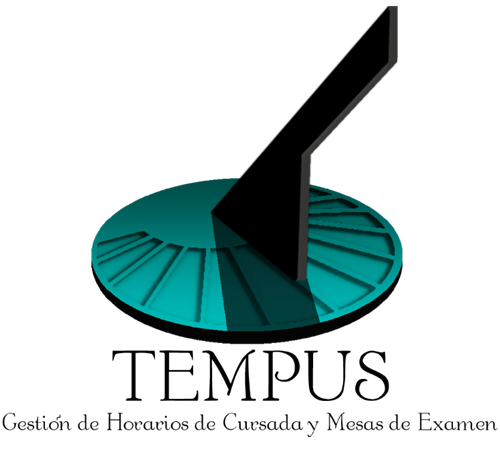


Arquitectura del Sistema

Tempus



*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 2](#_Toc257618836)

[Propósito 2](#_Toc257618837)

[Alcance 2](#_Toc257618838)

[Definiciones, Acrónimos, y Abreviaturas 2](#_Toc257618839)

[Referencias 2](#_Toc257618840)

[Panorama General 2](#_Toc257618841)

[Representación Arquitectónica 2](#_Toc257618842)

[Objetivos Arquitectónicos y Restricciones 2](#_Toc257618843)

[Objetivos Generales 2](#_Toc257618844)

[Objetivos Específicos 2](#_Toc257618845)

[Descripción de Procesos 2](#_Toc257618846)

[Vista de Caso de Uso 2](#_Toc257618847)

[Descripción de los Actores 2](#_Toc257618848)

[Contexto del sistema 2](#_Toc257618849)

[Vista Lógica 2](#_Toc257618850)

[Perspectiva General 2](#_Toc257618851)

[Paquetes de Diseño importantes arquitectónicamente 2](#_Toc257618852)

[Vista de Procesos 2](#_Toc257618853)

[Vista de Liberación 2](#_Toc257618854)

[Vista de Implementación 2](#_Toc257618855)

[Perspectiva General 2](#_Toc257618856)

[Capas 2](#_Toc257618857)

[Vista de Datos (opcional) 2](#_Toc257618858)

[Tamaño y Rendimiento 2](#_Toc257618859)

[Calidad 2](#_Toc257618860)

[Diagramas 2](#_Toc257618861)

[Diagramas de Despliegue 2](#_Toc257618862)

[Diagrama de Objetos 2](#_Toc257618863)

[Diagramas de Paquetes 2](#_Toc257618864)

Arquitectura del Sistema

Introducción

[La introducción del Documento de Arquitectura de Software provee un panorama completo que incluye el propósito, alcance, definiciones, acrónimos, abreviaturas y referencias.]

El presente documento provee una vista de alto nivel de la arquitectura del Sistema en desarrollo, propósitos, alcance, definiciones, acrónimos, abreviaturas, referencias, los objetivos y restricciones, los casos de uso 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

[Esta sección define el propósito del Documento de Arquitectura de Software, en la documentación del proyecto en conjunto y describe brevemente la estructura del documento. Identifica las audiencias específicas para el documento, con una indicación de cómo se espera que utilicen el documento.]

Este documento de arquitectura de software (por sus siglas en inglés, SAD) tiene como propósito brindar una visión comprensible de la arquitectura general, utilizando diferentes vistas de la arquitectura para ilustrar diferentes aspectos del sistema. Captura las decisiones más importantes en lo que respecta a la arquitectura del sistema que fueron tomadas en el proyecto.

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

Los desarrolladores puede utilizar este documento como base para la documentación del desarrollo de productos de software en proyectos de diferente porte.

El tutor 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.

Es de nuestro interés intercambiar con ellos ideas y resultados.

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

Alcance

[Una breve descripción de donde aplica, afecta o influencia el Documento de Arquitectura de Software.]

Este SAD profundiza principalmente en 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. Se atacará principalmente aquellos casos de uso que involucran interacción con los actores.

Finalmente, este documento es la descripción de arquitectura del caso de estudio, no es un instructivo de cómo elaborar un SAD; en otras palabras el lector no encontrará aquí comentarios sobre qué tipo de información debe ser incluida en el documento ni cuáles son los criterios a utilizar en casos generales.

Definiciones, Acrónimos, y Abreviaturas

[Esta sección provee las definiciones de todos los términos, acrónimos y abreviaturas requeridas para interpretar apropiadamente el Documento de Arquitectura de Software. Esta información puede referenciar al Glosario del Proyecto.]

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

[Esta sección describe la arquitectura de software para el sistema y como se representa. Enumera las vistas que son necesarias, como las vistas de Caso de Uso, Lógica, de Procesos, de Liberación e Implementación, así como una explicación de los tipos de elementos contenidos en el modelo.]

La características inherentes al sistema hace que la vista de casos de uso y la vista lógica sean las más relevantes y por ello serán las más extensas.

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. Este documento es una forma de comunicar el modelo del subsistema, presentando la información y discusiones estructuradamente.

La arquitectura del subsistema se descompone en las siguientes dimensiones:

Requerimientos: Requerimientos funcionales y no-funcionales del sistema.

Elaboración: Representación lógica del sistema y representación de tiempo de ejecución.

Implementación: Vista de módulos implementados, potenciales escenarios de infraestructura y el deployment de los módulos.

## 

## Representación

La siguiente sección detalla las vistas de la arquitectura que serán utilizadas para cubrir las dimensiones mencionadas.

La arquitectura del Subsistema de Reservas está representada siguiendo las recomendaciones de RUP. Las vistas necesarias para especificar dicho subsistema se enumeran a continuación:

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 de Restricciones: Describe restricciones tecnológicas, normativas, uso de estándares, entre otros, las cuales deben ser respetadas tanto por el proceso de desarrollo como por el producto desarrollado.

Vista QoS: Incluye aspectos de calidad, y describe los requerimientos no-funcionales del 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.

Vista de Deployment: Presenta aspectos físicos como topología, infraestructura informática, e instalación de ejecutables. Incluye además plataformas y software de base.

Estas vistas son las generales para cualquier sistema que se intente desarrollar, para nuestro caso, solo nos explayaremos en las que son significativas para este desarrollo.

La implementación todas estas, están sujetas a las demandas propias del desarrollo.

Las vistas presentadas forman en su conjunto una especificación completa del sistema, la cual se delinea en el siguiente diagrama. Las dependencias entre las vistas indican dependencias entre las vistas tanto a nivel de arquitectura como a nivel de diseño.



Objetivos Arquitectónicos y Restricciones

[Esta sección describe los requerimientos y objetivos que tienen algún impacto significante en la arquitectura; por ejemplo, seguridad, garantía, privacidad, portabilidad, distribución, reuso. Captura restricciones especiales que puedan aplicar estrategias de diseño e implementación, herramientas de desarrollo, estructura del equipo, programación, código legado, etc.]

Objetivos Generales

[Esta sección describe de manera general los objetivos del proyecto, con el ánimo de comprender que motiva al grupo de desarrollo emprender el desarrollo]

El objetivo general del Sistema de Gestión de horarios de cursada y mesas de examen, es permitir la realización de la carga de horarios de cursada y mesas de examen haciendo uso de las planillas de calculo que poseen actualmente los empleados de Secretaria Académica de la UNPA-UARG. Esta tarea debe hacerse accediendo al sistema a través de su sitio web e identificándose mediante un correo institucional.

También se permite a usuarios públicos que accedan desde una aplicación móvil consultar los horarios de cursada y mesas de examen que han sido previamente cargados por Secretaría Académica. Recibiendo además notificación por cambios ocurridos en las mesas de examen.

Objetivos Específicos

[Esta sección describe los objetivos que tienen algún impacto significante en la arquitectura. Intenta describir detalladamente que es lo que se pretende obtener con el desarrollo del proyecto.]

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

Integrar uargflow basado en la arquitectura cliente-servidor. Usando el patrón de diseño mvc con POO y PHP para nuestro sitio web.

Además se debe establecer la comunicación entre el sitio web y la app con la base de datos mysql

Investigar técnicas para poder importar un archivo Excel a la base de datos. El esquema de base de datos debe tener en cuenta las características de los archivos que se deben importar al sistema pero sin desperdiciar las características de MER.

Con respecto a la seguridad se debe considerar el acceso a información privada.

Portabilidad por la aplicación.

Integrar los conocimientos mencionados en una solución de software que se convierta en una herramienta práctica para los docentes, alumnos y personal de Secretaria Académica de la UNPA-UARG.

Descripción de Procesos

[Esta sección describe los procesos que son significativos a la arquitectura del sistema se debe tener en cuenta el flujo de información y que personas interactúan con la misma]

El proceso comienza cuando un docente o alumno quiere consultar a través de la app por horarios de cursada o mesas de examen. Brindando información actualizada

Vista de Caso de Uso

[Esta sección lista casos de uso o escenarios de modelos de caso de uso, si representan alguna funcionalidad central o significante en el sistema final, o si tienen una gran cobertura arquitectónica—forman muchos elementos arquitectónicos o si ilustran un específico punto delicado de la arquitectura.]

Descripción de los Actores

[Esta sección se describen todos los actores que tienen relación con los casos de uso relevantes para la arquitectura, en esta descripción se debe hacer hincapié en el por qué de la relación con el caso de uso en cuestión dando una descripción del actor y sus funciones]

Contexto del sistema

[Esta sección se debe mostrar el sistema dentro de su contexto mostrando la relación con otros sistemas o sub- sistemas, también se pude incluir un diagrama de contexto de ser necesario]

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

[Esta sección describe la descomposición del sistema dentro del proceso ligeros (simple hilos de control) y procesos pesados (grupos de procesos ligeros). Organice la sección por grupos de procesos que se comunican e interactúan. Describe el principal modo de comunicación entre procesos, tal como mensajes, interrupciones.]

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 (opcional)

[Se debe colocar una descripción del almacenamiento de datos persistentes del sistema. Esta sección es opcional si hay o no datos persistentes, o si la traducción entre el modelo del diseño y el modelo de los datos es trivial.]

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.]

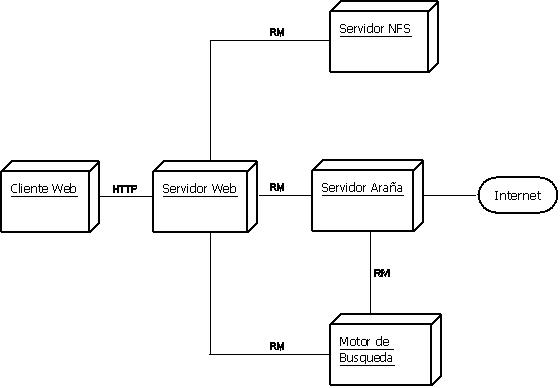


Diagrama de Objetos

[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. ]

