***Juan Carlos Hurtado Morales***

***Ricardo Fabián Juárez Santoyo***

***Miguel Alejandro Vazquez Alberto***

***Efrain Jared Ramírez Díaz***

|  |
| --- |
| *Documento de Arquitectura del software* |

Índice de Contenido

1 Introducción 3

2 Resumen Arquitectónico 3

2.1 Estilo Arquitectónico 3

2.2 Objetivos de la Arquitectura 3

3 Diagrama de Componentes. 4

3.1 VISTA. 5

3.2 DATOS. 6

3.3 COMUNES 6

3.4 MODELO 6

4 Vista de Casos de Uso 7

5 Diagrama de despliegue. 7

5.1 Visión General 7

6 Diagrama de Base de Datos 8

**Documento de Arquitectura del Software**

# Introducción

En el siguiente documento se presenta la arquitectura del proyecto de Mini Sice, se abordan temas como el resumen arquitectónico, el cual está dividido en estilo arquitectónico en el cual definimos que vamos a utilizar el estilo de capas; el otro tema son los objetivos de la de la arquitectura en el cual describimos como influencian los requisitos funcionales y no funcionales.

También se incluyen los diagramas de componentes que son herramientas que nos sirven para tener una visión general y amplia de la funcionalidad de la partición de los módulos o componentes del sistema. Es una forma muy visual de ver cómo funcionan y se relacionan los componentes del sistema.

El documento en formato original hace mención a los casos de uso, pero el equipo decidió trabajar con historias de usuario, esto para tratar de hacer más fácil la redacción y el entendimiento. Por último, se adjunta el diagrama de la base de datos que será utilizado en el proyecto, tanto de forma visual para ver la relación de las tablas y una descripción en una tabla de cada una de las tablas de la base de datos.

# Resumen Arquitectónico

## Estilo Arquitectónico

Este sistema será desarrollado haciendo uso de la arquitectura por capas, al dividir la funcionalidad del mismo en la capa de aplicación, negocio y Datos. En la capa de aplicación se desarrollara el entorno grafico por el cual interactuará directamente con el usuario, la capa de negocio será dedicada a el comportamiento lógico de la comunicación entre la aplicación y el servidor de datos, además de las validaciones en los formularios que estarán a disposición del usuario, y por último en la capa de Datos se gestionara directamente la conexión con la base de datos que contendrá la información generada a través del sistema.

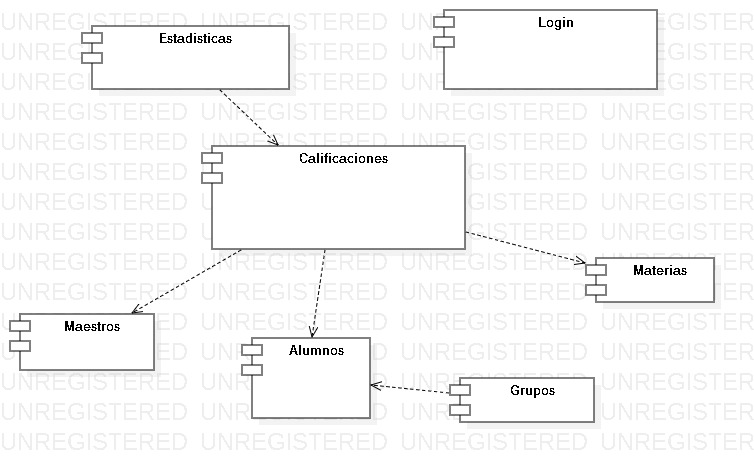
## Objetivos de la Arquitectura

La presente arquitectura pretende cubrir los objetivos de calidad esperados, siendo estos los siguientes:

|  |  |  |  |
| --- | --- | --- | --- |
| **Atributo de calidad** | **Rango Requerido** | **Motivo** | **Estrategias para lograr el objetivo** |
| Compatibilidad distintos navegadores | Chrome, Firefox | Flexibilidad para el usuario final | * Utilizar lenguajes de programación que tengan soporte para estos navegadores * Estandarizar la codificación de los procesos |
| Funcionamiento estable | 30 usuarios simultáneos | Evitar problemas de conexión al servidor | * Optimizar las consultas a la base de datos a través de:   + El uso de vistas   + Simplificar las consultas * Utilizar servicios de hosting que satisfagan las necesidades del software |
| Escalabilidad de plataforma | Móvil, escritorio | Posible compatibilidad con distintas plataformas (móvil o escritorio). | * Uso de notación JSON en el intercambio de información * Implementar diseño responsivo |

# Diagrama de Componentes.

En esta sección se detallan de qué manera están organizados los componentes del sistema y se da una descripción de cada uno de los componentes así como la tecnología a utilizar en cada uno de estos.



A continuación describiremos cada componente:

| C\_01 Estadísticas | |
| --- | --- |
| Descripción: | Componente que muestra al alumno con el mejor promedio después de cada periodo, los maestros y administradores son los únicos usuarios que tienen permiso para ver este componentes. |
| Requerimientos: | HU\_09 Estadísticas. |
| Interfaces Disponibles: | Este componente tiene relación con C\_02 Ya que estará presente en algunas de las funcionalidades relacionadas con dicho componente. |

| C\_02 Calificaciones | |
| --- | --- |
| Descripción: | Componente que permite al maestro registrar calificaciones para el alumno. La relación del alumno con la calificación es que este podrá consultar la calificación en el sistema. |
| Requerimientos: | HU\_07 Calificaciones Alumnos  HU\_10 Calificaciones Maestros |
| Interfaces Disponibles: | Este componente tiene relación con Maestros y alumnos ya que cada uno de estos tendrá sus propias funcionalidades |

| C\_03 Alumnos | |
| --- | --- |
| Descripción: | En este componente permite al administrador registrar, consultar y editar alumnos. |
| Requerimientos: | HU\_08 Alumnos (Administrador) |
| Interfaces Disponibles: | Este Componente proporcionara la información necesaria para que C\_02 y C\_04 funcionen de la manera establecida. |

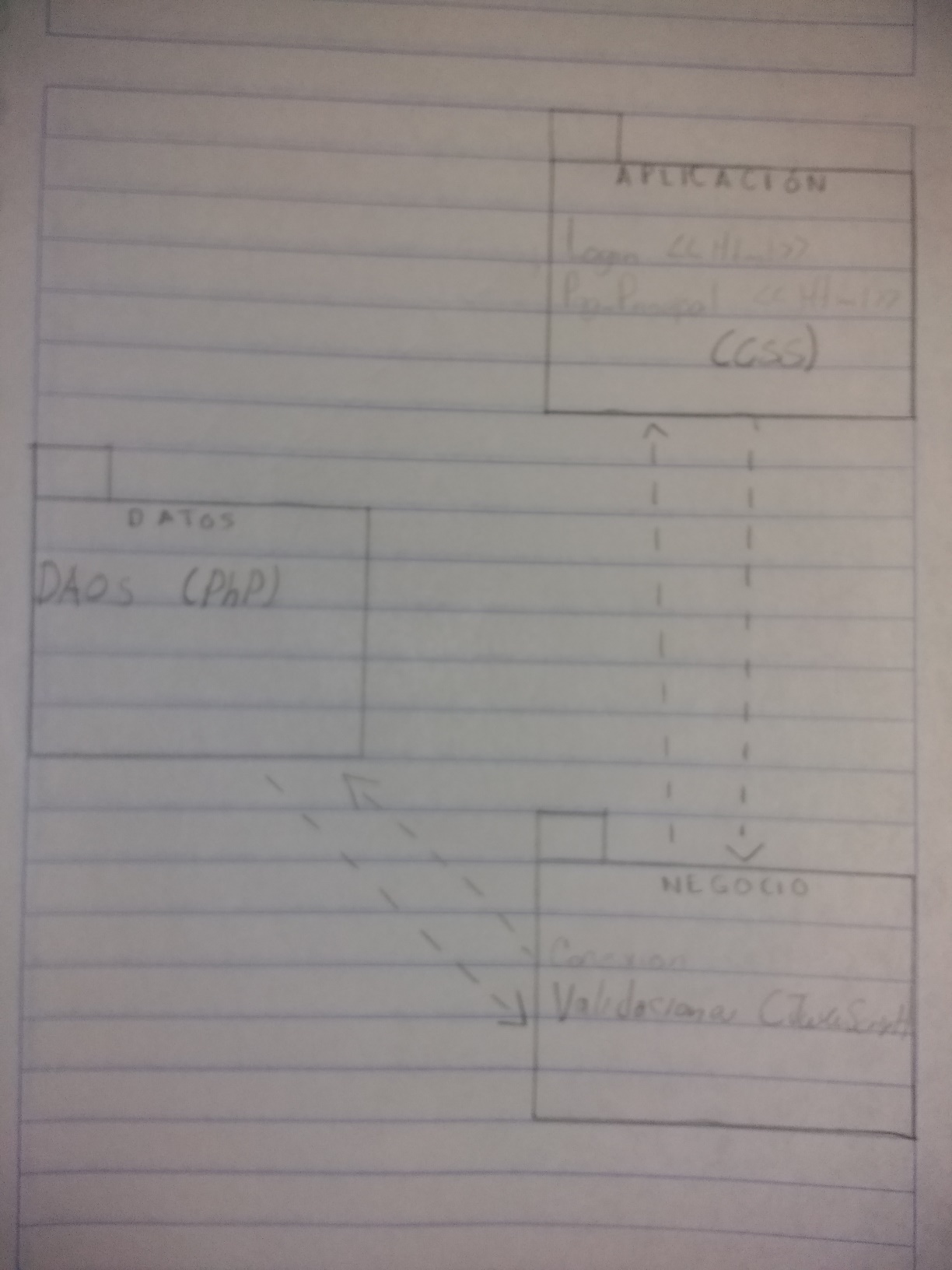
| C\_04 Grupos | |
| --- | --- |
| Descripción: | Este componente contiene la funcionalidad relacionada con los grupos de alumnos definidos por grado y grupo. |
| Requerimientos: | HU\_02 Asignación de Grupos  HU\_06 Generación de Grupos |
| Interfaces Disponibles: | Este componente depende de C\_03 ya que utiliza la información generada por este |

| C\_05 Maestros | |
| --- | --- |
| Descripción: | En este componente se encontraran las vistas necesarias para la gestión de Maestros, el administrador podrá consultar, editar y eliminar maestros. |
| Requerimientos: | HU\_03 Maestros  HU\_10 Calificaciones Maestro |
| Interfaces Disponibles: | Este Componente proporcionara la información necesaria para que C\_01 funcione de la manera establecida. |

| C\_06 Materias | |
| --- | --- |
| Descripción: | En este componente se encontraran las vistas necesarias para la gestión de Materias, las materias vienen predefinidas dependiendo del grado escolar. |
| Requerimientos: | HU\_05 Materias |
| Interfaces Disponibles: | Este Componente proporcionara la información necesaria para que C\_01 funcione de la manera establecida. |

| C\_07 Login | |
| --- | --- |
| Descripción: | A través de este componente será posible la autenticación del usuario que esté utilizando el sistema para acceder a las funcionalidades ofrecidas conforme al tipo de usuario autenticado. |
| Requerimientos: | HU\_01 Login |
| Interfaces Disponibles: | Este componente es independiente sin embargo es indispensable para que el sistema brinde las funcionalidades requeridas |

Por otra parte, cada uno de estos componentes deberá estar diseñado de manera que pueda ser conectado, mejorado y ampliado independientemente, por lo que se sugiere que cada componente se diseñe con base en la división lógica de capas: Aplicación, negocio y datos, con la siguiente distribución:



A continuación se describen cada uno de ellos:

## DATOS.

| C-10: Acceso a Datos | |
| --- | --- |
| Descripción: | * Es la interacción con la base de datos en el lenguaje PHP |
| Requerimientos: | Creación de base de datos |
| Interfaces Disponibles: | Negocio |

## NEGOCIO

| C-20: Útil | |
| --- | --- |
| Descripción: | Es la interacción entre las interfaces con lo que se debe de guardar o pedir en las bases de datos |
| Requerimientos: | Saber usar en lenguaje de JavaScript |
| Interfaces Disponibles: | Aplicación y Datos |

## APLICACIÓN

| C-30: Interfaces | |
| --- | --- |
| Descripción: | * Es la interacción máquina-humano ayuda a la captura y/o muestra de datos en HTML acompañado de estética con CSS |
| Requerimientos: | Saber el uso de HTML con CSS para la creación de las interfaces para el usuario |
| Interfaces Disponibles: | Datos |

En resumen, el presente proyecto se desarrollará bajo las siguientes tecnologías:

* **HTML 5.** Es el lenguaje para la base principal del proyecto, ya que se desarrollará una página web.
* **CSS.** Este lenguaje es útil para darle estilo a la página web, es para darle un buen estilo de visto al usuario.
* **MySQL 6.2.5**. Para parte de la persistencia se utilizara dicho gestor de base de datos, ya que no tiene costo de licencia por el uso comercial del mismo.
* **JavaScript**. Es para la parte lógica del proyecto, es el lenguaje de programación con la que se va a programar esta parte del proyecto.
* **PHP**. Servirá para trabajar del lado del servidor.

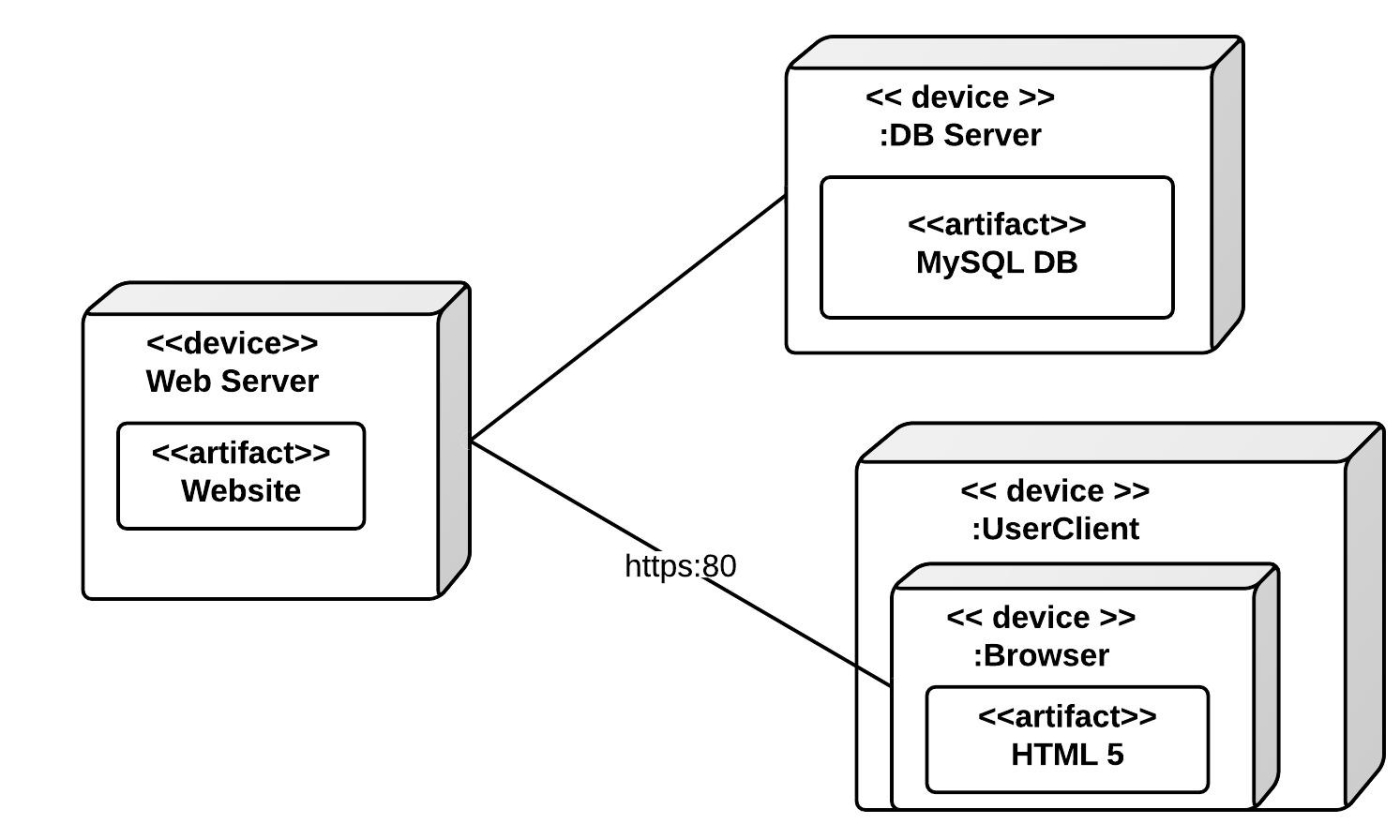
# Vista de Casos de Uso

*En nuestro caso no trabajamos con casos de uso, nuestro proyecto es con Historias de usuario, las cuales se encuentran en el documento de historias de usuario.*

# Diagrama de despliegue.

## Visión General

A continuación se mostrará el diagrama de despliegue de cómo se implantará el sistema:



En el diagrama se muestra la distribución física requerida para implementar la aplicación y se muestran los componentes requeridos en el lado del cliente y en el lado del servidor.

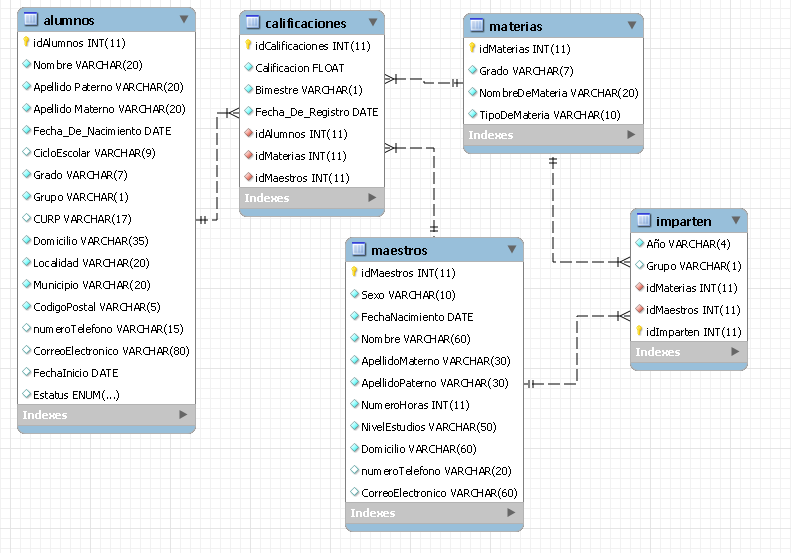
# Diagrama de Base de Datos

A continuación se lista el modelo de datos del sistema de manera parcial, con el motivo de que éste será refinado mediante el desarrollo iterativo del mismo.

**Nombre de la base de datos**: Escuela

**SGBD:** MySQL 6.3.5

**Diagrama Relacional**



|  |  |
| --- | --- |
| **Rastreabilidad con los Requerimientos** | |
| **Tabla** | **Requerimiento(s) relacionados.** |
|  | |
| **NOMBRE DE LA BD** | |
| **Alumnos** | **Login, Asignación de grupos, Materias, Calificaciones, Consultar** |
| **Calificaciones** | **Materias, Alumnos y Maestros** |
| **Maestros** | **Alumnos, Grupos, Materias, Calificaciones** |
| **Materias** | **Alumnos, Maestros y Grupos** |
| **Imparten** | **Alumnos, Maestros y Grupos** |