Arquitectura del Sistema

Sistema VASPA

VASPA Team

Nicolás Sartini



*La Arquitectura del software, comprende el conjunto de elementos estáticos, propios del diseño intelectual del sistema, que definen y dan forma tanto al código fuente, como al comportamiento del software en tiempo de ejecución.*

*Naturalmente este diseño arquitectónico ha de ajustarse a las necesidades y requisitos del proyecto. Este documento describe en términos generales, las ideas principales detrás de la arquitectura escogida para el mismo.*



Tabla de contenido

[Introducción 4](#_Toc530853534)

[Propósito 4](#_Toc530853535)

[Alcance 4](#_Toc530853536)

[Definiciones, Acrónimos, y Abreviaturas 5](#_Toc530853537)

[Referencias 5](#_Toc530853538)

[Panorama General 5](#_Toc530853539)

[Representación Arquitectónica 5](#_Toc530853540)

[Representación 6](#_Toc530853541)

[Objetivos Arquitectónicos y Restricciones 6](#_Toc530853542)

[Objetivos Generales 6](#_Toc530853543)

[Objetivos Específicos 7](#_Toc530853544)

[Descripción de Procesos 7](#_Toc530853545)

[Vista de Caso de Uso 9](#_Toc530853546)

[Descripción de los Actores 9](#_Toc530853547)

[Contexto del sistema 10](#_Toc530853548)

[Vista Lógica 10](#_Toc530853549)

[Perspectiva General 10](#_Toc530853550)

[Paquetes de Diseño importantes arquitectónicamente 10](#_Toc530853551)

[Vista de Procesos 10](#_Toc530853552)

[Vista de Liberación 16](#_Toc530853553)

[Vista de Implementación 16](#_Toc530853554)

[Perspectiva General 16](#_Toc530853555)

[Capas 16](#_Toc530853556)

[Vista de Datos 16](#_Toc530853557)

[Tamaño y Rendimiento 17](#_Toc530853558)

[Calidad 17](#_Toc530853559)

[Diagramas 18](#_Toc530853560)

[Diagramas de Despliegue 18](#_Toc530853561)

[Diagrama de Objetos 18](#_Toc530853562)

[Diagramas de Paquetes 19](#_Toc530853563)

Arquitectura del Sistema

Introducción

El presente documento proporciona una vista de alto nivel de la arquitectura del Sistema VASPA, los objetivos y restricciones, los casos de uso más significativos, los patrones de arquitectura aplicados y las principales decisiones de diseño.

Este documento da una vista general del resto de los artefactos generados en el proceso de desarrollo.

Propósito

El propósito de este documento (por sus siglas en inglés, SAD) consiste en brindar una visión comprensible de la arquitectura general, utilizando diferentes vistas de la arquitectura para ilustrar los diferentes aspectos del sistema VASPA. Captura las decisiones más importantes de la arquitectura del sistema VASPA que fueron obtenidas a lo largo del proyecto.

El SAD está dirigido a distintos tipos de actores involucrados en el desarrollo del proyecto, tales como el Líder del Proyecto, Desarrolladores y clientes.

El Líder del Proyecto puede tomar este documento como base para mostrar la importancia de la arquitectura en el desarrollo de software, así como el rol del arquitecto de software.

Los Desarrolladores puede utilizar este documento como base para la documentación del desarrollo de productos de software en diferentes proyectos.

Desde el punto de vista del desarrollador, este documento le brindará, una buena razón para darle a la arquitectura de software la importancia que tiene en todo proyecto de desarrollo.

Alcance

El SAD pretende definir la arquitectura de software que será utilizada para el desarrollo del sistema VASPA. Como se verá posteriormente en este documento, esta representación arquitectónica es especificada principalmente por las vistas de casos de uso y lógica, incluyendo algunos elementos significativos de las otras vistas.

En este documento se le dará principal importancia a los casos de uso de la funcionalidad principal del sistema VASPA. Se abordarán principalmente aquellos casos de uso que involucran interacción con los actores.

Definiciones, Acrónimos, y Abreviaturas

* **SAD:** System Architecture Document por sus siglas en ingles. Hace referencia al Documento de Arquitectura del Sistema.
* **MVC:** Model View Controller por sus siglas en ingles. Hace referencia a la arquitectura del sistema Modelo Vista Controlador.

Referencias

[Esta sección provee una lista completa de documentos referenciados en cualquier lugar del Documento de Arquitectura de Software. Identifica cada documento por su título, número de reporte (si aplica), fecha, organización que publica, puede reverenciarse a un apéndice o a otro documento.]

Panorama General

[Esta sección describe el contenido y organización del Documento de Arquitectura de Software.]

Representación Arquitectónica

La arquitectura elegida para el sistema VASPA es la Modelo Vista Controlador (MVC), la cual se describe a continuación.

La arquitectura Modelo Vista Controlador (MVC) es un estilo arquitectónico de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.

* El Modelo contiene la representación de los datos que maneja el sistema, su lógica de negocio, y sus mecanismos de persistencia.
* La Vista, compone la información que se envía al usuario y los mecanismos interacción con éste.
* El Controlador, actúa como intermediario entre el Modelo y la Vista, gestionando el flujo de información entre ellos y las transformaciones para adaptar los datos a las necesidades de cada uno.

Tanto la vista como el controlador dependen del modelo, el cual no depende de las otras clases. Esta separación introduce el concepto de independencia, permitiendo así la construcción y prueba del modelo independientemente de la vista, permitiendo cambios en el controlador y la vista sin afectar la funcionalidad del sistema.

Se trata de un modelo muy maduro y que ha demostrado su validez a lo largo de los años en todo tipo de aplicaciones, y sobre multitud de lenguajes y plataformas de desarrollo.

## 

## Representación

Este documento presenta la arquitectura MVC como una serie de vistas: vista de casos de uso, vista lógica, vista de procesos, vista de liberación, vista de implementación y vista de datos. Estas vistas se basan en el Lenguaje Unificado de Modelado (UML).

* Vista de Casos de Uso: Describe el proceso de negocio más significativo y el modelo del dominio. Presenta los actores y los casos de uso para el sistema.
* Vista Lógica: Describe la arquitectura del sistema presentando varios niveles de refinamiento. Indica los módulos lógicos principales, sus responsabilidades y dependencias.
* Vista de Procesos: Describe los procesos concurrentes del sistema.
* Vista de Implementación: Describe los componentes de deployment construidos y sus dependencias.
* Vista de Datos: Presenta los modelos de datos, los servicios de persistencia y los servicios de transaccionalidad utilizados.

Las vistas de casos de uso y la vista lógica serán consideradas las más relevantes en el SAD.

La arquitectura está representada por diferentes vistas utilizando notación UML de forma que permitan visualizar, entender y razonar sobre los elementos significativos de la arquitectura e identificar las áreas de riesgo que requieren mayor detalle de elaboración.

Las siguientes secciones detallan las vistas de la arquitectura que serán utilizadas.

Objetivos Arquitectónicos y Restricciones

Objetivos Generales

El objetivo general del proyecto será:

Desarrollar un sistema de Visualización, Administración/ Gestión y Seguimiento de Programas de las diferentes asignaturas, principalmente para los Empleados Administrativos de Secretaría Académica, que permita a los mismos un manejo simple y se presente como una herramienta útil para su uso en las tareas específicas que cotidianamente llevan a cabo, mejorando así, la metodología de trabajo, como así también generar y/o mejorar vínculos con los profesionales de distintas carreras. Posibilitando además estrechar distancias entre los docentes, alumnos y personal administrativo mediante la utilización de Software en el ámbito académico.

Objetivos Específicos

Para lograr el objetivo mencionado, se describen a continuación metas específicas que se proponen alcanzar.

Investigar cómo los Empleados Administrativos de Secretaría Académica realizan sus actividades actualmente, de qué manera, qué herramientas utilizan, como así también comprender el proceso actual de trabajo y cada una de sus actividades que lo componen, para así poder desarrollar software que se adapte y mejore significativamente su labor diaria.

Investigar formas alternativas y más eficientes, para mejorar el proceso realizado actualmente, dando soluciones a los problemas que tienen, para así aportar un valor agregado, extra al sistema en desarrollo.

Aplicar las buenas prácticas adquiridas en las asignaturas de Ingeniería de Software de la carrera.

Aplicar estándares de documentación y de programación, basados en las plantillas de la Metodología PSI.

Investigar sobre técnicas y lenguajes de programación, como así también tecnologías complementarias que permitan llevar a cabo el desarrollo en cuestión.

Descripción de Procesos

El proceso comienza cuando los empleados administrativos del área Secretaría Académica observan en su planilla (hoja de cálculo) donde tienen registrados datos relevantes de los programas de las asignaturas, como la vigencia del mismo. En base a esto, solicitan, vía correo electrónico, el programa de asignatura correspondiente a los docentes responsables. El mismo puede ser o bien un nuevo programa o su correspondiente actualización. El programa debe actualizarse una vez transcurrido el período de tres años de vigencia o si ha ocurrido un cambio dentro del equipo de cátedra de la asignatura.

Los docentes titulares deben presentar dos copias firmadas en papel (aunque el ideal del área sería que se lo envíen en formato .doc/.docx vía correo electrónico). Una vez presentados los mismos, los empleados administrativos prosiguen a efectuar el control formal de los mismos, donde se verificará que el formato del documento, el código de la asignatura y el código de la carrera sean los correspondientes, contenidos mínimos, etcétera. Si no se han encontrado defectos en el armado del mismo, la Secretaria Académica prosigue a firmar dicho programa. El mismo, es enviado a su respectivo departamento (departamento de Ciencias Exactas y Naturales o departamento de Ciencias Sociales).

En el departamento al que fue enviado, se efectuaran las revisiones pertinentes al contenido académico que será impartido en la asignatura. Si la revisión es exitosa, el mismo será firmado correspondientemente por el director del departamento.

Acto seguido, el programa es devuelto a Secretaría Académica para culminar la formal aprobación del mismo.

Una vez terminado el proceso de evaluación y se han colocado las firmas de los responsables, los empleados administrativos de Secretaría Académica prosiguen a realizar la digitalización del documento (programa) para realizar una copia en formato digital, archivar la correspondiente copia en papel en un sistema de archivos y actualizar la información sobre los programas que se encuentran en un archivo de una hoja de cálculo en su computadora.

Los problemas con los que cuenta dicho proceso son los siguientes:

Problema 1: No existe un procedimiento claramente definido para el desarrollo del proceso interno.

Problema 2: No existe una respuesta por parte de los docentes cuando se les solicita vía correo electrónico, la posterior presentación de los programas.

Problema 3: No existe control para verificar a qué profesores ya se les ha solicitado que pasen a firmar o entregar el programa. Tampoco existe un registro de las veces que se les fueron solicitados dichos pedidos.

Problema 4: No existe un seguimiento sobre cada uno de los programas, es decir, su estado. El empleado de Secretaría Académica no sabe dónde se encuentra el programa durante el proceso de firmas.

Vista de Caso de Uso

Un caso de uso es una secuencia general de eventos, que describe todas las posibles acciones entre los posibles actores y el sistema, para una determinada funcionalidad.

En esta sección se describen los Casos de Uso del sistema VASPA, en donde se abarcan todas las funcionalidades del sistema, se muestran los actores que interactúan en el sistema y las funcionalidades asociadas.

Descripción de los Actores

|  |  |
| --- | --- |
| Nombre del Actor: | Invitado |
| Definición : | Es cualquier persona que no se haya autentificado en el sistema y que puede visualizar y descargar los programas de asignatura que desee. |

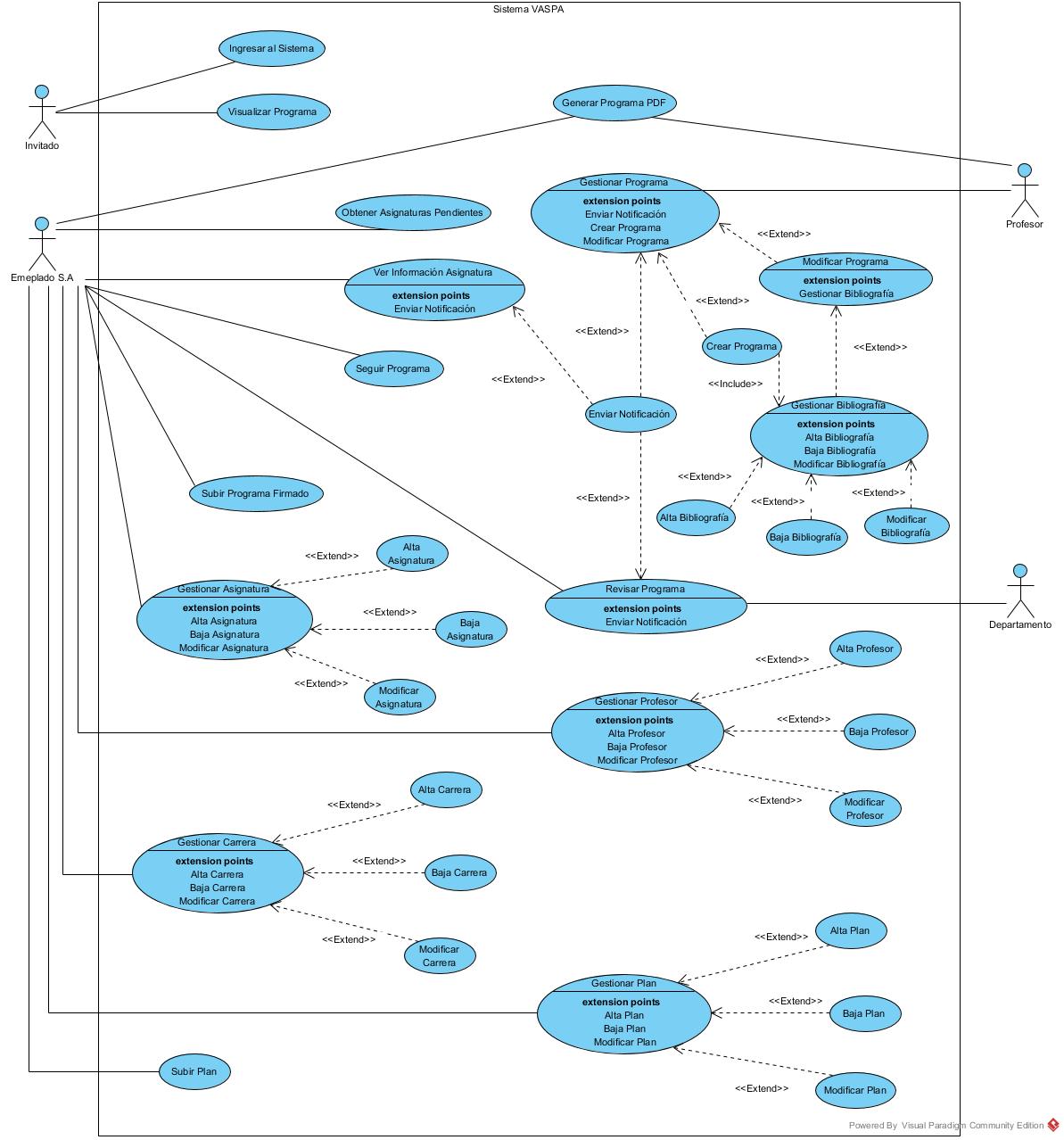
|  |  |
| --- | --- |
| Nombre del Actor: | Empleado S.A |
| Definición : | Es el encargado de gestionar los programas de asignaturas (solicitar, controlar, modificar, cargar). Posteriormente, es el encargado de realizar la aprobación del mismo, mediante su firma. |

|  |  |  |
| --- | --- | --- |
| Nombre del Actor: | | Profesor |
| Definición : | Es el titular de asignaturas, y es el encargado de cargar el formulario con los datos correspondientes para generar el programa de la materia de la cual es responsable y de presentar el programa (impreso) en Secretaria Académica. Posteriormente realiza la aprobación del mismo, mediante su firma. | |

|  |  |  |
| --- | --- | --- |
| Nombre del Actor: | | Departamento |
| Definición : | Es el responsable de controlar/ revisar los programas de asignatura. Posteriormente, es el encargado de realizar la aprobación del mismo, mediante su firma. | |

Contexto del sistema

En el siguiente diagrama se puede observar como los distintos casos de uso interactúan unos con otros y como está compuesto el contexto general del sistema.



Vista Lógica

[En esta sección se describen las partes significantes de arquitectura del modelo de diseño, tal como su composición dentro del sistema y paquetes. Y para cada paquete importante, su descomposición dentro de estructuras (clases) y utilidades de las mismas. Podría introducir estructuras arquitectónicas importantes y describir sus responsabilidades, así como relaciones, operaciones y atributos importantes.]

Perspectiva General

[Esta sección describe la composición general de modelo de diseño en términos de jerarquía y capas.]

Paquetes de Diseño importantes arquitectónicamente

[Para cada paquete significante, incluye una sección con su nombre, una descripción breve y un diagrama con estructuras significantes y paquetes contenidos dentro del paquete.

Para cada estructura significante en el paquete, incluye su nombre, una breve descripción, y, opcionalmente, una descripción de algunas de sus principales responsabilidades, operaciones y atributos.]

Vista de Procesos

El proceso comienza cuando los empleados administrativos del área Secretaría Académica observan en su planilla (hoja de cálculo) donde tienen registrados datos relevantes de los programas de las asignaturas, como la vigencia del mismo. En base a esto, solicitan, vía correo electrónico, el programa de asignatura correspondiente a los docentes responsables. El mismo puede ser o bien un nuevo programa o su correspondiente actualización. El programa debe actualizarse una vez transcurrido el período de tres años de vigencia o si ha ocurrido un cambio dentro del equipo de cátedra de la asignatura.

Los docentes titulares deben presentar dos copias firmadas en papel (aunque el ideal del área sería que se lo envíen en formato .doc/.docx vía correo electrónico). Una vez presentados los mismos, los empleados administrativos prosiguen a efectuar el control formal de los mismos, donde se verificará que el formato del documento, el código de la asignatura y el código de la carrera sean los correspondientes, contenidos mínimos, etcétera. Si no se han encontrado defectos en el armado del mismo, la Secretaria Académica prosigue a firmar dicho programa. El mismo, es enviado a su respectivo departamento (departamento de Ciencias Exactas y Naturales o departamento de Ciencias Sociales).

En el departamento al que fue enviado, se efectuaran las revisiones pertinentes al contenido académico que será impartido en la asignatura. Si la revisión es exitosa, el mismo será firmado correspondientemente por el director del departamento.

Acto seguido, el programa es devuelto a Secretaría Académica para culminar la formal aprobación del mismo.

Una vez terminado el proceso de evaluación y se han colocado las firmas de los responsables, los empleados administrativos de Secretaría Académica prosiguen a realizar la digitalización del documento (programa) para realizar una copia en formato digital, archivar la correspondiente copia en papel en un sistema de archivos y actualizar la información sobre los programas que se encuentran en un archivo de una hoja de cálculo en su computadora.

Los problemas con los que cuenta dicho proceso son los siguientes:

Problema 1: No existe un procedimiento claramente definido para el desarrollo del proceso interno.

Problema 2: No existe una respuesta por parte de los docentes cuando se les solicita vía correo electrónico, la posterior presentación de los programas.

Problema 3: No existe control para verificar a qué profesores ya se les ha solicitado que pasen a firmar o entregar el programa. Tampoco existe un registro de las veces que se les fueron solicitados dichos pedidos.

Problema 4: No existe un seguimiento sobre cada uno de los programas, es decir, su estado. El empleado de Secretaría Académica no sabe dónde se encuentra el programa durante el proceso de firmas.

Según el diseño que se le dio al sistema, nos encontramos con las siguientes funcionalidades principales:

|  |  |
| --- | --- |
| **Ingresar al Sistema** | |
| Descripción : | Permitir al invitado acceder al sistema, para obtener los permisos necesarios para cargar, visualizar, modificar y descargar el programa de asignatura correspondiente.  Permitir al invitado poder autentificarse en el sistema mediante su correo institucional y tener acceso a determinadas acciones de acuerdo a su rol. |

|  |  |
| --- | --- |
| **Visualizar Programa** | |
| Descripción : | Permitir al Invitado visualizar el programa (documento PDF). |

|  |  |
| --- | --- |
| **Generar Programa PDF** | |
| Descripción : | Permitir al Profesor y al Empleado Secretaría Académica generar un programa en formato .PDF a partir de los formularios. |

|  |  |
| --- | --- |
| **Gestionar Programa** | |
| Descripción : | Permitir al empleado de Secretaría Académica gestionar los Formularios de los Programas de Asignaturas de la UNPA - UARG existentes en el sistema, mediante la baja de los mismos.  Permitir al Profesor cargar y modificar datos de los formularios de los programas de las asignaturas de las cuales es responsable en la UNPA – UARG. |

|  |  |
| --- | --- |
| **Ver Información Asignatura** | |
| Descripción : | Permitir al Empleado Secretaría Académica visualizar la información importante (docente titular y vigencia del programa) de una asignatura. |

|  |  |
| --- | --- |
| **Obtener Asignaturas Pendientes** | |
| Descripción : | Permitir al empleado de Secretaría Académica obtener un listado de las asignaturas, a partir de una carrera seleccionada, en la cuales no se presentaron los programas de acuerdo a un año especifico. |

|  |  |
| --- | --- |
| **Seguir Programa** | |
| Descripción : | Permitir al Empleado Secretaría Académica el seguimiento del programa, para saber dónde y cuantos días se encuentra el programa durante el proceso de firmas. |

|  |  |
| --- | --- |
| **Subir Programa Firmado** | |
| Descripción : | Permitir al Empleado Secretaría Académica cargar en el sistema el programa escaneado del documento (programa impreso) firmado por todas las autoridades correspondientes (Profesor responsable, Secretaría Académica, departamento). |

|  |  |
| --- | --- |
| **Gestionar Asignatura** | |
| Descripción : | Permitir al empleado de Secretaría Académica gestionar las Asignaturas existentes en el sistema, mediante el alta, baja y modificación de las mismas. |

|  |  |
| --- | --- |
| **Revisar Programa** | |
| Descripción **:** | Permitir al Empleado Secretaría Académica y al jefe del departamento revisar las secciones correspondientes del formulario del programa, realizando observaciones (en el caso que corresponda) al final del mismo, para que el docente realice las modificaciones necesarias. |

|  |  |
| --- | --- |
| **Gestionar Profesor** | |
| Descripción : | Permitir al empleado de Secretaría Académica gestionar los profesores existentes en el sistema, mediante el alta, baja y modificación de los mismos. |

|  |  |
| --- | --- |
| **Gestionar Carrera** | |
| Descripción : | Permitir al empleado de Secretaría Académica gestionar las carreras existentes en el sistema, mediante el alta, baja y modificación de las mismas. |

|  |  |
| --- | --- |
| **Gestionar Plan** | |
| Descripción : | Permitir al empleado de Secretaría Académica gestionar los Planes de Estudios de Carreras de la UNPA - UARG existentes en el sistema, mediante el alta, baja y modificación de los mismos. |

|  |  |
| --- | --- |
| **Subir Plan** | |
| Descripción : | Permitir al Empleado Secretaría Académica cargar en el sistema el plan escaneado del documento (plan impreso). |
|  |  |
|  |  |
| **Enviar Notificación** | |
| Descripción : | Permitir al Empleado de Secretaría Académica enviar notificaciones de aviso al profesor, notificándole los programas de asignaturas pendientes que debe presentar. |

|  |  |
| --- | --- |
| **Crear Programa** | |
| Descripción : | Permitir al Profesor crear un programa de asignatura ingresando los datos requeridos. |
|  |  |
| **Modificar Programa** | |
| Descripción : | Permitir al profesor modificar/actualizar los datos que considere necesarios del programa. |
|  |  |
| **Gestionar Bibliografía** | |
| Descripción : | Permitir al profesor gestionar la bibliografía de los programas de asignatura, mediante el alta, baja y modificación de las mismas. |
|  |  |

Vista de Liberación

[Esta sección describe una o más configuraciones físicas de la red (hardware) en las cuales el software se libera y funciona. Es una vista del modelo de liberación. En un mínimo para cada configuración debe indicar los nodos físicos (computadoras, CPUs) que ejecutan el software y sus interconexiones (bus, LAN, punto al punto, etcétera.) también incluye un mapa de los procesos de la vista de procesos sobre los nodos físicos.]

Vista de Implementación

[Esta sección describe la estructura total del modelo de la puesta en marcha, la descomposición del software en capas y subsistemas en el modelo de implementación y cualquier componente arquitectónico significativo.]

Perspectiva General

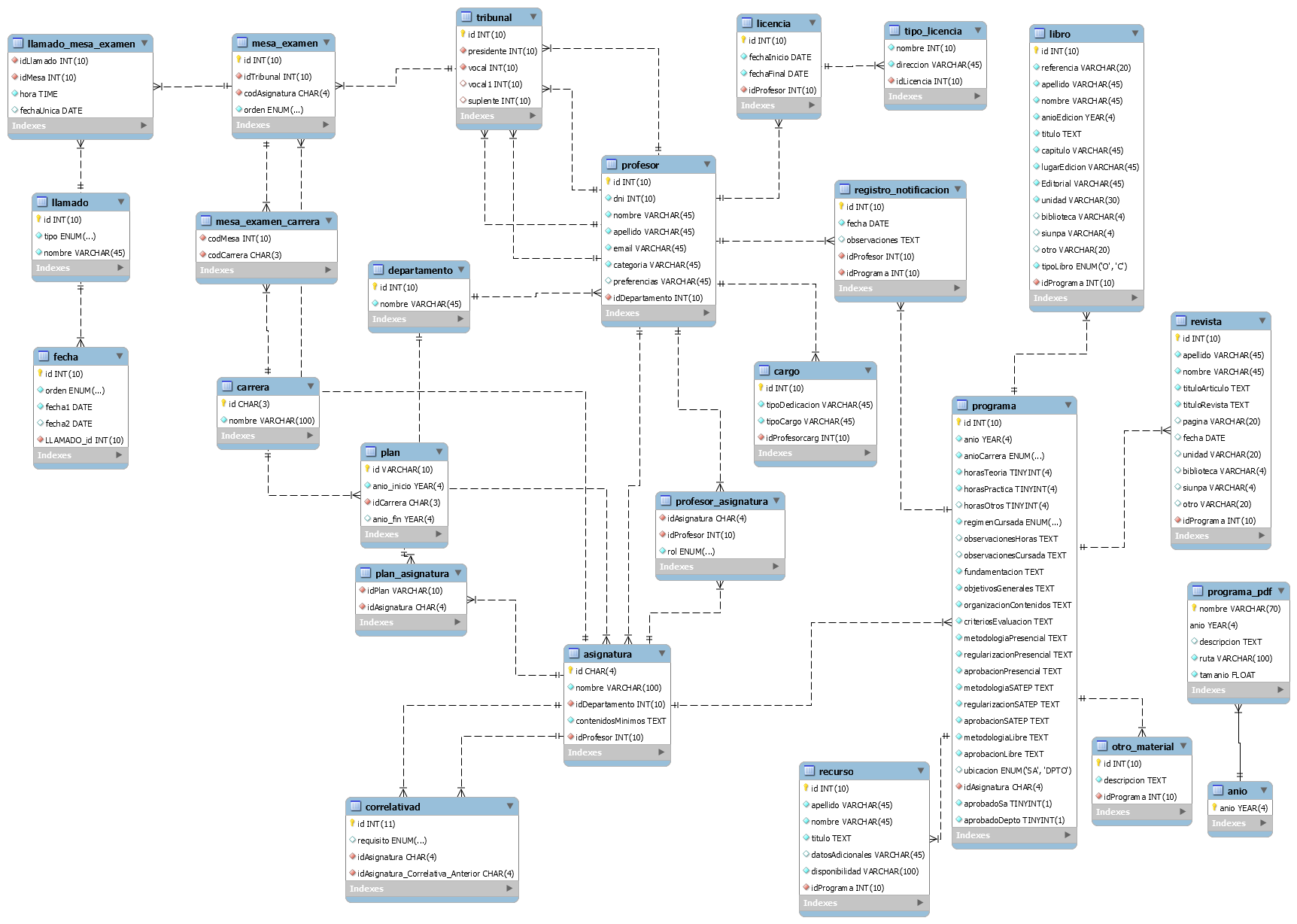
[Esta subdivisión nombra y define varias capas y su contenido, las reglas que gobiernan la inclusión a una capa dada, y los límites entre las capas. Incluir un diagrama componente que demuestre las relaciones entre las capas.]

Capas

[Para cada capa, se debe incluir una subdivisión con su nombre, una enumeración de los subsistemas situados en la capa, y un diagrama de componentes.]

Vista de Datos

En esta vista se explica la manera como fueron almacenados los datos en la base de datos del Sistema VASPA. En el diagrama de entidad‐relación se pueden ver las tablas utilizadas para la persistencia de los datos.



Tamaño y Rendimiento

[Colocar una descripción con las características principales de la dimensión del software que afectan la arquitectura, así como las restricciones de desempeño del objetivo.]

Calidad

[Describir cómo la arquitectura del software contribuye a la capacidad general (con excepción de funcionalidad) del sistema: extensibilidad, confiabilidad, portabilidad, etcétera. Si estas características tienen significación especial, por ejemplo seguridad, garantía o implicaciones de privacidad, deben ser claramente delineadas.]

Diagramas

Diagramas de Despliegue

[Básicamente este tipo de diagrama se utiliza para modelar el Hardware utilizado en la implementación del sistema y las relaciones entre sus componentes. Los elementos usados por este tipo de diagrama son nodos, componentes y asociaciones. En el UML 2.0 los componentes ya no están dentro de nodos, en cambio puede haber artefactos (archivo, un programa, una biblioteca o Base de datos) u otros nodos dentro de nodos.

Además los Diagramas de Despliegue muestran la configuración en funcionamiento del sistema incluyendo su software y su hardware. Para cada componente de un diagrama es necesario que se deba documentar las características técnicas requeridas, el tráfico de red, el tiempo de respuesta, etc.]

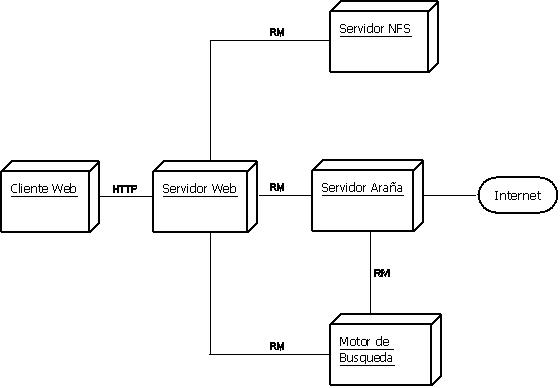


Diagrama de Objetos

[Forma parte de la vista estática del sistema. En este diagrama se modelan las instancias de la clase del Diagrama de Clases, cabe aclarar que el mismo cuenta con objetos y enlaces. En estos diagramas también es posible encontrar las clases para tomar como referencia su instanciación. En otras palabras el Diagrama de Objetos muestra un conjunto de objetos y sus relaciones en un momento concreto. Los Diagramas de Objetos son realmente útiles para modelar estructuras de datos complejas. ]



Diagramas de Paquetes

[Los diagramas de Paquetes se usan para reflejar la organización de paquetes y sus elementos. Los usos más comunes de para los diagrama de paquete son para organizar diagramas de casos de uso y diagramas de clases, estos paquetes son como grandes contenedores de clases.

Los elementos contenidos en un paquete comparten el mismo espacio de nombres, esto significa que los elementos contenidos en un mismo espacio de nombres específico deben tener nombres únicos. Como otra característica de estos diagramas, cada paquete se debe identificar con un nombre único y opcionalmente mostrar todos los elementos dentro del mismo. ]

