestionar registro de usuario



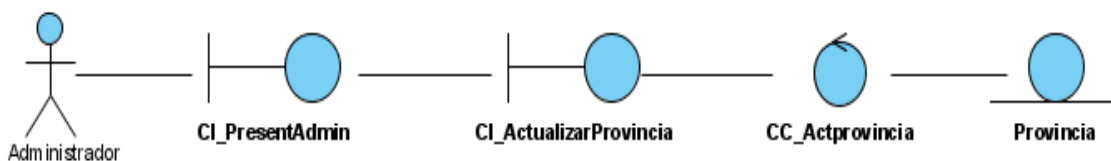
Generar estadística



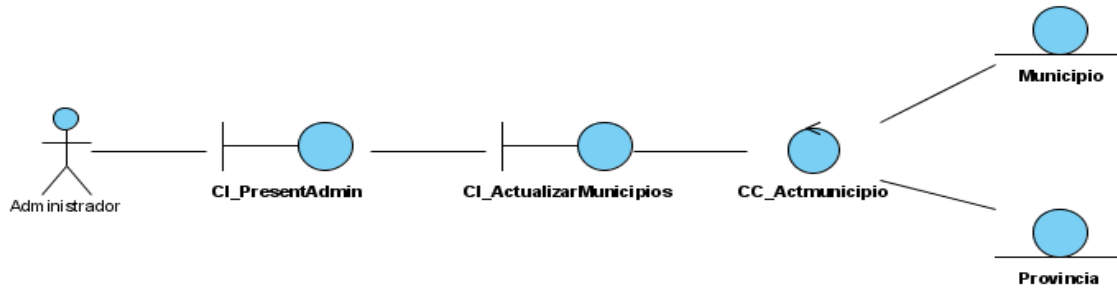
Actualizar nomenclador especialidad



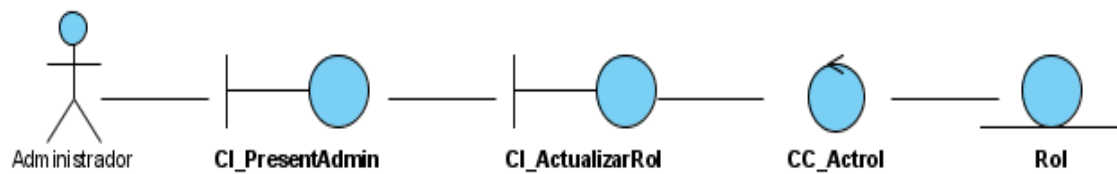
Actualizar nomenclador provincia



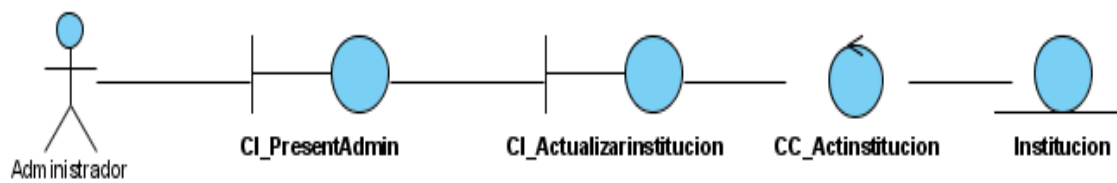
Actualizar nomenclador municipios



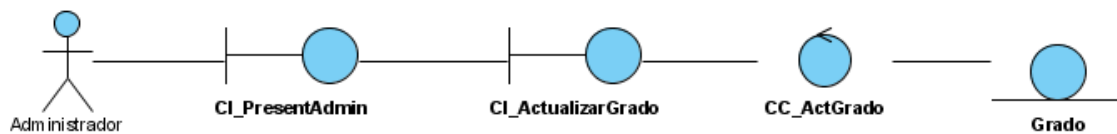
Actualizar nomenclador rol



Actualizar nomenclador institución

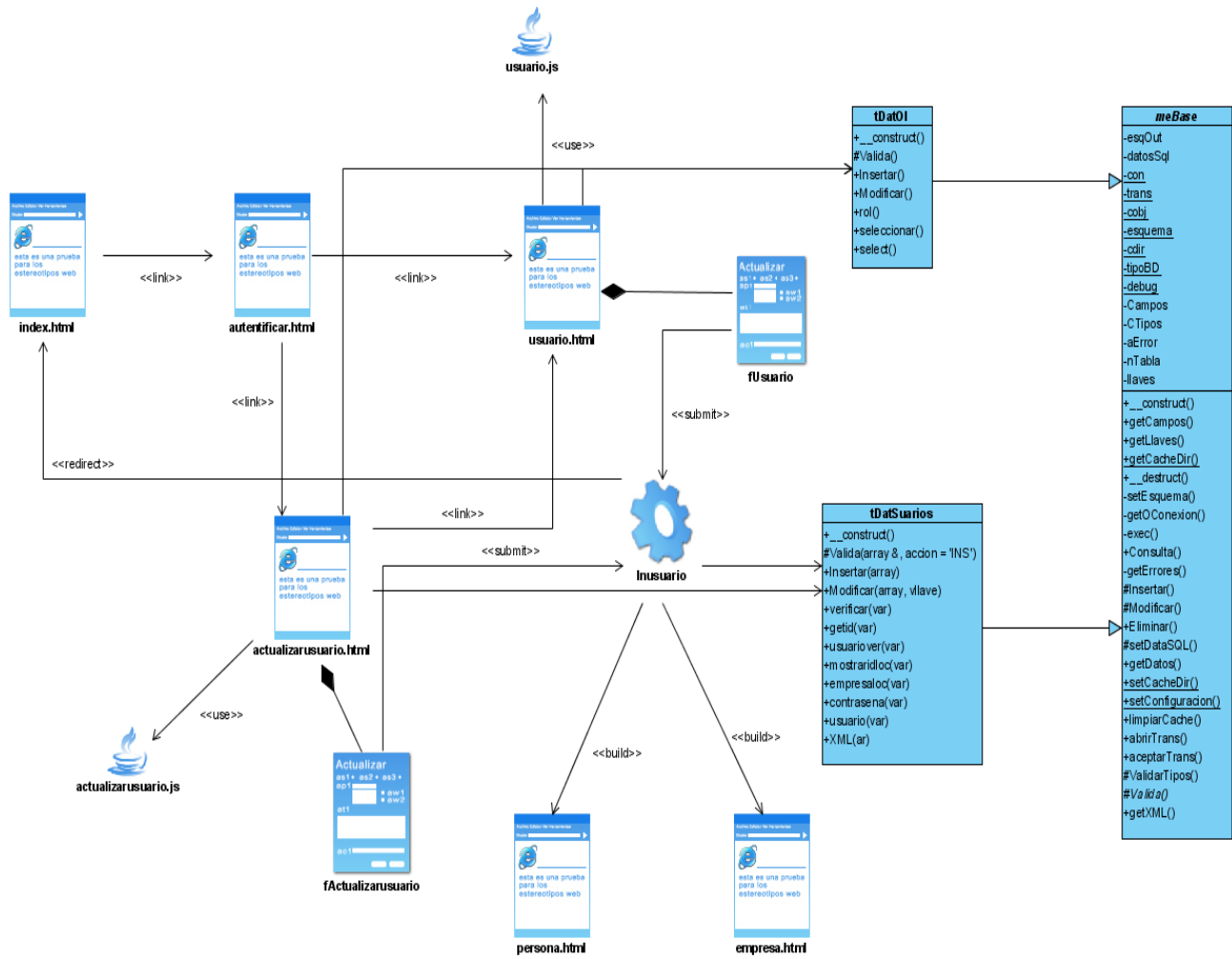


Actualizar nomenclador grado

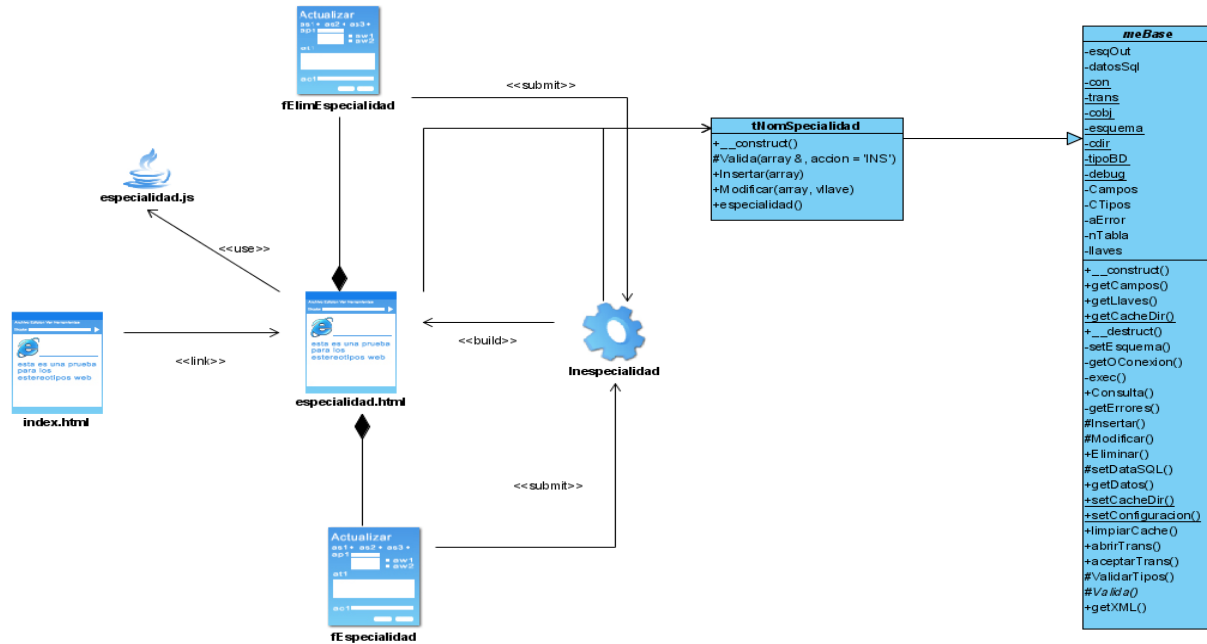


Anexo 3

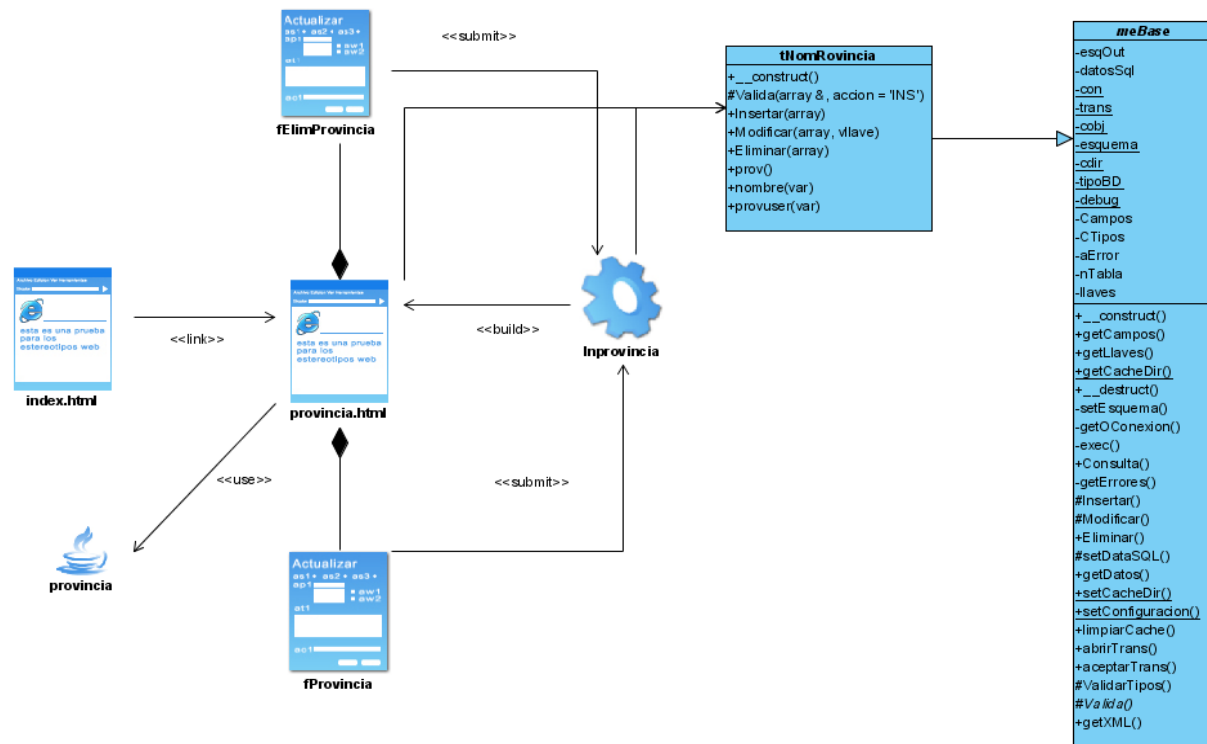
Gestionar registro de usuario



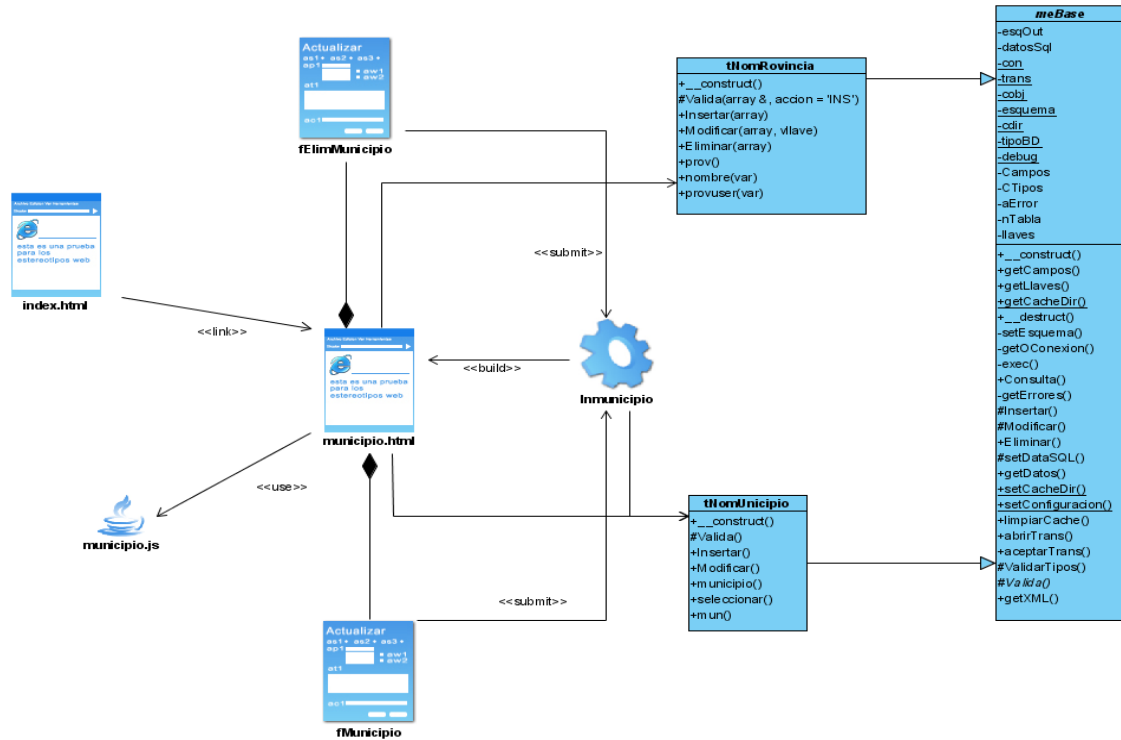
Actualizar nomenclador especialidad



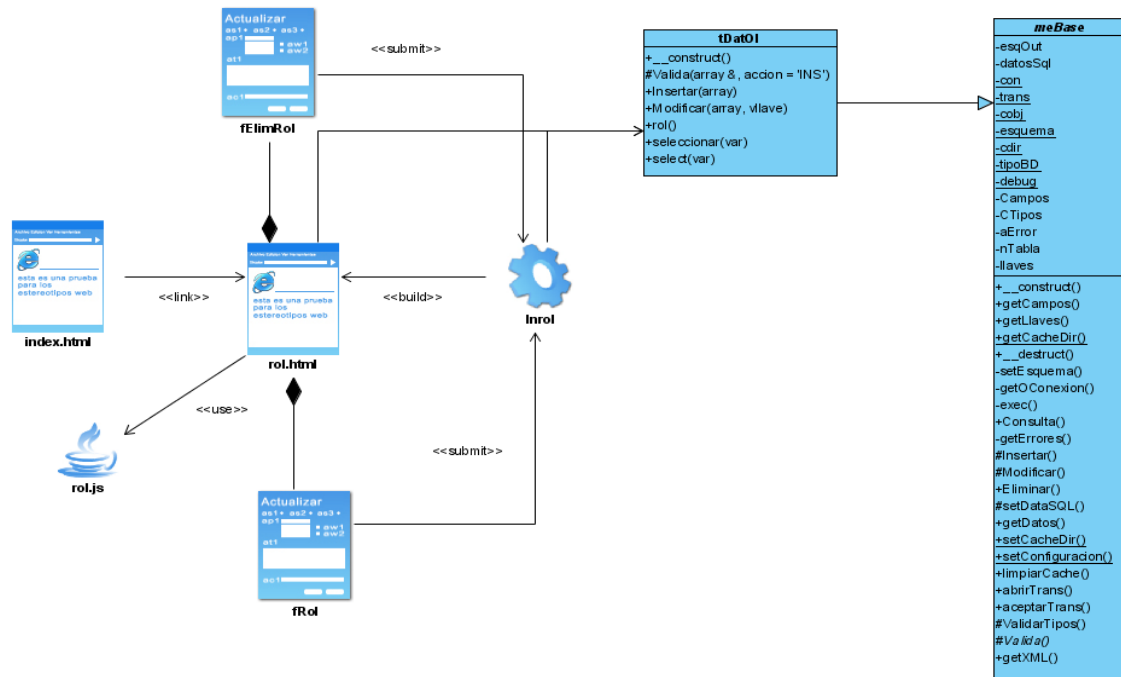
Actualizar nomenclador provincia



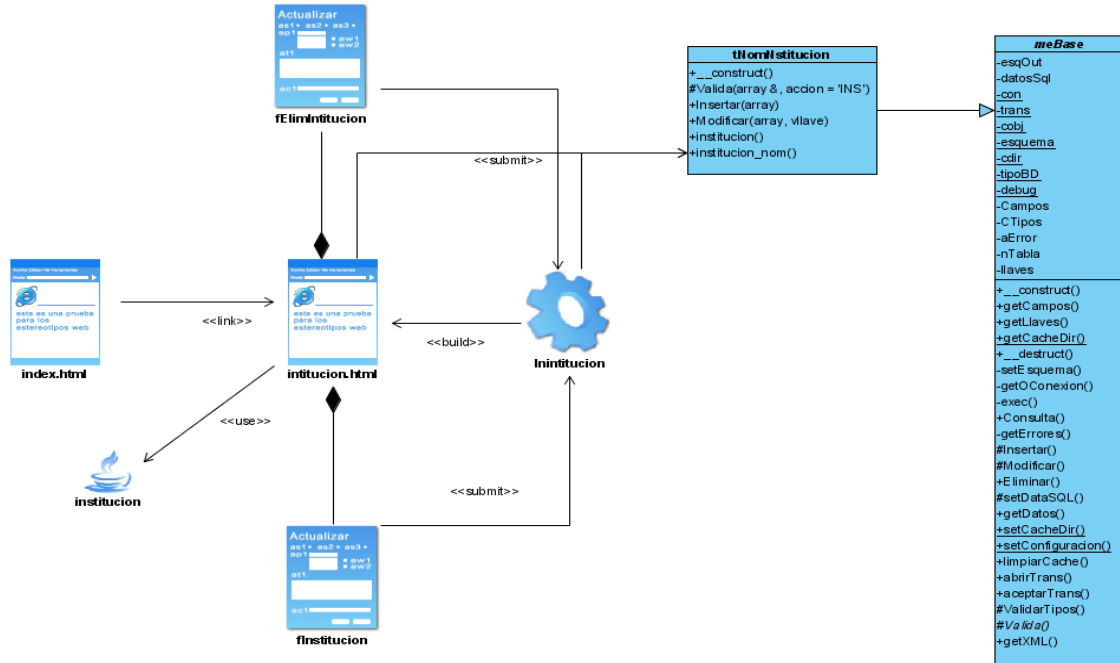
Actualizar nomenclador municipio



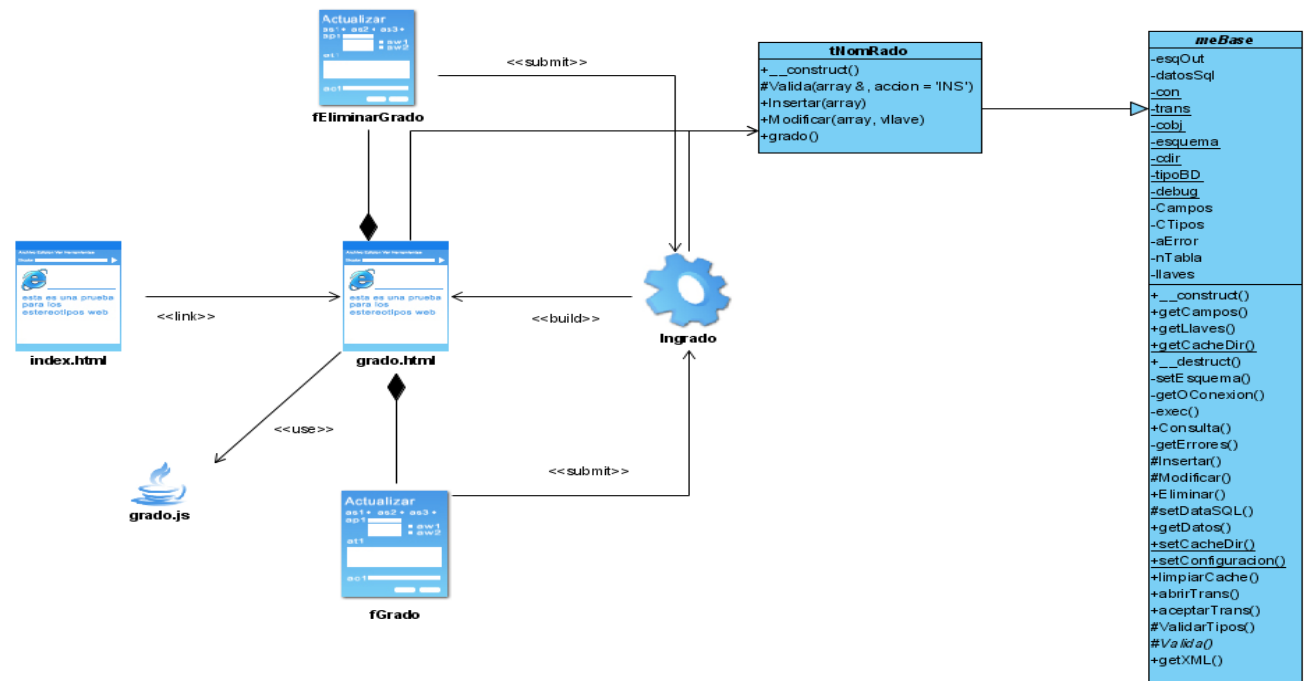
Actualizar nomenclador rol



Actualizar nomenclador institución

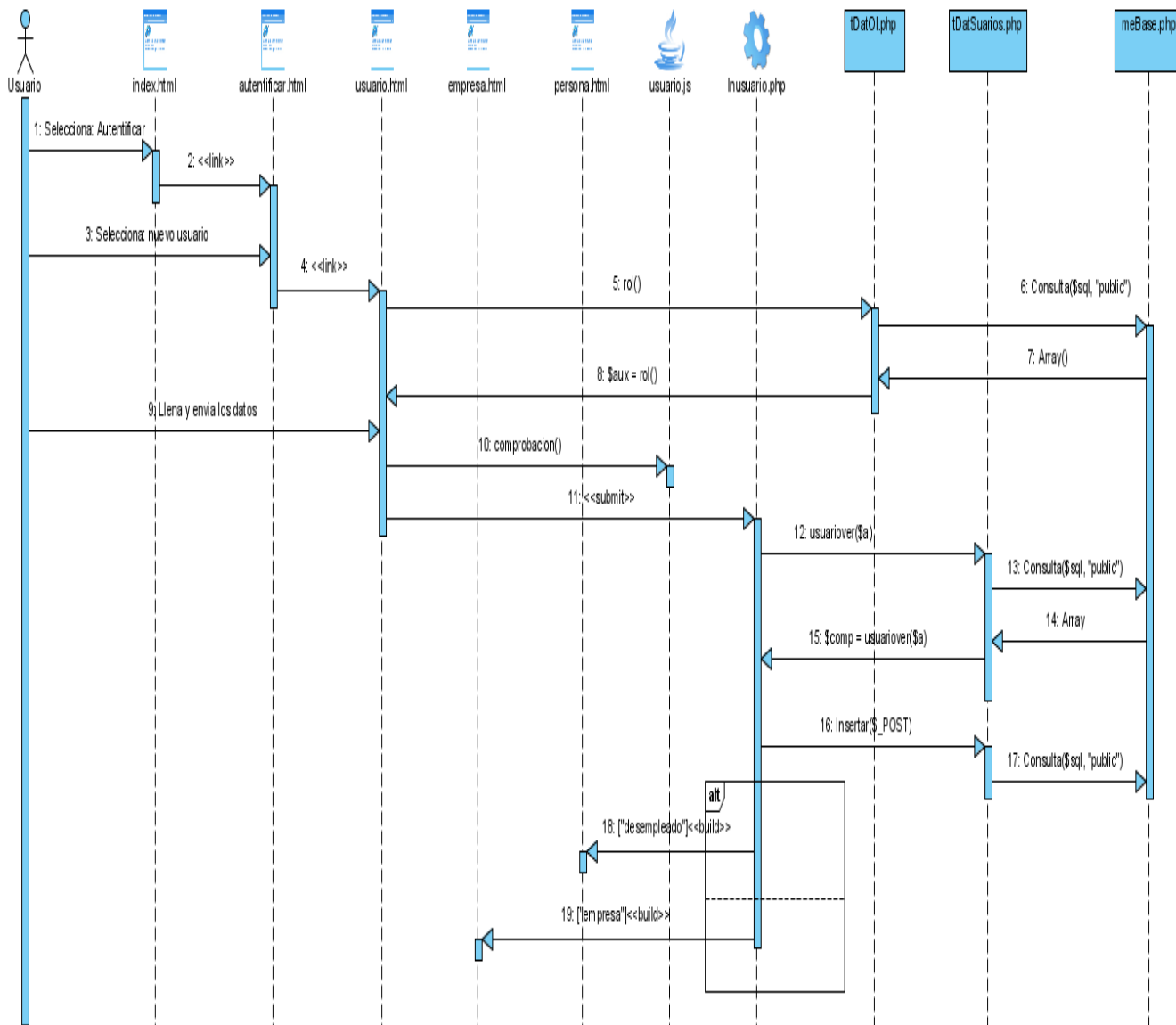


Actualizar nomenclador grado

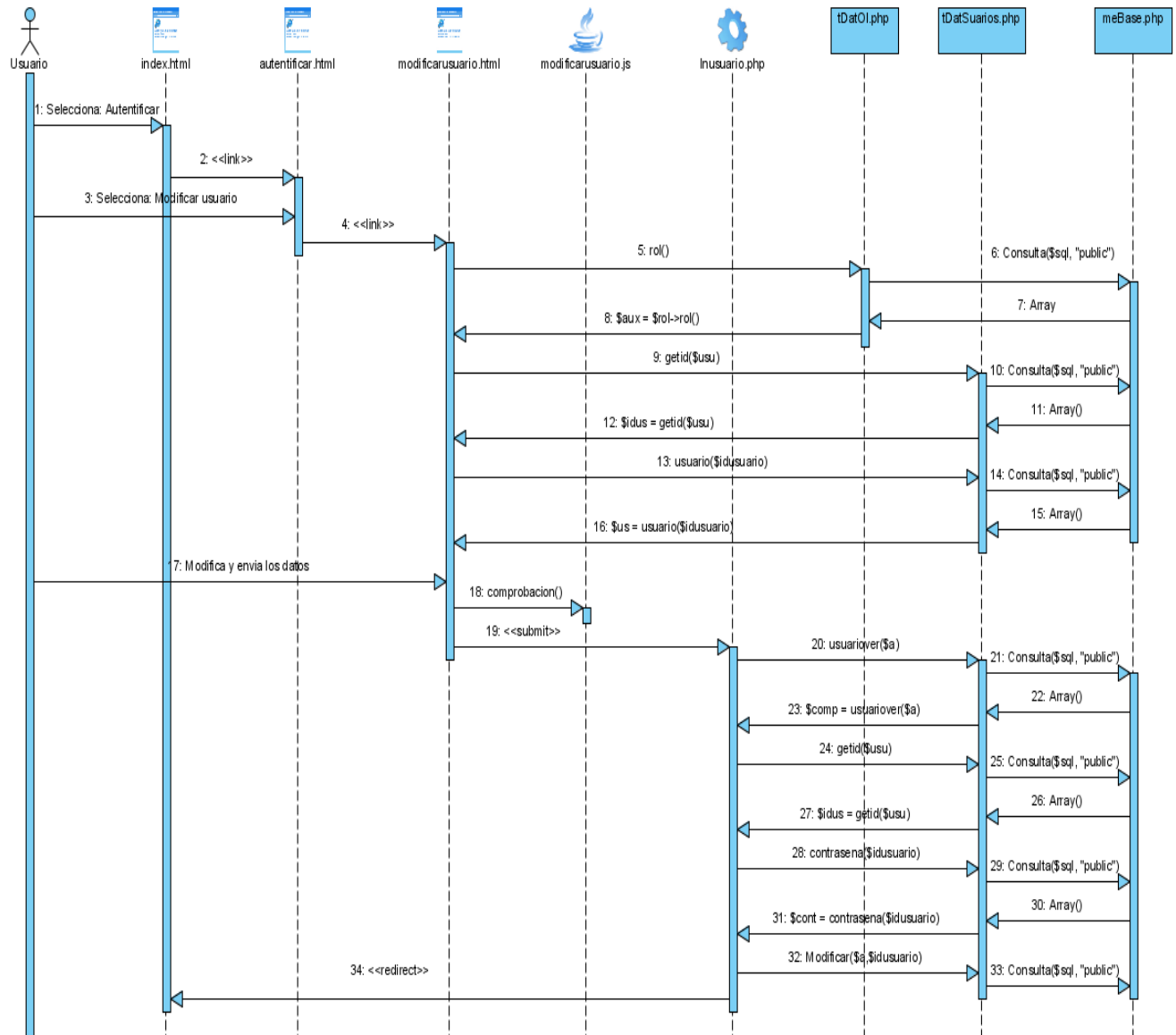


Anexo 4

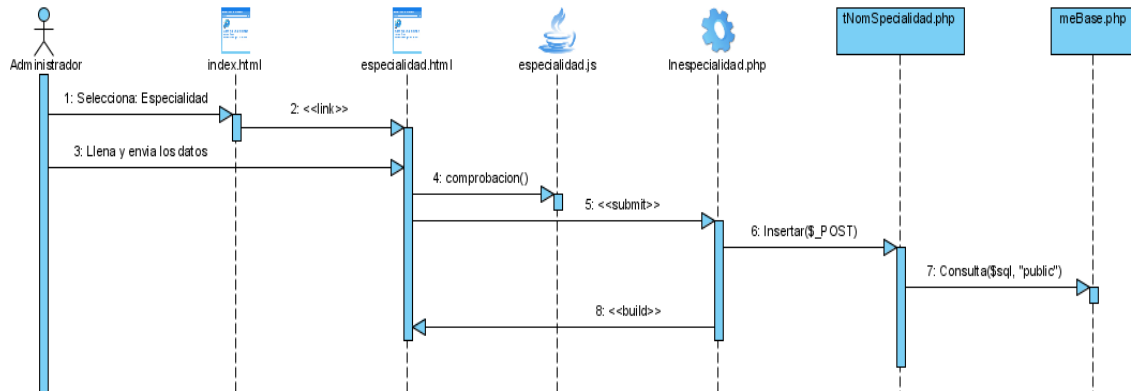
Gestionar registro del usuario (Insertar)



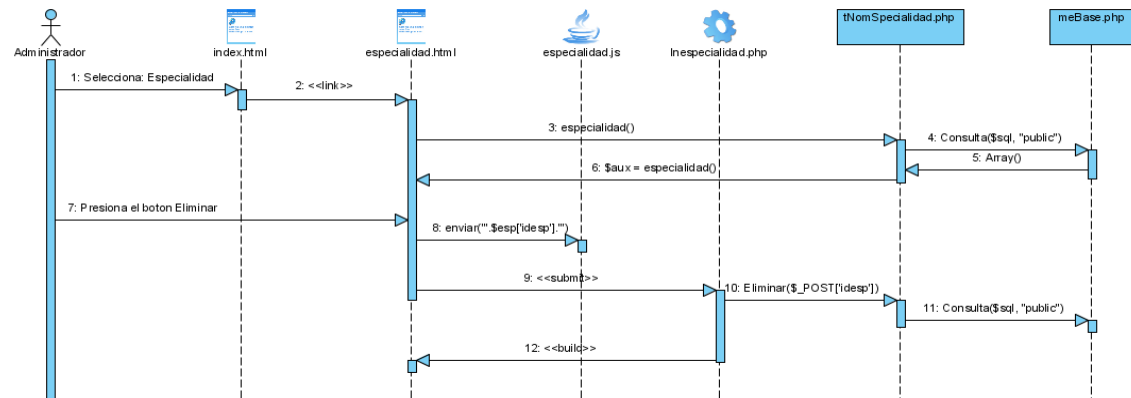
Gestionar registro del usuario (Modificar)



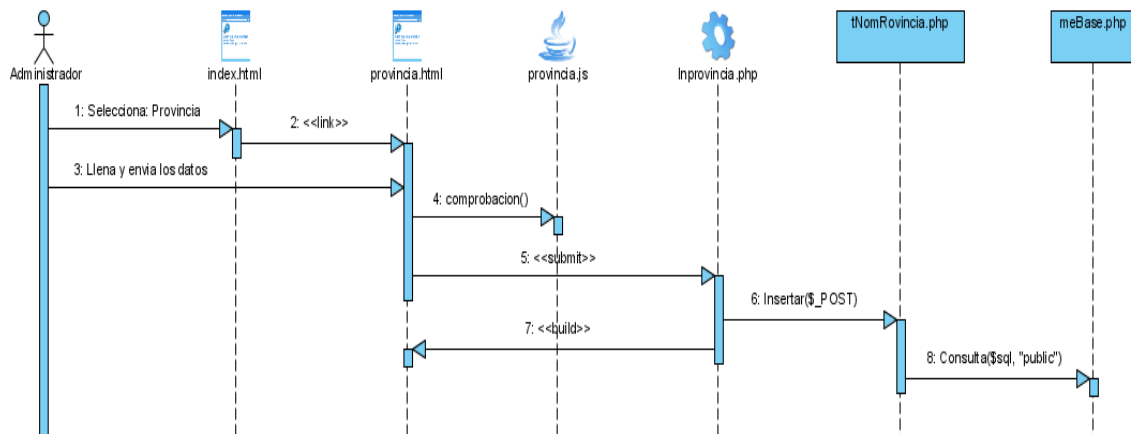
Actualizar nomenclador especialidad (Insertar)



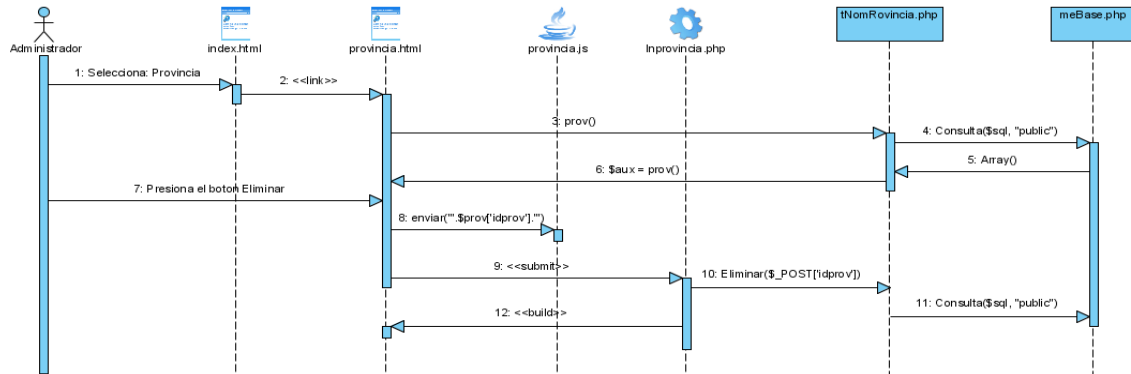
Actualizar nomenclador especialidad (Eliminar)



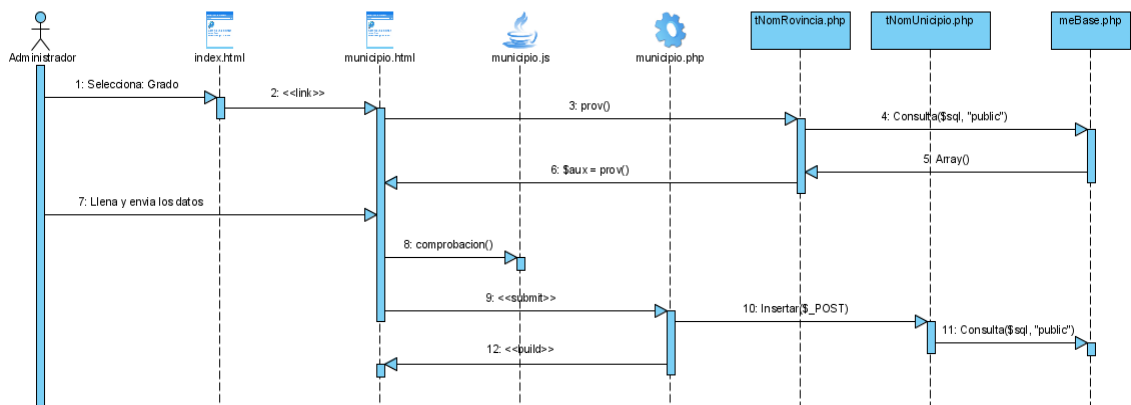
Actualizar nomenclador provincia (Insertar)



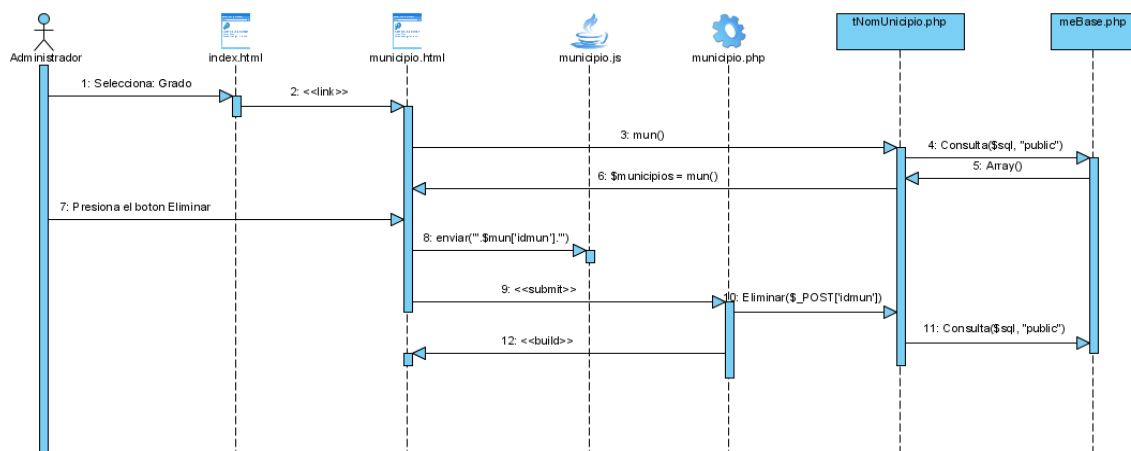
Actualizar nomenclador provincia (Eliminar)



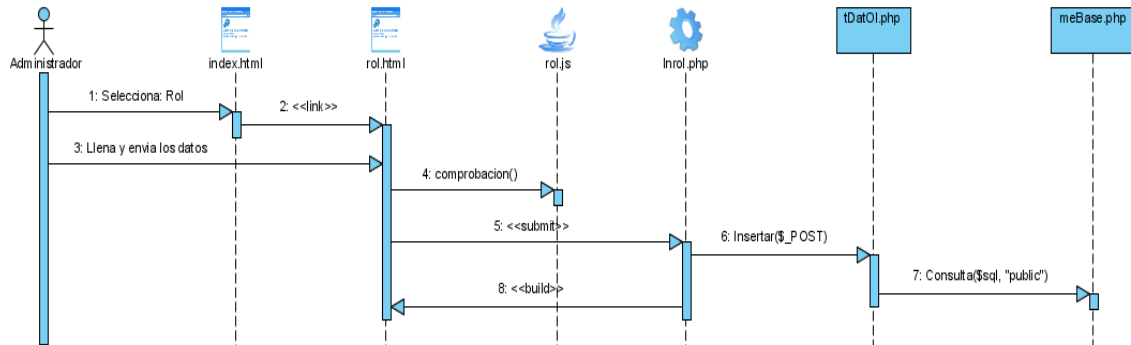
Actualizar nomenclador municipio (Insertar)



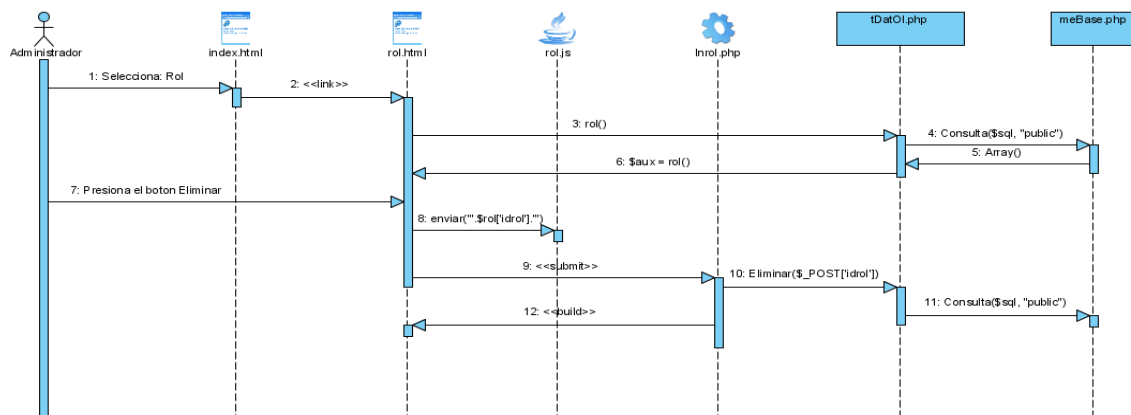
Actualizar nomenclador municipio (Eliminar)



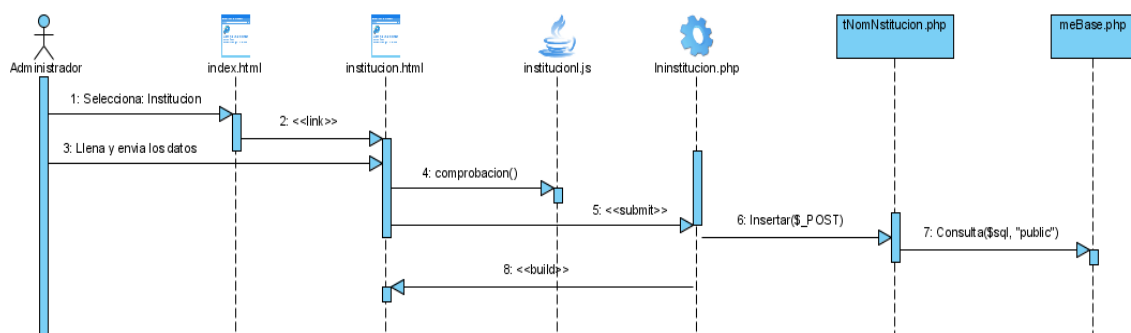
Actualizar nomenclador rol (Insertar)



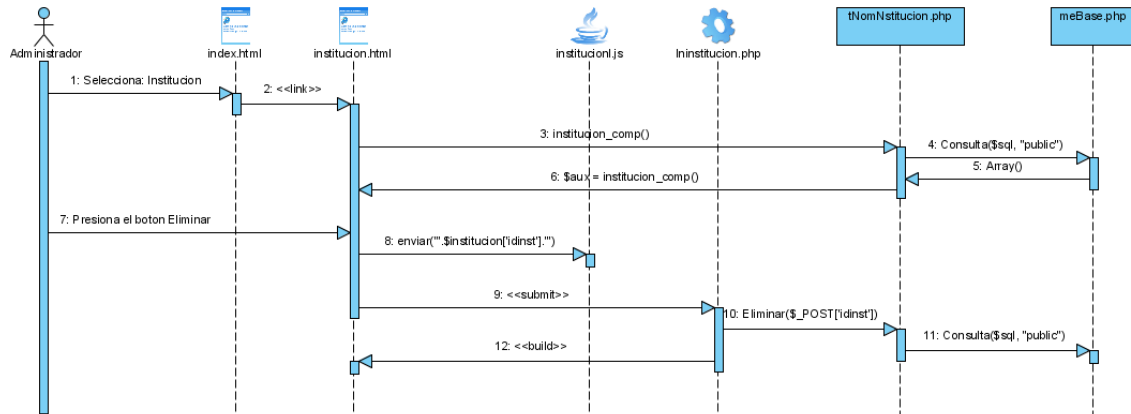
Actualizar nomenclador rol (Eliminar)



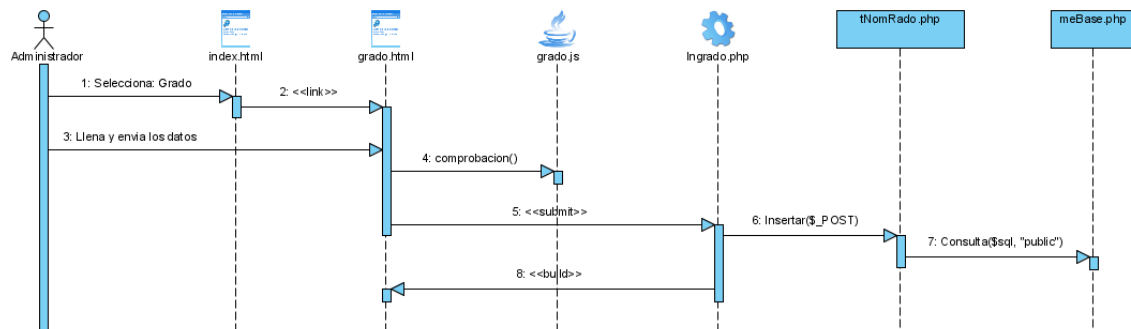
Actualizar nomenclador institución (Insertar)



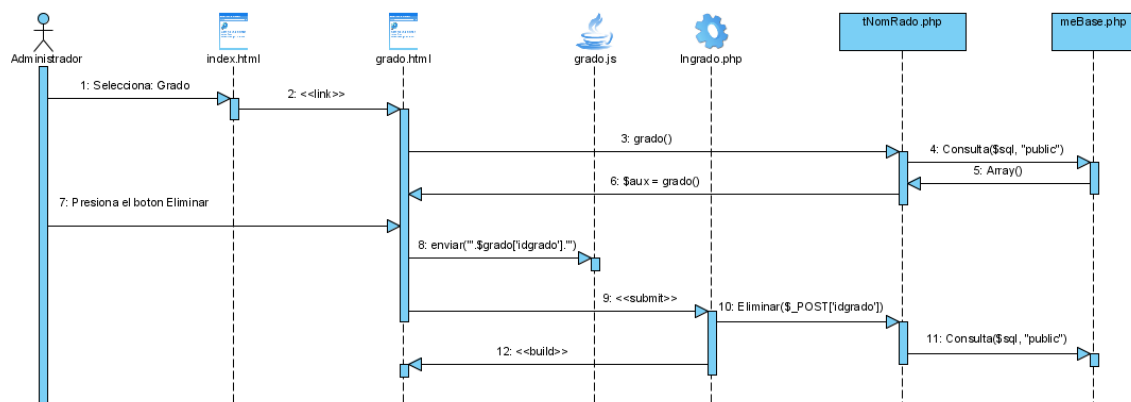
Actualizar nomenclador institución (Eliminar)



Actualizar nomenclador grado (Insertar)



Actualizar nomenclador grado (Eliminar)



Anexo 5

Nombre: nomunicipio		
Descripción: Dirección de la persona o empresa.		
Atributo	Tipo	Descripción
idmun	integer	Identificador del municipio.
nombre	varchar(50)	Nombre del municipio.
idprov	integer	Identificador de la provincia al cual pertenece el municipio.

Nombre: nomprovincia		
Descripción: Dirección de la persona o empresa.		
Atributo	Tipo	Descripción
idprov	integer	Identificador de la provincia.
nombre	varchar(20)	Nombre de la provincia.

Nombre: nomgrado		
Descripción: Dirección de la persona o empresa.		
Atributo	Tipo	Descripción
idgrado	integer	Identificador del grado de escolaridad.
nombre	varchar(20)	Nombre del grado de escolaridad.

Nombre: nominstitucion		
Descripción: Dirección de la persona o empresa.		
Atributo	Tipo	Descripción
idinst	integer	Identificador de la institución.
institucion	varchar(20)	Tipo de institución.
nombre	varchar(50)	Nombre de la institución.

Nombre: nomempresa		
Descripción: Datos de la empresa.		
Atributo	Tipo	Descripción
idempresa	integer	Identificador de la empresa.
nombre	varchar(100)	Nombre de la empresa.
idloc	integer	Identificador de la dirección de la empresa.
idusuario	integer	Identificador del usuario que representa la empresa.

Nombre: nomespecialidad		
Descripción: Datos de la especialidad.		
Atributo	Tipo	Descripción
idesp	integer	Identificador de la especialidad.
especialidad	varchar(50)	Nombre de la especialidad.

Nombre: datrol		
Descripción: Datos del rol.		
Atributo	Tipo	Descripción
idrol	integer	Identificador del rol.
tipo	varchar(50)	Tipo de rol.

Nombre: datusuarios		
Descripción: Datos del usuario.		
Atributo	Tipo	Descripción
idusuario	integer	Identificador del usuario.
usuario	varchar(50)	Nombre del usuario.
contrasena	varchar(256)	Contraseña del usuario.
idrol	integer	Identificador del rol del usuario.

Anexo 6

Nombre: tDatOI	
Tipo de clase: Acceso a datos	
Atributo	Tipo
Responsabilidades:	
Nombre:	__construct()
Descripción:	Método que crea un objeto de tipo rol.
Nombre:	Valida(& \$array, \$accion='INS')
Descripción:	Valida que la entrada de datos a una consulta sea la correcta.
Nombre:	Insertar(\$array)
Descripción:	Envía el arreglo entrado por parámetro a la clase meBase para insertar los elementos de este en la base de datos.
Nombre:	rol()
Descripción:	Obtiene todos los roles existentes.
Nombre:	seleccionar(\$var)
Descripción:	Obtiene el tipo de rol dado el identificador del rol pasado por parámetro.
Nombre:	select(\$var)
Descripción:	Obtiene el tipo de rol dado el identificador de un usuario pasado por parámetro.

Nombre: tDatSuarios	
Tipo de clase: Acceso a datos	
Atributo	Tipo
Responsabilidades:	
Nombre:	__construct()
Descripción:	Método que crea un objeto de tipo usuario.
Nombre:	Valida(& \$array, \$accion='INS')
Descripción:	Valida que la entrada de datos a una consulta sea la correcta.
Nombre:	Insertar(\$array)
Descripción:	Envía el arreglo entrado por parámetro a la clase meBase para insertar los elementos

	de este en la base de datos.
Nombre:	verificar(\$var)
Descripción:	Obtiene el usuario y la contraseña pasándole como parámetro al usuario.
Nombre:	getid(\$var)
Descripción:	Obtiene el identificador del usuario pasándole como parámetro el usuario.
Nombre:	usuariover(\$var)
Descripción:	Obtiene el usuario pasándole el mismo usuario como parámetro.

Nombre: tNomMpresa	
Tipo de clase: Acceso a datos	
Atributo	Tipo
Responsabilidades	
Nombre:	__construct()
Descripción:	Método que crea un objeto de tipo empresa.
Nombre:	Valida(& \$array, \$accion='INS')
Descripción:	Valida que la entrada de datos a una consulta sea la correcta.
Nombre:	Insertar(\$array)
Descripción:	Envía el arreglo entrado por parámetro a la clase meBase para insertar los elementos de este en la base de datos.
Nombre:	idempresa(\$var)
Descripción:	Obtiene el identificador de la empresa pasándole como parámetro el identificador del usuario de esa empresa.

Nombre: tNomNstitucion	
Tipo de clase: Acceso a datos	
Atributo	Tipo
Responsabilidades:	
Nombre:	__construct()
Descripción:	Método que crea un objeto de tipo institucion.

Nombre:	Valida(& \$array, \$accion='INS')
Descripción:	Valida que la entrada de datos a una consulta sea la correcta.
Nombre:	Insertar(\$array)
Descripción:	Envía el arreglo entrado por parámetro a la clase meBase para insertar los elementos de este en la base de datos.
Nombre:	institucion()
Descripción:	Obtiene una lista con todas las instituciones existentes en la base de datos.
Nombre:	institucion_nom()
Descripción:	Obtiene una lista con el identificador y el nombre de todas las instituciones existentes en la base de datos.

Nombre: tNomRado	
Tipo de clase: Acceso a datos	
Atributo	Tipo
Responsabilidades:	
Nombre:	__construct()
Descripción:	Método que crea un objeto de tipo gradogrado.
Nombre:	Valida(& \$array, \$accion='INS')
Descripción:	Valida que la entrada de datos a una consulta sea la correcta.
Nombre:	Insertar(\$array)
Descripción:	Envía el arreglo entrado por parámetro a la clase meBase para insertar los elementos de este en la base de datos.
Nombre:	grado()
Descripción:	Obtiene una lista con todos los grados existentes en la base de datos.

Nombre: tNomRovincia	
Tipo de clase: Acceso a datos	
Atributo	Tipo
Responsabilidades:	

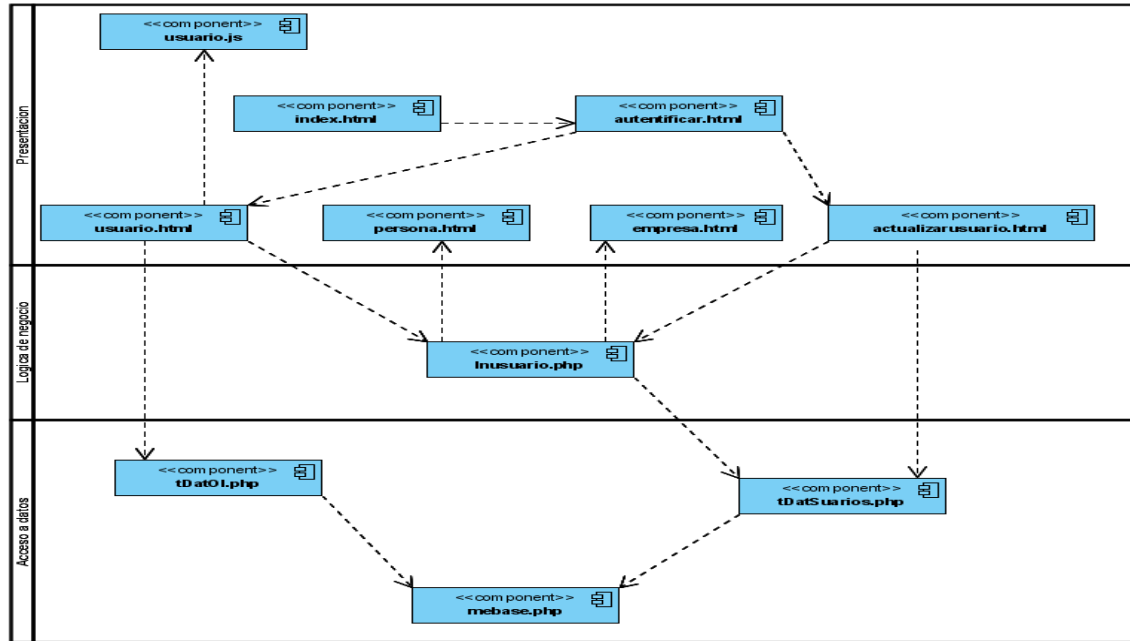
Nombre:	__construct()
Descripción:	Método que crea un objeto de tipo provincia.
Nombre:	Valida(& \$array, \$accion='INS')
Descripción:	Valida que la entrada de datos a una consulta sea la correcta.
Nombre:	Insertar(\$array)
Descripción:	Envía el arreglo entrado por parámetro a la clase meBase para insertar los elementos de este en la base de datos.
Nombre:	prov()
Descripción:	Obtiene una lista con todas las provincias existentes en la base de datos.
Nombre:	nombre(\$var)
Descripción:	Obtiene el nombre de la provincia pasándole como parámetro el identificador de la provincia.
Nombre:	provuser(\$var)
Descripción:	Obtiene el identificador de la provincia de un usuario determinado pasándole como parámetro el identificador del usuario.

Nombre: tNomSpecialidad	
Tipo de clase: Acceso a datos	
Atributo	Tipo
Responsabilidades:	
Nombre:	__construct()
Descripción:	Método que crea un objeto de tipo especialidad.
Nombre:	Valida(& \$array, \$accion='INS')
Descripción:	Valida que la entrada de datos a una consulta sea la correcta.
Nombre:	Insertar(\$array)
Descripción:	Envía el arreglo entrado por parámetro a la clase meBase para insertar los elementos de este en la base de datos.
Nombre:	especialidad()
Descripción:	Obtiene una lista de las especialidades existentes en la base de datos.

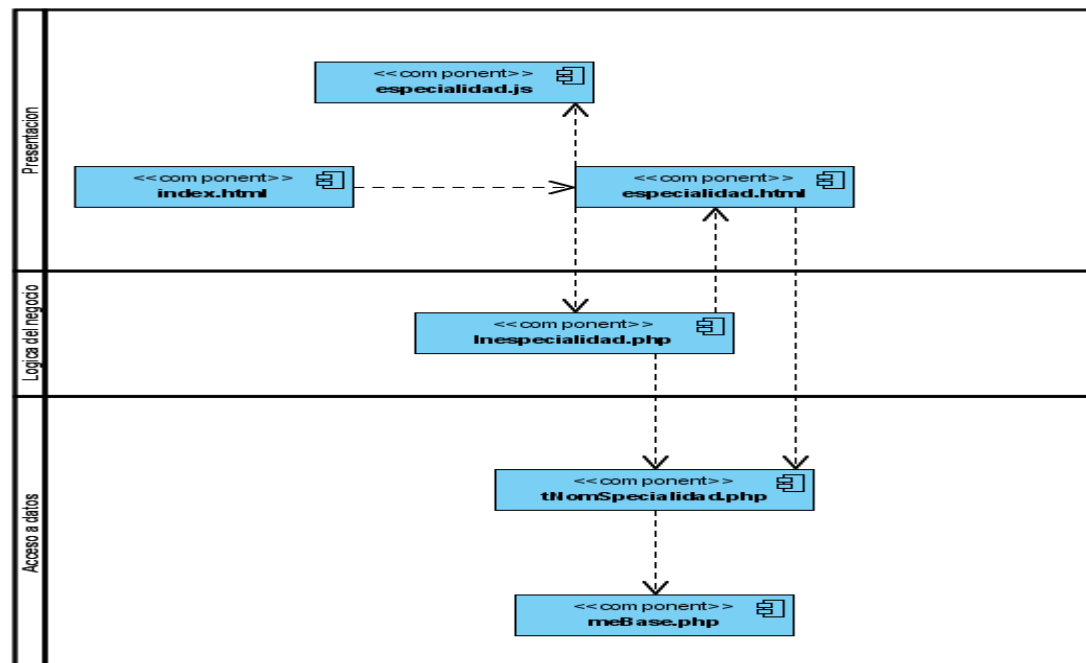
Nombre: tNomUnicipio	
Tipo de clase: Acceso a datos	
Atributo	Tipo
Responsabilidades:	
Nombre:	__construct()
Descripción:	Método que crea un objeto de tipo municipio.
Nombre:	Valida(& \$array, \$accion='INS')
Descripción:	Valida que la entrada de datos a una consulta sea la correcta.
Nombre:	Insertar(\$array)
Descripción:	Envía el arreglo entrado por parámetro a la clase meBase para insertar los elementos de este en la base de datos.
Nombre:	municipio()
Descripción:	Obtiene una lista de todos los municipios, identificadores de estos municipios y la provincia al cual pertenecen estos municipios.
Nombre:	seleccionar(\$var)
Descripción:	Obtiene una lista de municipios y sus identificadores pasándole como parámetro el identificador de la provincia a la cual pertenecen estos municipios.
Nombre:	mun()
Descripción:	Obtiene una lista con el nombre y el identificador de cada municipio existentes en la base de datos.

Anexo 7

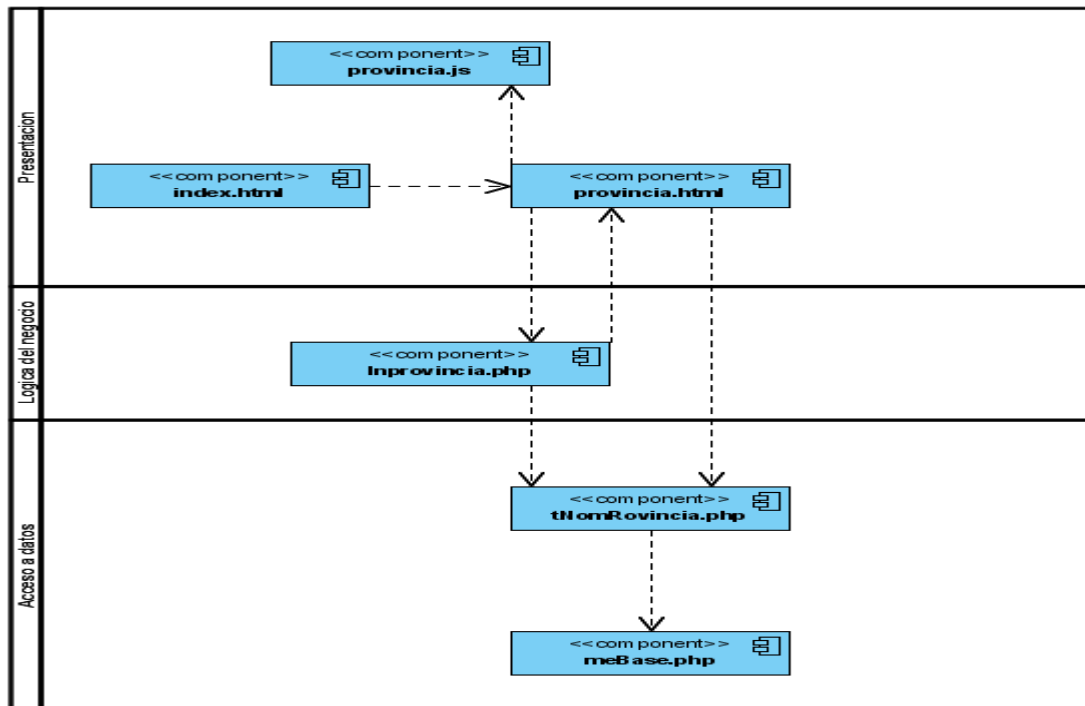
Gestionar registro del usuario



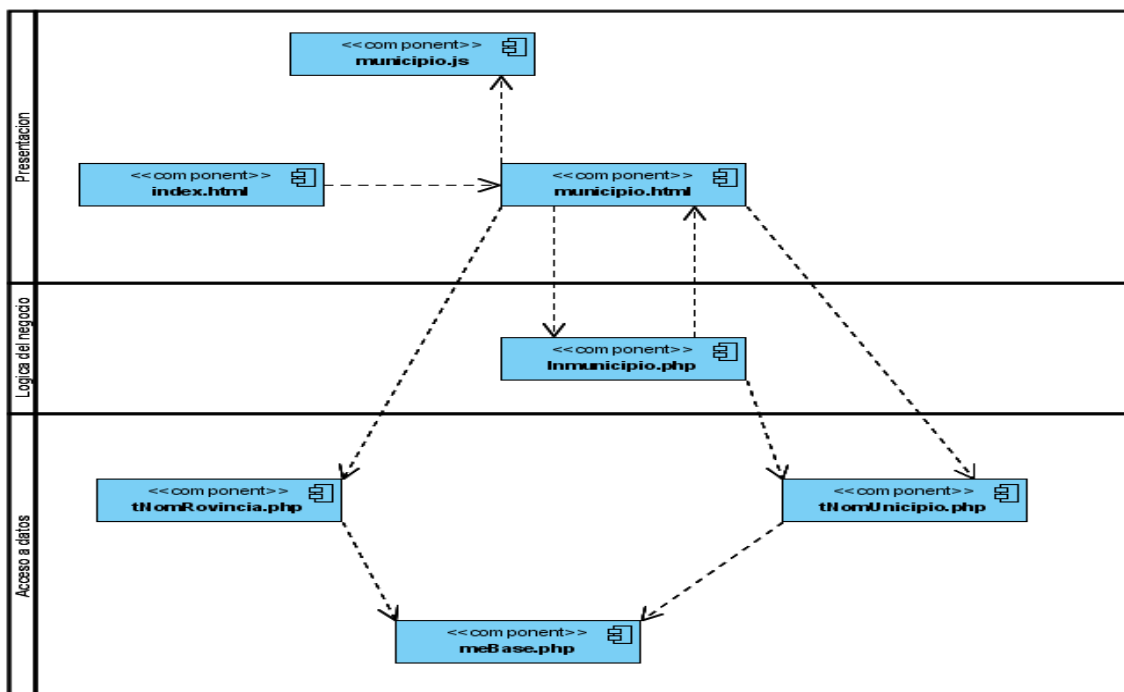
Actualizar nomenclador especialidad



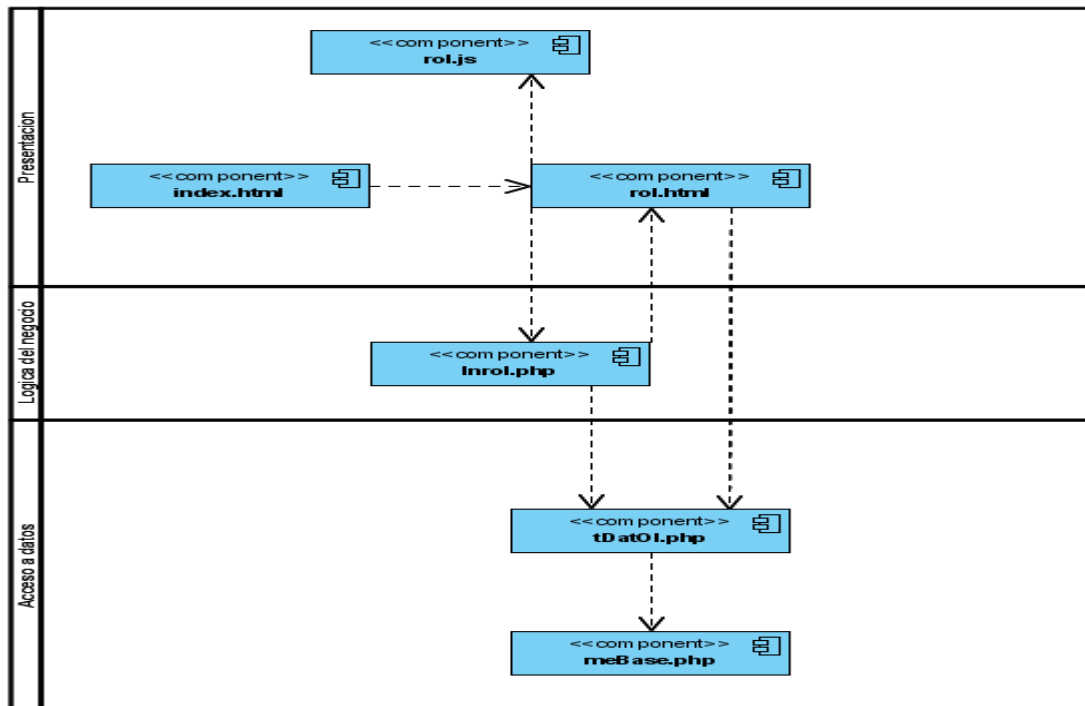
Actualizar nomenclador provincia



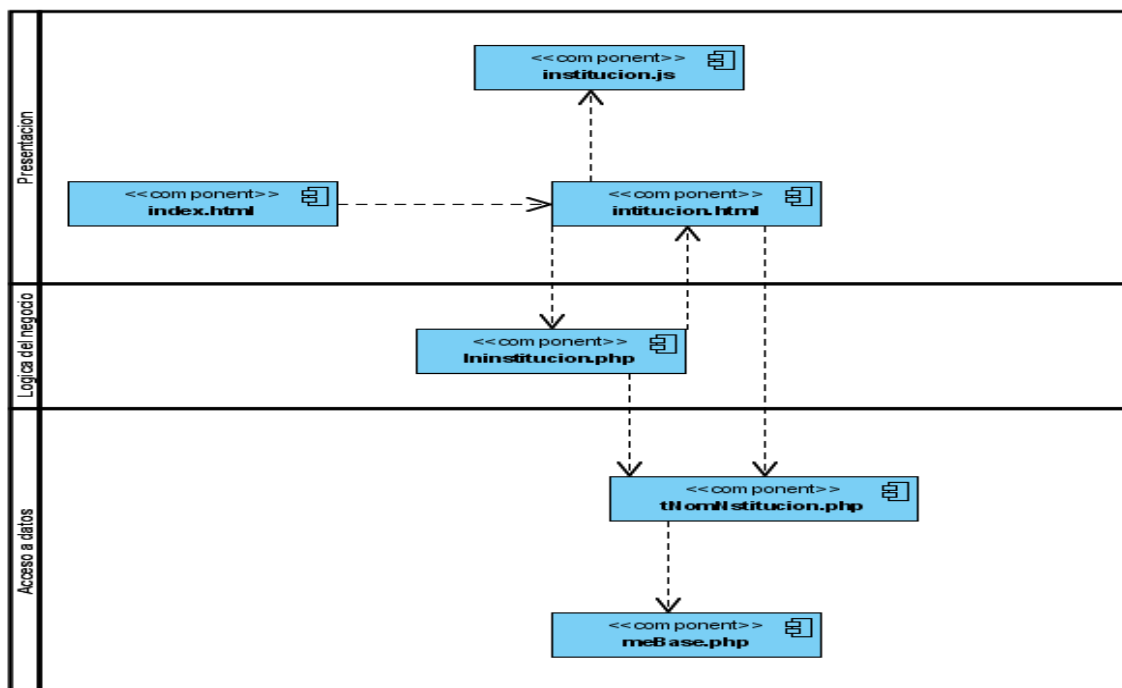
Actualizar nomenclador municipio



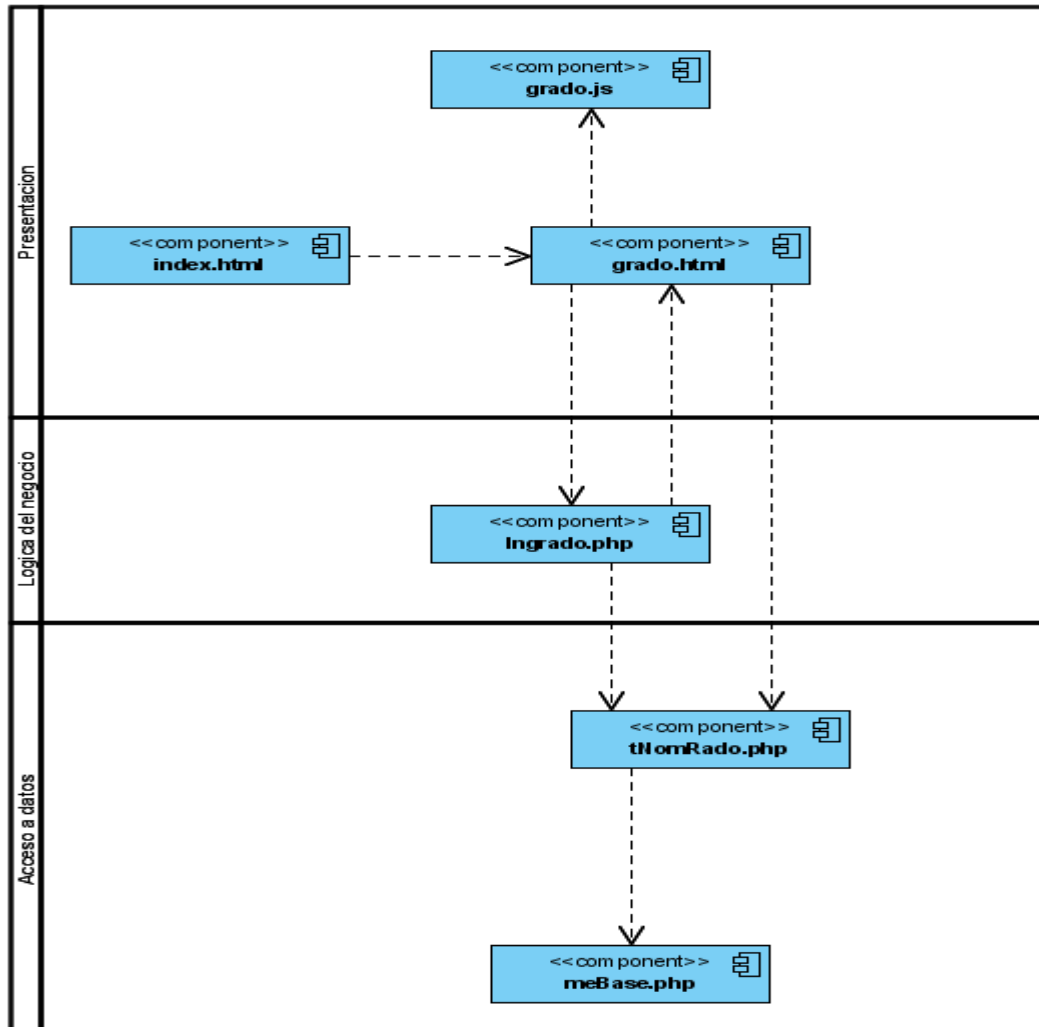
Actualizar nomenclador rol



Actualizar nomenclador institución



Actualizar nomenclador grado



Anexo 8

CU Gestionar registro del usuario (nuevo).		
Entrada	Resultados	Condiciones
Usuario: "reynold" Contraseña: "reynold" Rol: "desempleado"	El sistema creo un nuevo usuario, con los datos especificados.	

CU Gestionar registro del usuario (modificar).		
Entrada	Resultados	Condiciones
		El usuario debe de estar registrado y autenticado en el sitio.
Usuario: "prueba" Contraseña: "reynold" Nueva Contraseña: "prueba" Rep. Contraseña: "prueba" Rol: "empresa"	El sistema modificó correctamente los datos del usuario.	

CU Actualizar nomenclador provincia (Insertar).		
Entrada	Resultados	Condiciones
Provincia: "desempleado"	Provincia insertada satisfactoriamente.	

CU Actualizar nomenclador provincia (Eliminar).		
Entrada	Resultados	Condiciones
Botón: "Eliminar"	Provincia eliminada satisfactoriamente.	

CU Actualizar nomenclador rol (Insertar).		
Entrada	Resultados	Condiciones
Inserte rol: "desempleado"	Rol insertado satisfactoriamente.	

CU Actualizar nomenclador rol (Eliminar).		
Entrada	Resultados	Condiciones
Botón: "Eliminar"	Rol eliminado satisfactoriamente.	

CU Actualizar nomenclador grado (Insertar).		
Entrada	Resultados	Condiciones
Grado de escolaridad: "12"	Grado insertado satisfactoriamente.	

CU Actualizar nomenclador grado (Eliminar).		
Entrada	Resultados	Condiciones
Botón: "Eliminar"	Grado eliminado satisfactoriamente.	

CU Actualizar nomenclador institución (Insertar).		
Entrada	Resultados	Condiciones
Institución: "Preuniversitario" Nombre: "Carlos Marx"	Institución insertada satisfactoriamente.	

CU Actualizar nomenclador institución (Eliminar).		
Entrada	Resultados	Condiciones
Botón: "Eliminar"	Institución eliminada satisfactoriamente.	

CU Actualizar nomenclador municipio (Insertar).		
Entrada	Resultados	Condiciones
Municipio: "Limonar" Provincia: "Matanzas"	Municipio insertado satisfactoriamente.	

CU Actualizar nomenclador municipio (Eliminar).		
Entrada	Resultados	Condiciones
Botón: "Eliminar"	Municipio eliminado satisfactoriamente.	

CU Actualizar nomenclador especialidad (Insertar).		
Entrada	Resultados	Condiciones
Especialidad: "obrero"	Especialidad insertada satisfactoriamente.	

CU Actualizar nomenclador especialidad (Eliminar).		
Entrada	Resultados	Condiciones
Botón: "Eliminar"	Especialidad eliminada satisfactoriamente.	