Versión 1.0

Integrantes:

Derian Molina, Starling Nuñez y Georffrey Carmona

Proyecto cal

Documento de arquitectura de software

TecnoLógico de costa rica

# Historial de revisiones

|  |  |  |  |
| --- | --- | --- | --- |
| Fecha | Sección trabajada | Descripción | Autor |
| 17/12/2018 | Todas | Creación de la estructura base del documento | Derian Molina, Georffrey Carmona, Starling Nuñez. |
| 26/12/2018 | Toda la sección 1 | Se desarrollaron las secciones | Derian Molina, Georffrey Carmona, Starling Nuñez. |
| 27/12/2018 | Toda la sección 2 | Se desarrollaron las secciones | Derian Molina, Georffrey Carmona, Starling Nuñez. |
| 28/12/2018 | Toda la sección 3 | Se desarrollaron las secciones | Derian Molina, Georffrey Carmona, Starling Nuñez. |
| 2 | Toda la sección 4,5 | Se desarrollaron las secciones | Derian Molina, Georffrey Carmona, Starling Nuñez. |
| 4 | Toda la sección 6,7 | Se desarrollaron las secciones | Derian Molina, Georffrey Carmona, Starling Nuñez. |
| 6 | Toda la sección 8,9 | Se desarrollaron las secciones | Derian Molina, Georffrey Carmona, Starling Nuñez. |
| 7 | Toda la sección 10,11 | Se desarrollaron las secciones | Derian Molina, Georffrey Carmona, Starling Nuñez. |

# Tabla de Contenido

[Historial de revisiones 1](#_Toc534803833)

[Tabla de Contenido 2](#_Toc534803834)

[Tabla de Ilustraciones 3](#_Toc534803835)

[1. Introducción 4](#_Toc534803836)

[1.1 Propósito 4](#_Toc534803837)

[1.2 Alcance 4](#_Toc534803838)

[1.3 Definiciones, acrónimos y abreviaciones 5](#_Toc534803839)

[1.4 Referencias 5](#_Toc534803840)

[1.5. Visión General 6](#_Toc534803841)

[2. Representación de la arquitectura 6](#_Toc534803842)

[2.1 Arquitecturas de acuerdo a los atributos de calidad 6](#_Toc534803843)

[2.2 Representación arquitectónica 7](#_Toc534803844)

[2.2.1 Cliente 7](#_Toc534803845)

[2.2.1.1 Presentación 8](#_Toc534803846)

[2.2.1.1.1 Modelo-Vista-Controlador 8](#_Toc534803847)

[2.2.1.2 Seguridad 9](#_Toc534803848)

[2.2.1.3 Lógica de Negocio 9](#_Toc534803849)

[2.2.2 Servidor 10](#_Toc534803850)

[2.2.3 Capas 10](#_Toc534803851)

[2.3 Representación arquitectónica 10](#_Toc534803852)

[3. Objetivos y restricciones Arquitectónicas 11](#_Toc534803853)

[3.1 Objetivos 11](#_Toc534803854)

[3.2 Restricciones 11](#_Toc534803855)

[3.3 Resumen de requerimientos 12](#_Toc534803856)

[4. Vista de Casos de Uso 12](#_Toc534803857)

[5. Vista Lógica 14](#_Toc534803858)

[5.1 Visión General 15](#_Toc534803859)

[5.2 Diseño arquitectónico de paquetes importantes 15](#_Toc534803860)

[5.2.1 Paquete de Presentación 15](#_Toc534803861)

[5.2.2 Lógica de Negocio 15](#_Toc534803862)

[5.2.3 Conexión 15](#_Toc534803863)

[5.2.4 Datos 15](#_Toc534803864)

[6. Vista de Procesos 15](#_Toc534803865)

[7. Vista de Despliegue 16](#_Toc534803866)

[8. Vista de Implementación 17](#_Toc534803867)

[8.1 Visión General 18](#_Toc534803868)

[8.2 Capas 18](#_Toc534803869)

[8.2.1 Presentación 18](#_Toc534803870)

[8.2.2 Seguridad 19](#_Toc534803871)

[8.2.3 Lógica de Negocio 19](#_Toc534803872)

[8.2.4 Datos 20](#_Toc534803873)

[9. Vista de Datos 21](#_Toc534803874)

[10. Tamaño y desempeño 22](#_Toc534803875)

[10.1 Concurrencia de usuarios 22](#_Toc534803876)

[11. Calidad 23](#_Toc534803877)

[Seguridad 23](#_Toc534803878)

[Despliegue de la información 23](#_Toc534803879)

[Validación de la información del usuario 23](#_Toc534803880)

[Usabilidad 23](#_Toc534803881)

[Visuales 23](#_Toc534803882)

# Tabla de Ilustraciones

[Ilustración 1 MVC 9](#_Toc534803790)

[Ilustración 2 Modelo Vista 4+1 de Kruchten 11](#_Toc534803791)

[Ilustración 3 Vista de Casos de Uso 13](#_Toc534803792)

[Ilustración 4 Vista Lógica 14](#_Toc534803793)

[Ilustración 5 Vista de Procesos 16](#_Toc534803794)

[Ilustración 6 Vista Despliegue 17](#_Toc534803795)

[Ilustración 7 Vista Implementación 17](#_Toc534803796)

[Ilustración 8 Capa Presentación 18](#_Toc534803797)

[Ilustración 9 Capa de Seguridad 19](#_Toc534803798)

[Ilustración 10 Capa Lógica del Negocio 20](#_Toc534803799)

[Ilustración 11 Capa de Datos 21](#_Toc534803800)

[Ilustración 12 Vista de Datos 22](#_Toc534803801)

# 1. Introducción

En este apartado del documento lo que se pretende es dar una visión general del documento de arquitectura del proyecto propuesto, llamado de aquí en adelante como SAD. Además, en este apartado se abarcará el propósito, alcance, visión general del documento, definiciones, entre otros.

## 1.1 Propósito

Este documento se realiza con el fin especificar la arquitectura de software a ser utilizada para el desarrollo del sistema ProyectoCAL que es el producto del trabajo de los miembros del equipo de trabajo. Con dicho documento se pretende mostrar en términos arquitectónicos y de diseño todos los requerimientos definidos en el ERS (Documento de especificación de requerimientos de software).

El documento está orientado a los interesados del proyecto (Stakeholders). Estos son cliente, sponsor, administrador o gerente del proyecto y el equipo de trabajo. Además, este documento es parte de los entregables del curso Proyecto de Ingeniería de Software realizado por los estudiantes autores del mismo, bajo el seguimiento de la Ingeniera María Estrada Sánchez.

## 1.2 Alcance

El alcance de este documento aplica única y específicamente al software propuesto, el cual comprende de un sistema para la gestión de trámites en el centro Académico de Limón (Control de salida, concursos de asistencia, reservación de aulas/laboratorios, entre otros). Estas funcionalidades del sistema estarán conectadas a un motor de bases de datos para la administración de toda la información. Además, el desarrollo del proyecto tendrá 3 iteraciones y a lo largo del trabajo se deberán presentar documentos, por ejemplo, el SAD, ERS, papers, entre otros.

## 1.3 Definiciones, acrónimos y abreviaciones

|  |  |
| --- | --- |
| Concepto | Descripción |
| SAD | Software Architecture Design |
| CRUD | Se refiere a las acciones que se pueden generar con un motor de base de datos:   * Crear * Leer * Actualizar * Borrar |
| CAL | Centro Académico de Limón |
| ERS | Especificación de requerimientos de software, el cual se encarga de describir las funcionalidades del sistema y como se van a desempeñar. |

Tabla 1: Definiciones, Acrónimos y Abreviaciones

## 1.4 Referencias

Escobar, I. G. (2005). *UTN*. Obtenido de UTN: http://apit.wdfiles.com/local--files/start/06\_apit\_rol\_del\_arquitecto.pdf

Flores, L. U. (01 de 06 de 2011). *Documentando sistemas*. Obtenido de Documentando sistemas: https://drive.google.com/file/d/0B2o07nw5oICzZnBrTFV5cHhVb28/edit

IEEE. (25 de Junio de 1998). *Kybele*. Obtenido de Kybele: http://www.kybele.etsii.urjc.es/docencia/IS4/extra/IEEE%20830-1998%20%5BENG%5D.pdf

IEEE. (22 de 10 de 2008). *UCM*. Obtenido de UCM: https://www.fdi.ucm.es/profesor/gmendez/docs/is0809/ieee830.pdf

Rouse, M. (Febrero de 2007). *techtarget*. Obtenido de techtarget: https://searchsoftwarequality.techtarget.com/definition/software-requirements-specification

## 1.5. Visión General

# 2. Representación de la arquitectura

En este apartado del documento se describe la arquitectura de diseño seleccionada para el sistema ProyectoCAL.

## 2.1 Arquitecturas de acuerdo a los atributos de calidad

Este método está basado en el “Modelo ISO/IEC 9126” que se concentra en la selección de atributos de calidad para la arquitectura de software. Las actividades que describe dicho modelo es la siguiente:

|  |  |
| --- | --- |
| Actividad | Descripción |
| Análisis | Se estudian los requerimientos funcionales y no funcionales para seleccionar atributos e calidad. |
| Métricas | Según el modelo ISO 9126 se definen las métricas. |
| Arquitecturas candidatas | De acuerdo a las métricas definidas se realiza una selección posible de arquitecturas candidatas. |
| Comparativas | Se realiza un análisis de comparación de las posibles arquitecturas existentes. |
| Prioridades | Se definen prioridades para sub características de calidad de acuerdo a los requerimientos de calidad del sistema. |
| Selección de arquitectura | Se realiza un estudio de las posibles arquitecturas y se selecciona la mejor arquitectura para el sistema a desarrollar. |

Existen diferentes atributos de calidad definidos según la norma ISO 9126 con sus categorías y sub categorías, pero en el desarrollo del sistema propuesto se describen los atributos seleccionados que se describen a continuación:

|  |  |
| --- | --- |
| Categoría | Sub Categoría |
| Funcionalidad | Idoneidad |
| Seguridad |
| Precisión |
| Interoperabilidad |
| Mantenibilidad | Estabilidad |
| Facilidad de pruebas |
| Facilidad al cambio |
| Usabilidad | Capacidad de aprender |
| Operabilidad |
| Atractivo |

La primera parte de priorización se llevó a cabo mediante el análisis de las posibles arquitecturas existentes que cumplían con cada atributo de calidad y una vez que se tiene definido el peso de cada atributo se determina la mejor arquitectura para el sistema propuesto. Luego de realizar el proceso de comparación entre las arquitecturas, se decide unificar dos arquitecturas “cliente-servidor” (Manejo de datos concurrentes entre varios usuarios) y “capas” (Optimización, reutilización y refinamiento).

## 2.2 Representación arquitectónica

La arquitectura seleccionada para este sistema es la cliente-servidor en capas, la cual se describe a continuación:

Cliente-Servidor en Capas: Modelo basado en dos o más maquinas. Una aplicación es Cliente-Servidor si los datos se almacenan por aparte de la vista presentación. Entonces se podría decir que el servidor es un motor de base de datos y el cliente es el proceso que recupera o crea los procesos. La idea principal es que el servidor permita el acceso a múltiples usuarios a la misma fuente de datos.

### 2.2.1 Cliente

La filosofía del Cliente-Servidor es que cada máquina realiza sus propias tareas. Son diseñados para la gestión compleja de los datos y las comunicaciones para almacenamiento de los datos. Conforme va creciente, se van encontrando diferentes necesidades. Por esto, los clientes son más potentes, siendo capaces de soportar gráficos más robustos y procesamiento de datos y los servidores cuentan con más capacidad de almacenamiento.

Los clientes estarán definidos por las siguientes capas:

* Presentación
* Seguridad
* Lógica del Negocio

#### 2.2.1.1 Presentación

Representado por el Modelo-vista-controlador. El uso de este diseño indica la ventaja para poder utilizar varias vistas en el mismo modelo y la aplicación de diferentes tipos de look and feel en cada vista sin la necesidad de modificar el modelo.

El modelo se encarga de comunicarse con las otras capas que contiene el sistema.

##### 2.2.1.1.1 Modelo-Vista-Controlador

El modelo MVC (Model View Controller), el cual se divide en: entrada, procesamiento y salida. Estas áreas se representan de la siguiente forma:

* Modelo: Administración del comportamiento y los datos del sistema.
* Vista: Administración de la visualización de la información.
* Controlador: Interpretación de las acciones de entrada, informando al modelo y/o a la vista para que se modifiquen.

Ver *Ilustración 1 MVC.*

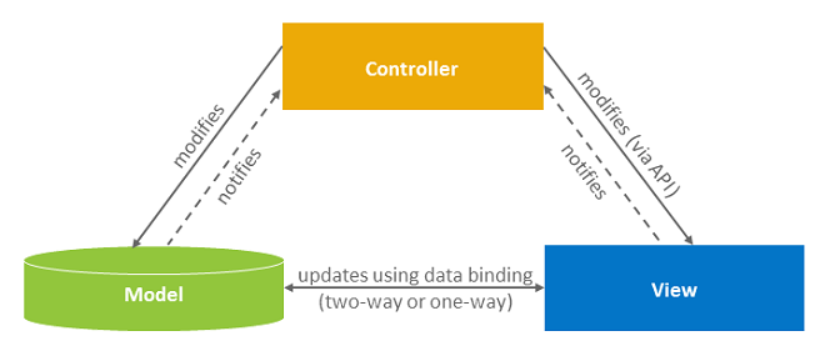


Ilustración 1 MVC

La vista y el controlador necesitan del modelo donde no depende de las otras clases permitiendo la construcción el modelo de forma independiente de la parte visual y modificaciones en el controlador sin afectar la funcionalidad del sistema. Además, la documentación de Patterns & Practices de Microsoft señala las siguientes ventajas:

* Adaptación al cambio: Muchas veces los requerimientos de interfaz de usuario pueden cambiar más rápido que las reglas de negocio.
* Soporte de vistas múltiples: Dado que la vista se separa del modelo y no existe dependencia, la interfaz de usuario en algunas ocasiones puede mostrar diversas vistas de los mismos datos de forma simultánea.

#### 2.2.1.2 Seguridad

Esta capa es la encargada de validar y autenticar a los usuarios que quieran ingresar al sistema, garantizando de esta forma que solo los usuarios registrados puedan utilizar el sistema y hacer uso del mismo.

#### 2.2.1.3 Lógica de Negocio

Esta capa es la que se encarga de realizar el procesamiento, administrando de esta forma el comportamiento y lo datos de la aplicación, es decir, se encarga de la gestión del sistema.

### 2.2.2 Servidor

El servidor se encargará de gestionar la lógica de negocio y del almacenamiento de los datos.

### 2.2.3 Capas

El patrón arquitectónico por capas ayuda a que los sistemas puedan descomponer en grupos las tareas donde cada nivel tiene una abstracción distinta. Por lo tanto, la capa quede definida de la siguiente forma:

* Presentación
* Lógica de la aplicación
* Datos

Estas capas poseen beneficios como el de reutilización (Si el sistema tiene un nivel de abstracción, interfaces documentadas bien definidas facilitan el desarrollo y la cantidad de defectos), soporte de estandarización (Un correcto nivel de abstracción permite el desarrollo de tareas e interfaces estandarizadas) y la posibilidad de intercambios (intercambios de implementaciones distintas de la misma capa que sean equivalentes).

## 2.3 Representación arquitectónica

Este documento representa la arquitectura en las siguientes vistas bajo el Lenguaje Unificado de Modelado (UML):

* La Vista de casos de uso
* La Vista de despliegue
* La Vista lógica
* La Vista de procesos
* La Vista de datos
* La Vista de implementación

Ver *Ilustración 2 Modelo Vista 4+1 de Kruchten*.

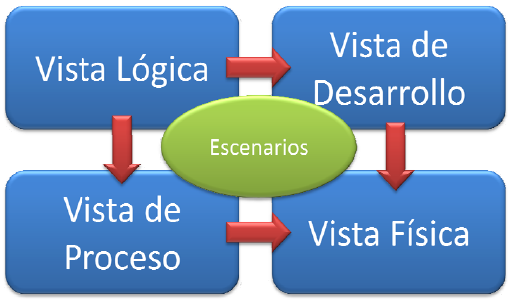


Ilustración 2 Modelo Vista 4+1 de Kruchten

* En la vista lógica se representan los requerimientos funcionales del sistema (Diagramas de clases y de secuencia)
* En la vista de procesos se representan los requerimientos no funcionales de desempeño y disponibilidad.
* En la vista de desarrollo se representa una descomposición en subsistemas.
* En la vista física se representan los requerimientos no funcionales de disponibilidad, confiabilidad, desempeño y escalabilidad.
* Los escenarios son las abstracciones de los requerimientos más importantes. En este caso se van a representar en diagramas de casos de uso.

# 3. Objetivos y restricciones Arquitectónicas

## 3.1 Objetivos

* Cumplir con todos los requerimientos que se describen en el documento de Visión.

## 3.2 Restricciones

* Diseñar la arquitectura de forma flexible para que permita la incorporación de nuevos módulos en el futuro.
* El diseño del sistema se debe hacer de tal forma para que disminuya el código duro.
* Las herramientas tecnológicas utilizadas para el desarrollo del sistema es .NET como entorno de desarrollo y SQL Server como el sistema gestor de base de datos.

## 3.3 Resumen de requerimientos

|  |  |  |
| --- | --- | --- |
| ID | Descripción | Prioridad |
| 1 | Ingresar al sistema | 1 |
| 2 | Realizar solicitud de aula o laboratorio | 3 |
| 3 | Cancelar solicitud de aulas o laboratorios | 8 |
| 4 | Gestión de aulas y laboratorios | 2 |
| 5 | Aprobar solicitud de aulas | 3 |
| 6 | Gestión de noticias | 4 |
| 7 | Gestión de actividades | 4 |
| 8 | Rechazar Solicitud de aulas | 3 |
| 9 | Gestión de Actividades | 2 |
| 10 | Mostrar Noticias y Actividades | 5 |

# 4. Vista de Casos de Uso

Los casos de uso son eventos generales, que describen la interacción entre los actores y el sistema, para una función en específico y en este apartado se detallan los casos de uso del sistema que se pretende desarrollar, en donde se contemplan las funcionalidades de la primera iteración, se visualizan los actores que tendrán interacción con el sistema y las funcionalidades que se le asocian. Dicho diagrama se puede ver en la *Ilustración 3 vista de casos de uso*.

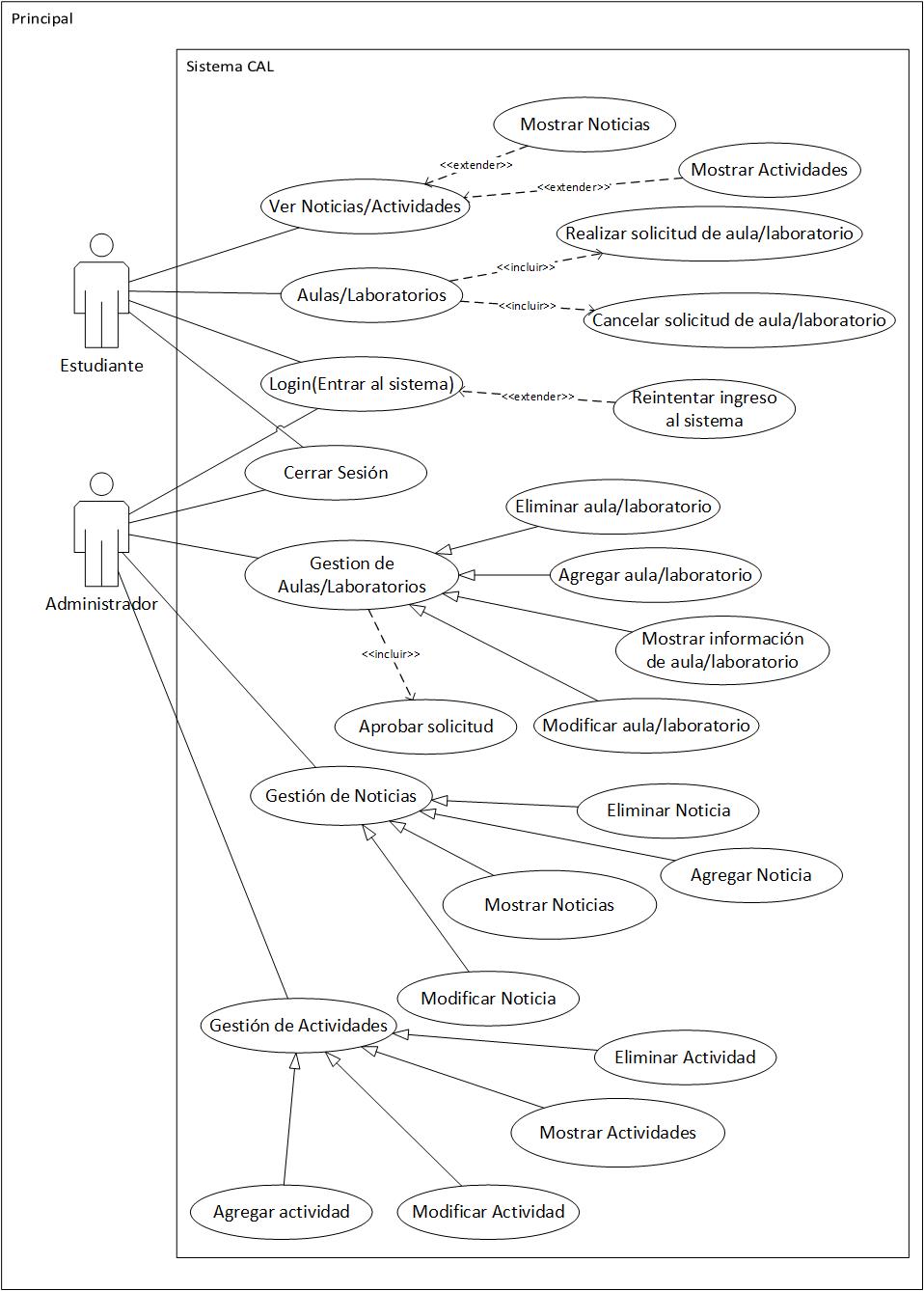


Ilustración 3 Vista de Casos de Uso

Resumen de Sub casos de Uso

|  |  |  |
| --- | --- | --- |
| Id | Caso de uso | Rol |
| 1 | Ingresar al sistema | Estudiante |
| 2 | Registrarse en el sistema | Estudiante |
| 3 | Agregar Aula | Administrador |
| 4 | Modificar Aula | Administrador |
| 5 | Deshabilitar Aula | Administrador |
| 6 | Reservar Aula | Administrador |
| 7 | Aprobar solicitud aula | Administrador |
| 8 | Rechazar solicitud aula | Administrador |
| 9 | Crear Noticia | Administrador |
| 10 | Modificar Noticia | Administrador |
| 11 | Eliminar Noticia | Administrador |
| 12 | Ver Noticias | Estudiante |
| 13 | Crear Actividad | Administrador |
| 14 | Modificar Actividad | Administrador |
| 15 | Eliminar Actividad | Administrador |
| 16 | Ver Actividades | Estudiante |
| 17 | Registrar Usuario | Administrador, Estudiante |
| 18 | Modificar Usuario | Estudiante |
| 19 | Ver Usuarios | Administrador |
| 20 | Realizar Solicitud de Aula | Estudiante |
| 21 | Cancelar Solicitud de Aula | Estudiante |
| 22 | Ver mi Cuenta | Estudiante, Administrador |

# 5. Vista Lógica

En esta vista se representa los requerimientos funcionales del sistema, es decir, se describen las partes del diseño como subsistemas o paquetes. Dicho diagrama se puede ver en la *Ilustración 4 vista Lógica*.

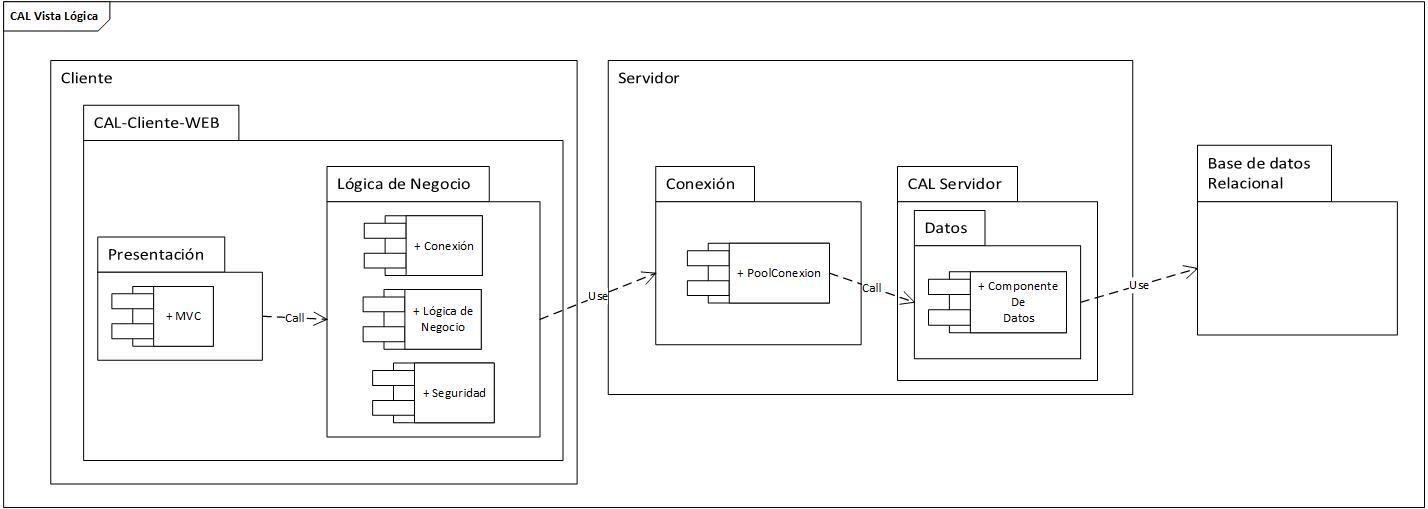


Ilustración 4 Vista Lógica

## 5.1 Visión General

Cabe desatacar que como se explicaba en el apartado 2.2 de este documento la arquitectura es Cliente-Servidor en capas y es por tal motivo que en dicha vista se muestra los módulos: Cliente y servidor. El cliente contiene las capas de presentación, seguridad y la lógica de negocio, mientras que el servidor se encarga del manejo de los datos del sistema y de la comunicación entre los clientes web cuando es requerido el paso de información desde o hacia la base de datos.

## 5.2 Diseño arquitectónico de paquetes importantes

### 5.2.1 Paquete de Presentación

Basado en el MVC, el cual se explicó en la sección *2.2.1.1.1 Modelo-Vista-Controlador*. Dicho paquete se encarga de desplegar la información que es mostrada a los diferentes usuarios del sistema.

### 5.2.2 Lógica de Negocio

Contiene toda la lógica para el modelamiento de las actividades del sistema, por ejemplo, la gestión de las noticias, actividades, usuarios, asistencias, entre otros.

### 5.2.3 Conexión

El paquete de conexión sirve para gestionar las diferentes conexiones entre el cliente y el servidor. Además, como el sistema es Web debe permitir la concurrencia entre varios clientes y el servidor, permitiendo de esta forma el uso por varios usuarios a la misma vez.

### 5.2.4 Datos

Este paquete es el que se encarga de gestionar la salida y entrada de información del sistema CAL a la base de datos, lo cual permite sustentar el funcionamiento del sistema.

# 6. Vista de Procesos

En la siguiente vista se muestra el mapeo de los procesos del sistema como una descomposición de subsistema para que pueda ser desarrollado por un desarrollador de software o grupo de ellos. Tomando en cuenta los requerimientos, la gestión del sistema, reutilización, y las restricciones que impone el lenguaje de programación. Ver *Ilustración 5 Vista de Procesos*

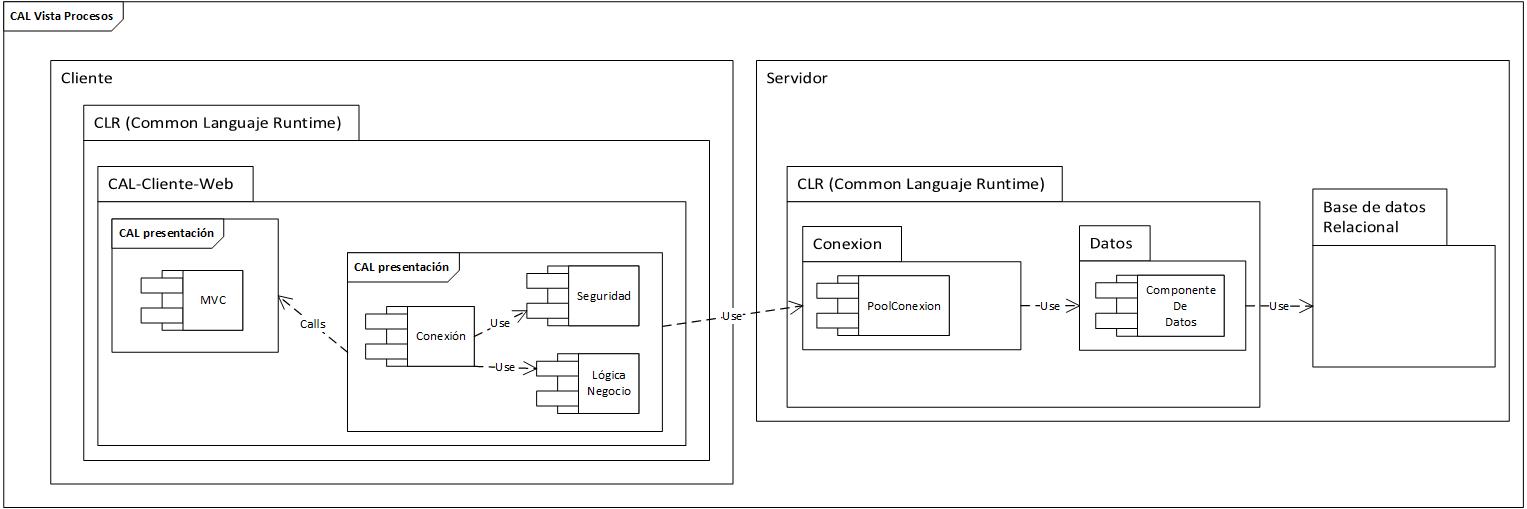


Ilustración 5 Vista de Procesos

El sistema ejecutara constantemente 2 procesos, es decir, el Common Languaje Runtime y a la base de datos SQL.

# 7. Vista de Despliegue

La arquitectura Cliente-Servidor es muy simple y permite que sea de la siguiente manera, ver *Ilustración 6 Vista Despliegue*.

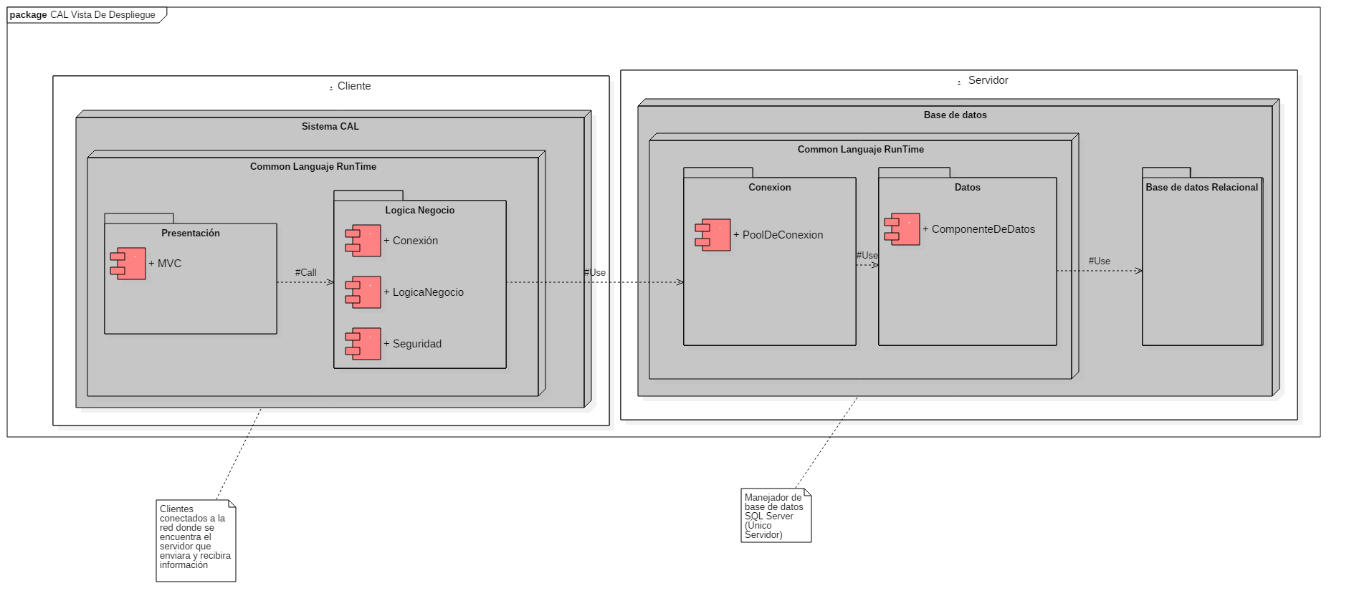


Ilustración 6 Vista Despliegue

Cada usuario podrá conectarse directamente al servidor para el intercambio de los datos y poder funcionar de forma correcta.

# 8. Vista de Implementación

En esta vista se describe visualmente como se implementó el sistema CAL (Capas), como se distribuyen y sus principales funciones. Ver *Ilustración 7 Vista Implementación*.

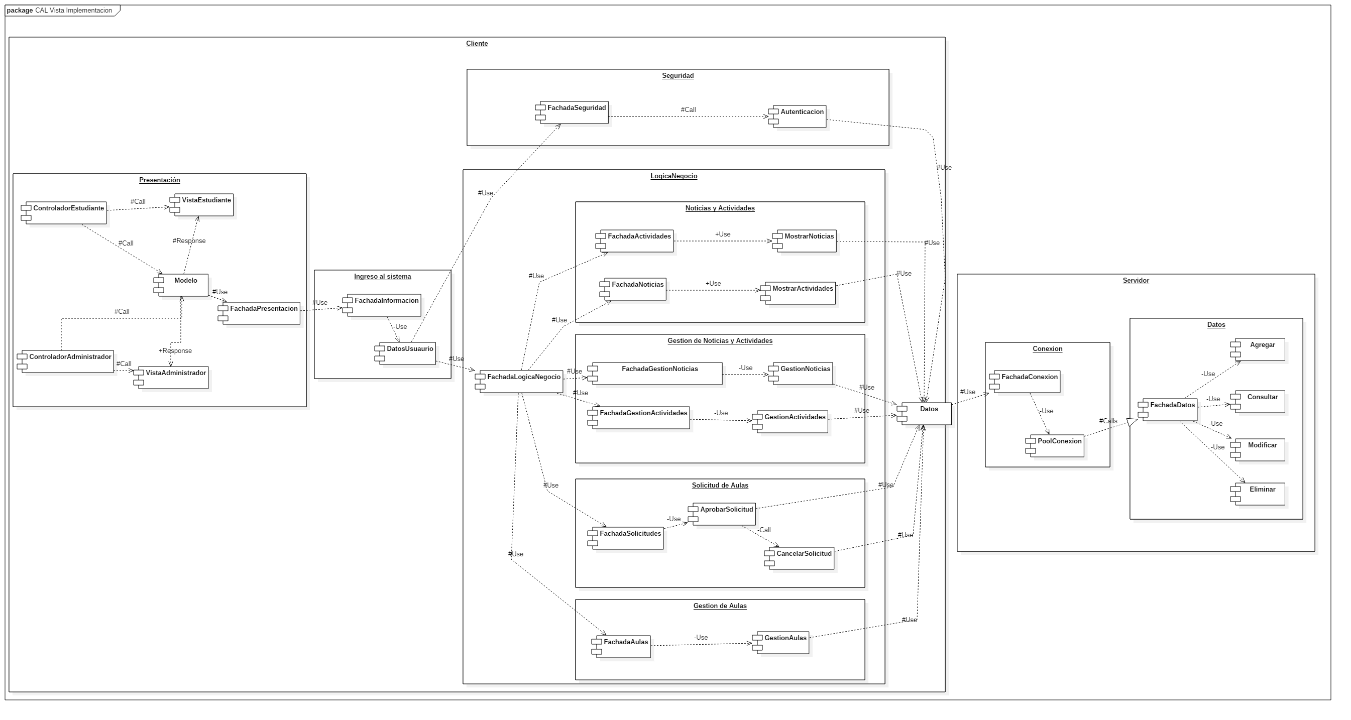


Ilustración 7 Vista Implementación

## 8.1 Visión General

La arquitectura del sistema está desarrollada bajo el modelo cliente-servidor en capas.

## 8.2 Capas

El sistema CAL se implementó en 4 capas, Cada una de esta se describe a continuación:

### 8.2.1 Presentación

Capa encargada de mostrar la información que se asocia a cada usuario, recibir las acciones de los usuarios y enviar sus peticiones a la capa siguiente que se encarga de gestionar los datos.

En *Ilustración 8 Capa Presentación* se puede observar la capa presentación

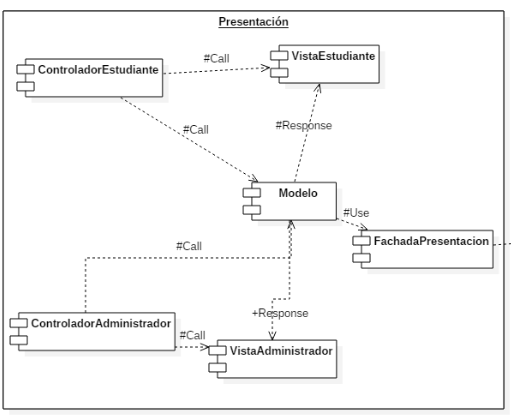


Ilustración 8 Capa Presentación

### 8.2.2 Seguridad

Esta capa se encarga de la validación de usuarios (Usuario y contraseña correctos) y se encarga de enviar la información según el usuario registrado en el momento. Ver *Ilustración 9 Capa de Seguridad*



Ilustración 9 Capa de Seguridad

### 8.2.3 Lógica de Negocio

Capa que se encarga de la gestión del comportamiento y los datos del sistema CAL, respondiendo de esta manera a las peticiones que realizan los usuarios desde la capa de presentación. Ver *Ilustración 10 Capa Lógica del Negocio.*

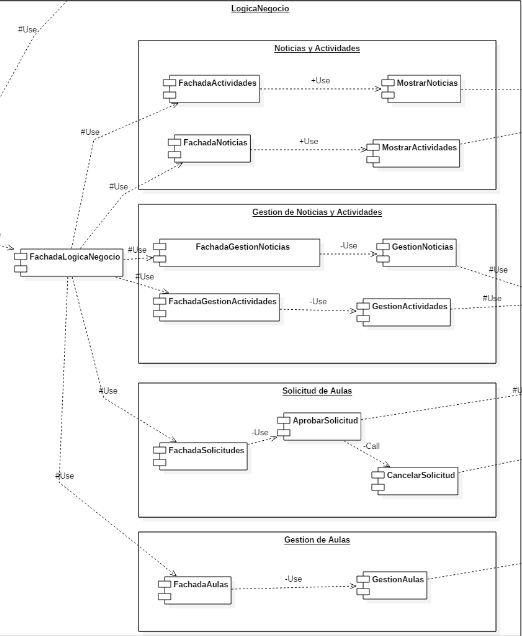


Ilustración 10 Capa Lógica del Negocio

### 8.2.4 Datos

Capa que se encarga del procesamiento de la información desde y hacia la base de datos para que esta capa funcione bien, también es la que se encarga de ingresar a la base de datos CAL. Ver *Ilustración 11 Capa de Datos.*

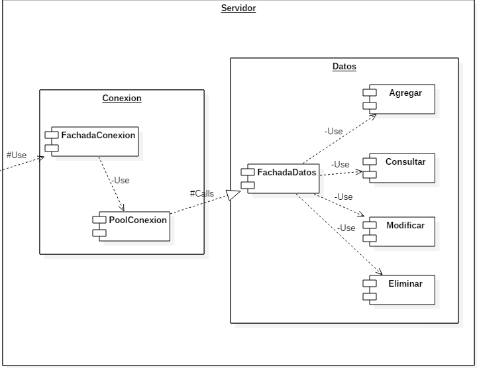


Ilustración 11 Capa de Datos

# 9. Vista de Datos

En esta vista se describe de manera breve como fue almacenada la información en la base de datos SQL Server. Esta vista es importante ya que, aquí es donde se gestionan los datos del usuario. En dicha vista se puede observar el diagrama entidad-relación *Ilustración 12 vista de Datos*, se pueden visualizar las tablas que se utilizaron para la persistencia de la información.

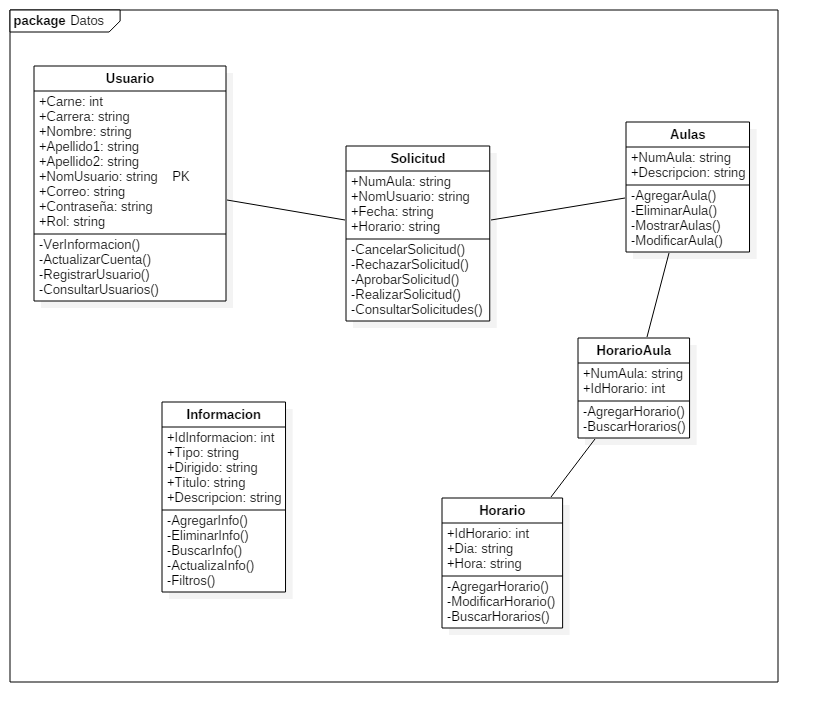


Ilustración 12 Vista de Datos

# 10. Tamaño y desempeño

Características generales del sistema que describen como impacta en el desempeño y la arquitectura del software.

## 10.1 Concurrencia de usuarios

El sistema CAL en primera instancia está capacitado para soportar los usuarios del Centro Académico de Limón.

# 11. Calidad

Descripción de como la arquitectura a las características que no son funcionales del sistema

## Seguridad

### Despliegue de la información

La información que se despliega se hace conforme al tipo de usuario, mostrando de esta forma la información apropiada para cada usuario.

### Validación de la información del usuario

El sistema CAL asegura que la información de cada usuario sea privada, ya que pueden existir personas con malas intenciones que trataran de borrar o cambiar la información que se encuentre registrada en el sistema.

## Usabilidad

### Visuales

El sistema CAL permite la gestión de la información a través de una interfaz simple y agradable con el Usuario.