

<b>System Architecture Description of SyncBoard</b>		
<b>Doc #</b>	<b>Version: 2.0</b>	<b>Page 1 / 15</b>

System Architecture Description of SyncBoard		
Doc #	Version: 2.0	Page 2 / 15

## **TABLE OF CONTENTS**

### **1 Introducción**

#### ***1.1 Visión de la arquitectura***

#### ***1.2 Abreviaturas y Glosario***

##### *1.2.1 Abreviaturas*

##### *1.2.2 Glosario*

#### ***1.3 Referencias***

##### *1.3.1 Referencias del Proyecto*

##### *1.3.2 Referencias Standard y Regulatorias*

#### ***1.4 Convenciones***

### **2 Arquitectura**

#### ***2.1 Descripción General de la Arquitectura***

#### ***2.2 Descripción general de la Arquitectura Lógica***

##### *2.2.1 Descripción Componente 1 de Software*

##### *2.2.2 Descripción Componente 2 de Software*

##### *2.2.3 Descripción Componente 3 de Software*

#### ***2.3 Descripción General de la arquitectura física***

##### *2.3.1 Descripción del Componente de Hardware 1*

##### *2.3.2 Descripción del Componente de Hardware 2*

##### *2.3.3 Descripción del Componente de Hardware 3*

#### ***2.4 Software COTS***

### **3 Dynamic behaviour of architecture**

#### ***3.1 Workflow / Sequence 1***

#### ***3.2 Workflow / Sequence 2***

### **4 Justification of architecture**

#### ***4.1 System architecture capabilities***

#### ***4.2 Network architecture capabilities***

#### ***4.3 Risk analysis outputs***

#### ***4.4 Human factors engineering outputs***

### **5 Requirements traceability**

System Architecture Description of SyncBoard		
Doc #	Version: 2.0	Page 3 / 15

## 1 Introduction

### 1.1 Document overview

Muchas veces Observamos cierta dificultad presencial o virtual para tomar apuntes de diferentes formas mientras nuestro profesor dicta una clase ,Imaginemos que no tengamos que preocuparnos por tomar apuntes antes de que el profesor borre lo hecho en el tablero, y en vez, que pudiéramos prestar atención a clase, sin necesidad de preocuparnos por estar tomando apuntes en todo momento, o el día que no pudiste estar en una clase, tengas acceso a todo lo hecho ese día. Aún mejor, que no tengas que usar un cuaderno y aún así no perderte de ningún apunte y puedas accederlos desde cualquier varios dispositivos electrónicos. Si tuviéramos un software capaz de grabar y de sincronizar cualquier interacción hecha desde un tablero digital, podríamos lograr esto, y es exactamente lo que SyncBoard es capaz de hacer. SyncBoard es un software que pone a disposición un tablero como ayuda visual para dictar una clase, al mismo tiempo SyncBoard graba cualquier interacción hecha por el profesor u otra persona en dicho tablero y la sincroniza con los apuntes de cada uno de los asistentes. Sin embargo, si como asistente necesitas tomar tus propios apuntes sobre lo que estás viendo en el tablero, también lo puedes hacer y este no será compartido con nadie, a menos que más adelante lo quisieras hacer. SyncBoard además facilita las interacciones que demandan una participación de los asistentes, como por ejemplo, cuándo tienes que “pasar al tablero” o cuando el profesor pide un trabajo o ejercicio en grupo.

### 1.2 Abbreviations and Glossary

#### 1.2.1 Abbreviations

#### 1.2.2 Glossary

- REST: las operaciones más importantes relacionadas con los datos en el sistema REST y las especificaciones HTTP que se utilizaran son cuatro : POST(crear) , Get(Leer) , PUT(editar) , DELETE( eliminar).
- SyncBoard: Tablero de clases mediante el cual se sincronizarán apuntes de un profesor a estudiante.
- DOM: Modelo de Objetos para la representación de documentos.
- NGINX : Archivo de Configuración para Balanceo de Cargas.

### 1.3 References

#### 1.3.1 Project References

#	Document Identifier	Document Title
[R1]	<a href="https://smarkkapp.com/en">https://smarkkapp.com/en</a>	Aplicación en la cual se basa el proyecto Syncboard Smarkkapp.com

## System Architecture Description of SyncBoard

Doc #

Version: 2.0

Page 4 / 15

#	Document Identifier	Document Title
[R2]	<a href="https://html5up.net/">https://html5up.net/</a>	Plantillas de html 5 up.

#	Document Identifier	Document Title
[R3]	<a href="http://www.aprendeprogramar.com/index.php?option=com_content&amp;view=article&amp;id=520:formulario-s-html-form-label-name-value-id-ejemplos-checkbox-option-button-combobox-cu00720b&amp;catid=69&amp;Itemid=192">http://www.aprendeprogramar.com/index.php?option=com_content&amp;view=article&amp;id=520:formulario-s-html-form-label-name-value-id-ejemplos-checkbox-option-button-combobox-cu00720b&amp;catid=69&amp;Itemid=192</a>	Link de formularios html para las vistas

### 1.3.2 Standard and regulatory References

#	Document Identifier	Document Title
[STD1]		Add your documents references. One line per document

### 1.4 Conventions

No aplica.

2

System Architecture Description of SyncBoard		
Doc #	Version: 2.0	Page 5 / 15

## Architecture

### 2.1 Architecture overview

**Perspectiva de Usuario:** Este es un software para uso organizacional al momento en que tengan que desarrollar una clase se comuniquen con una interfaz de tablero o canvas como tal y este nos genera una secuencia de puntos directa entre Profesor-Estudiante , aparte de esto la principal función es generar apuntes como tal y estos se podrán habilitar o deshabilitar para los estudiantes.

Los principales usuarios de Sync Board son el Profesor y el estudiante quienes serán como anteriormente lo mencionamos los que nos generarán apuntes dentro de este mismo , lo cual allí facilitamos el control de actividades para que nuestros apuntes puedan ser más efectivos a la hora del aprendizaje.

Nuestra aplicación maneja tres interfaces : La del Profesor , la del empleado y la del Tablero , la diferencia de las dos primeras radica en los privilegios que cada uno de estos roles tiene dentro del sistema , claramente el profesor tiene más privilegios que el estudiante porque debe tener control sobre las clases que va a dictar , y una posible sincronización de estos mismos apuntes que se van a generar por clase y estudiante suscrito a cada clase.

**Perspectiva del desarrollador:** Esta aplicación web fue desarrollada bajo los principios de API rest , arquitectura que como tal se apoya en el protocolo HTTP en donde cualquier cliente con este mismo se podrá comunicarse y también el uso de URLs permite acceder a cada una de las páginas del sitio web o recurso donde el usuario tendrá potestad de crear o dibujar en el tablero de Sync Board con las respectivas operaciones dentro de este protocolo (GET, POST , PUT , DELETE), también se utilizan conceptos de la arquitectura STOMP y webSockets , para que la aplicación logrará una comunicación bidireccional entre el cliente y el servidor logrando acciones en tiempo real desde varios equipos , todo esto mediante un mediador de mensajes ligadas a una serie de operaciones para dicha interacción “Connect , subscribe , send , Disconnect”.

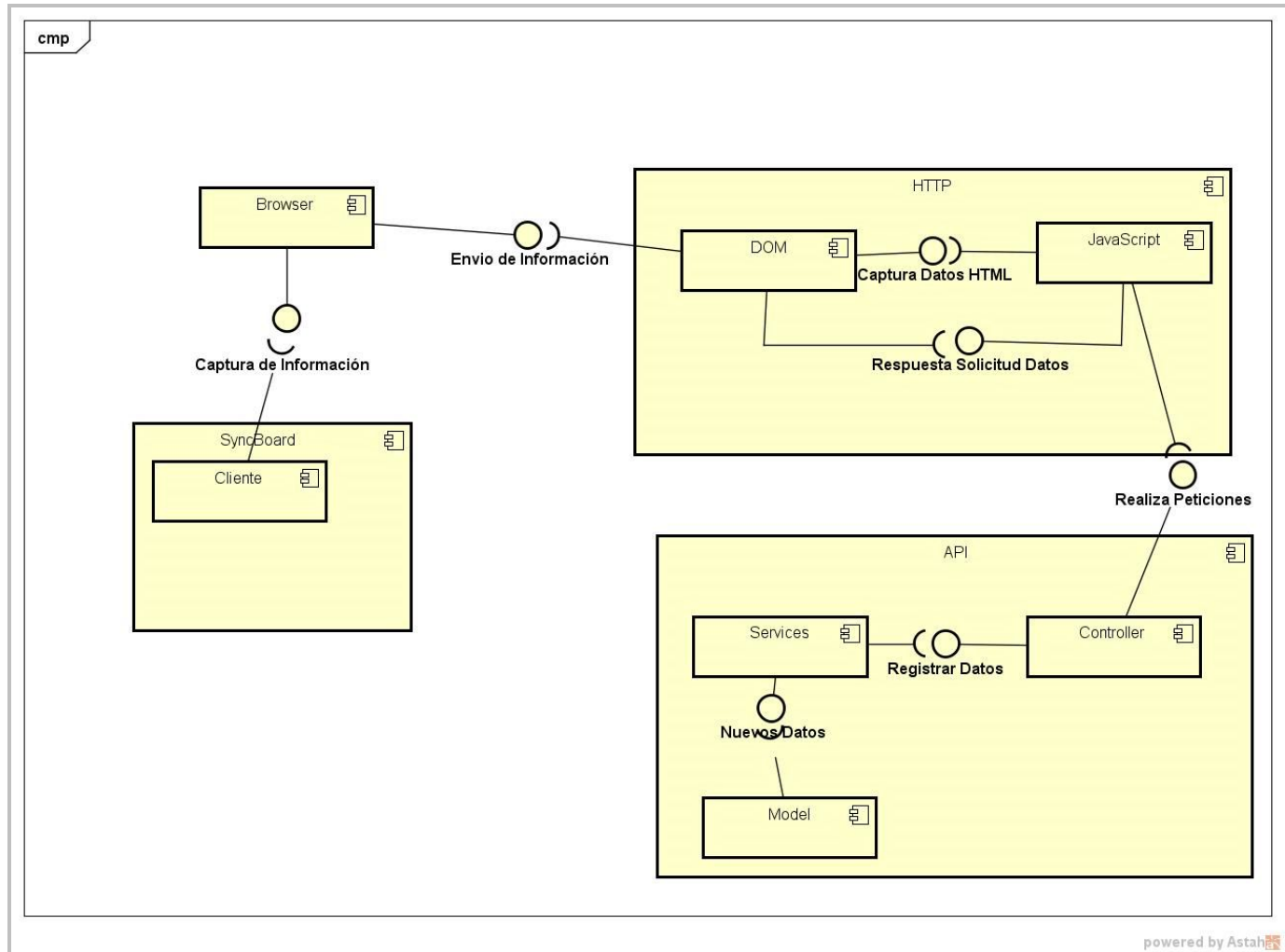
### 2.2 Logical architecture overview

## System Architecture Description of SyncBoard

Doc #

Version: 2.0

Page 6 / 15



System Architecture Description of SyncBoard		
Doc #	Version: 2.0	Page 7 / 15

### 2.2.1 Software Component 1 description

**Cliente:** Nuestro cliente o usuario es aquel que está dentro de Nuestro Aplicativo Sync Board que es aquel donde se registrará ya sea como estudiante o como profesor para poder interactuar Con nuestro sistema de Clases , es decir que cada usuario de tipo profesor será el que cree una Clase y de allí nuestro Browser captura la información respectiva.

### 2.2.2 Software Component 2 description

**Browser:** Como tal el browser es un sistema de navegación que captura la información o permite el manejo de esta misma, y como tal permite el acceso a la web que es donde nuestra aplicación está corriendo en HEROKU(PAAS).

### 2.2.3 Software Component 3 description

**DOM:** Como tal es la estructura de Objetos que genera el navegador cuando vamos a cargar un documento y este se puede alterar mediante JavaScript para cambiar dinámicamente los contenidos y aspecto de la página , es decir que también los usuarios se crean y podrán interactuar con un tablero dibujando.

### 2.2.4 Software Component 4 description

**JavaScript:** Es un lenguaje de programación utilizado para crear páginas web dinámicas donde incorpora efectos como texto y elementos que aparecen y desaparecen acciones acciones o eventos después de pulsar un botón y ventanas u otra páginas emergentes.

### 2.2.5 Software Component 5 description

**Controller:** En este paquete se encuentra las clases que definimos en nuestra aplicación para implementar las operaciones que se solicitan desde el javascript logrando así una comunicación entre la interfaz y lo lógico para que pueda traer y devolver datos , y así estos se puedan mostrar en el html.

### 2.2.6 Software Component 6 description

**Services:** Aquí encontramos las clases que tiene la labor de implementar las operaciones internas que hacen las peticiones HTTP par que el controlador solo requiera llamar estas actividades y devolver la información al JavaScript , logrando una comunicación entre la parte lógica mediante métodos que permiten modificar sus datos , es decir nuestros tópicos que se van requiriendo en el proceso del profesor y del estudiante también.

### 2.2.7 Software Component 7 description

**Lógica:** en este paquete se encuentra todas las clases con sus respectivos atributos y operaciones internas que le definimos para lograr el correcto comportamiento de la aplicación , estas clases son invocadas por el services donde allí realizan el tratamiento con los datos y logran el funcionamiento de todo.

Las principales clases son :

## System Architecture Description of SyncBoard

Doc #

Version: 2.0

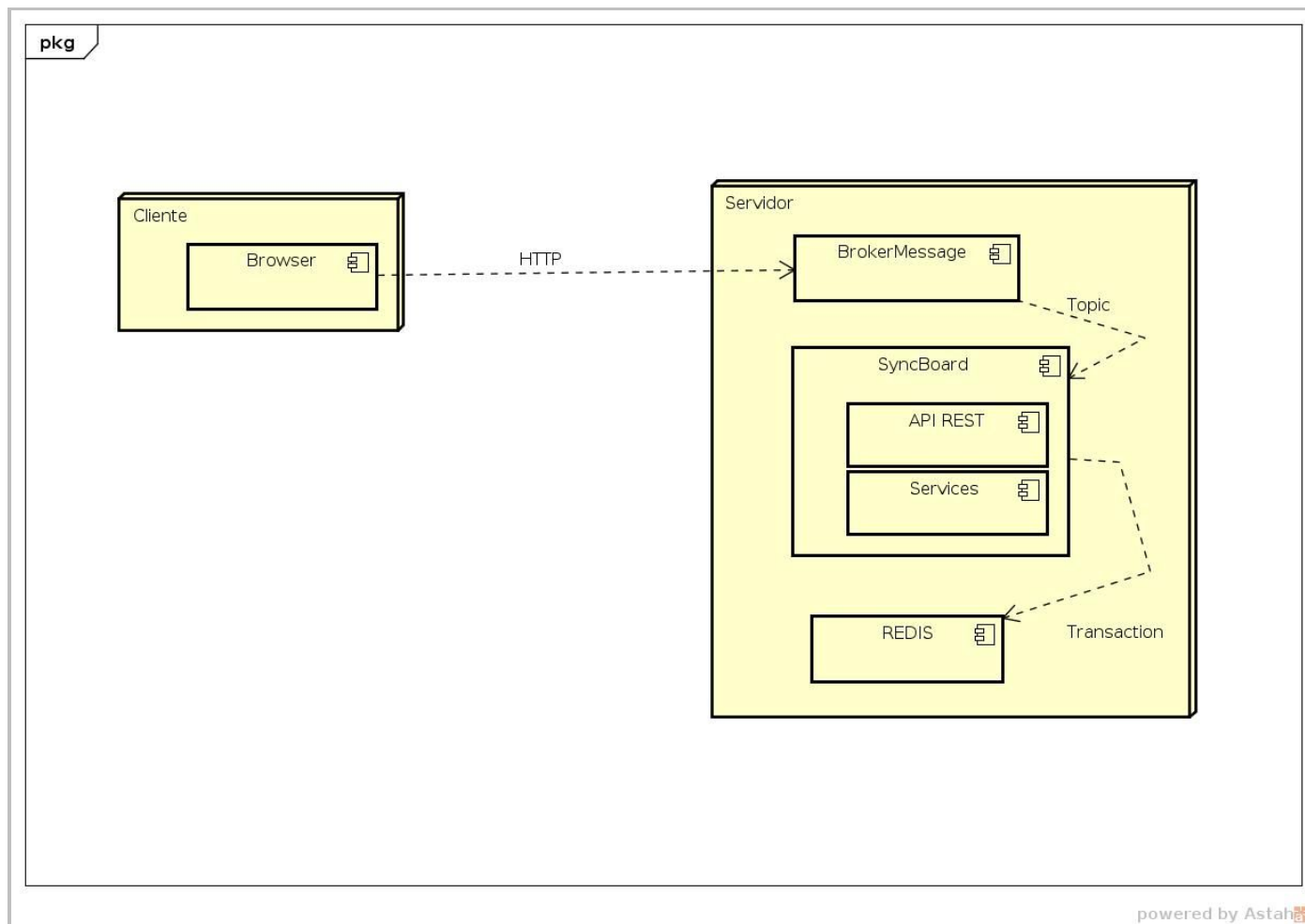
Page 8 / 15

**Tablero :** En esta clase se asocian las tareas correspondientes y las personas que se encuentran suscritas a nuestro tablero ya que el profesor es el que la crea , es decir solo miramos que estudiantes están inscritos a la clase que se está dictando , aparte cada estudiante puede dibujar libremente y el profesor tendrá la opción de habilitar o deshabilitar los apuntes para sus estudiantes inscritos.

**Usuario:** Esta clase define las características que debe poseer un usuario como nombre y demás información sobre el ya que esta implementa la otra clase tablero que requiere la información del profesor para poder ser creada la clase , y también la información de un estudiante para que pueda inscribirse a una clase y así poder tener su vista de tablero.

**Clase:** La clase es la que va a poseer nuestro tablero con una secuencia de dibujos es decir para que allí podamos dibujar como si estuviéramos dictando una clase en real , hablando más concretamente poseerá el profesor y estudiante un tablero con la respectiva información para que no haya inconsistencias.

### 2.3 Physical architecture overview



powered by Astah



System Architecture Description of SyncBoard		
Doc #	Version: 2.0	Page 9 / 15

### 2.3.1 Hardware Component 1 description

**NGINX:** Este es un servidor HTTP de código abierto que también incluye servicios de correo electrónico y demás , lo más importante es que está listo para ser utilizado como un proxy inverso o en reversa , como tal también su principal funcionalidad es que se utiliza para equilibrar la carga entre servidores back-end o para proporcionar almacenamiento en caché para un servidor back-end lento.

### 2.3.2 Hardware Component 2 description

**Broker:** El broker de mensajería es el que permite que nuestra aplicación sea asincrónica y de acoplamiento flexible en la que componentes independientes funcionan conjuntamente para llevar a cabo una tarea , estos componentes de aplicación son mensajes que contienen la información necesaria para terminar una tarea.

### 2.3.3 Hardware Component 3 description

**Sync Board:** Es nuestra aplicación que por dentro tiene el componente de Cliente que es aquel que utiliza un usuario independientemente sea estudiante o profesor para interactuar con nuestro tablero.

### 2.3.4 Hardware Component 4 description

**REDIS:** Para hacer el caché de la aplicación utilizamos redis el cual es un pequeño motor de base de datos basado en el almacenamiento en tablas de hashes pero que opcionalmente puede ser usada como una base de datos durable o persistente , en el que se guarda la información que es utilizada con mayor frecuencia en la aplicación.

## 2.4 Software COTS

## System Architecture Description of SyncBoard

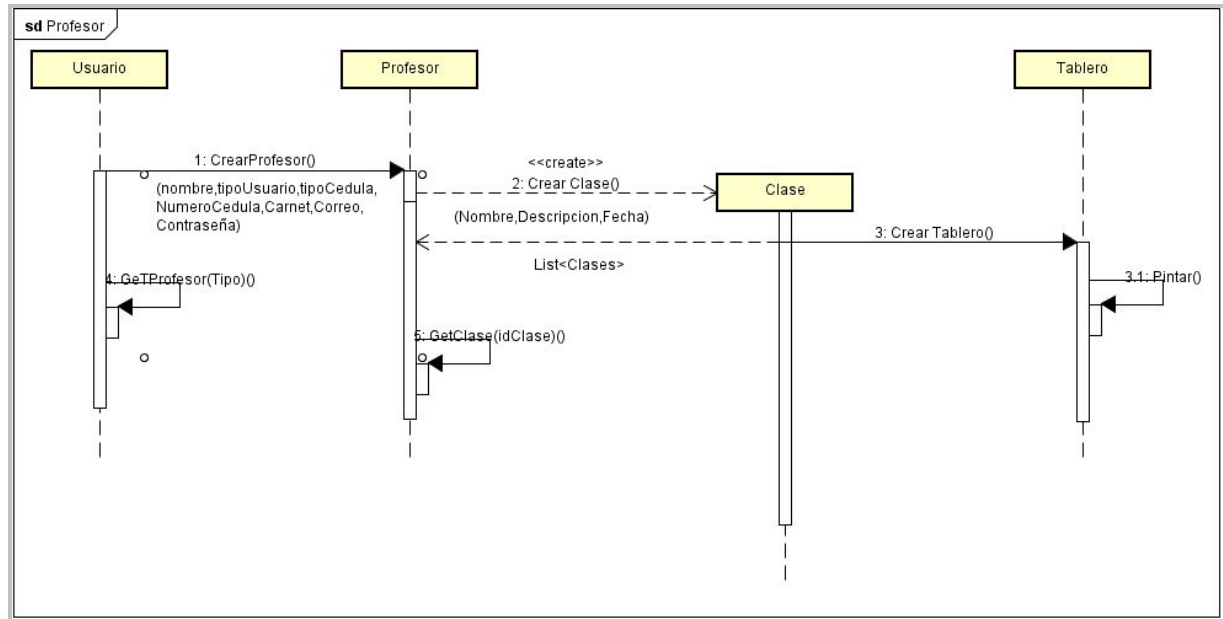
Doc #

Version: 2.0

Page 10 / 15

### Dynamic behaviour of architecture

#### 3.1 Workflow / Sequence 1



El Usuario tendrá la posibilidad de crear un Profesor el cual contiene parámetros como nombre, tipo de usuario, cedula\_numero, cedula\_tipo, carnet, correo y pass y de hacer las petición GET del Profesor que él requiera, mientras el Profesor tiene la posibilidad de crear una Clase, la cual retornara la lista de las Clases existentes al Profesor y cada Clase tendrá la posibilidad de crear un Tablero en el cual, el Tablero podrá repitan sobre el mismo.

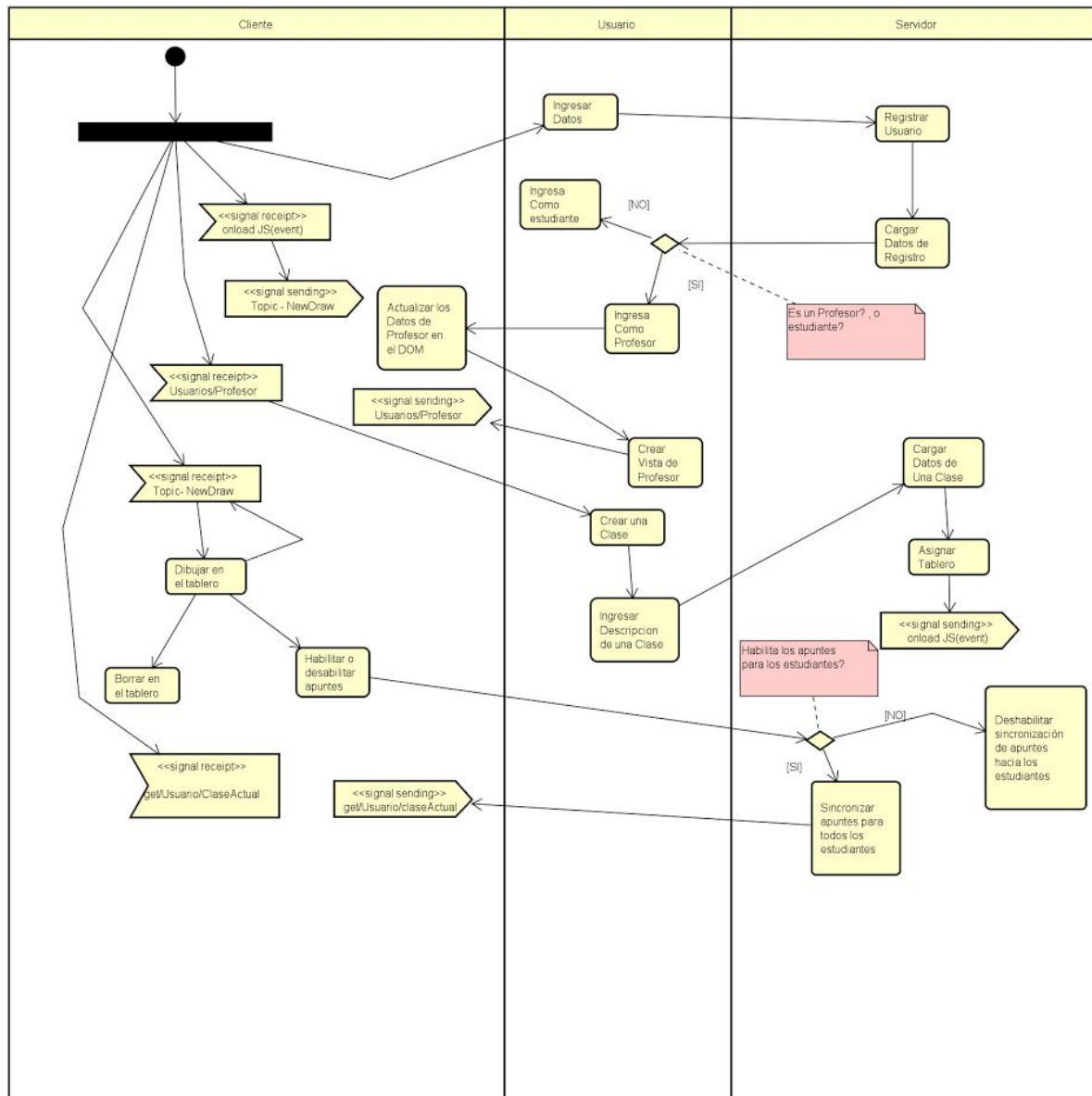
## System Architecture Description of SyncBoard

Doc #

Version: 2.0

Page 11 / 15

### 4 Diagrama de actividades



System Architecture Description of SyncBoard		
Doc #	Version: 2.0	Page 12 / 15

#### 4.1 Descripción del Componente 1

**Servidor:** Nuestro servidor va a ser aquel que nos cargará los datos de un profesor o estudiante respectivamente, aparte de esto, por cada clase registrada nos creará un tablero para interactuar con esa clase , también tendrá en cuenta el protocolo STOMP para la comunicación usando métodos HTTP para cargar datos de profesores

#### 4.2 Descripción del Componente 2

**Cliente:** Es aquel que nos maneja los eventos de crear un nuevo trazo o dibujo como tal , asignando cada tablero que el servidor está manejando para que cualquier persona pueda dibujar y este se sincronice con los demás.

#### 4.3 Descripción del Componente 3

**Usuario:** Este Componente nos actualiza los datos de registro en el DOM , aparte que ya una vez identificado como profesor , creará los datos de una clase, con sus respectivos requerimientos , para esto pues el usuario anteriormente nombrado es un estudiante o un profesor , en el caso de un estudiante este será aquel que se suscriba a nuestro servidor mediante peticiones para poder interactuar con SyncBoard.

System Architecture Description of SyncBoard		
Doc #	Version: 2.0	Page 13 / 15

## 5 Justification of architecture ATRIBUTOS DE CALIDAD

### 5.1 Justificación de la Arquitectura ATRIBUTOS DE CALIDAD

#### 5.2 Network architecture capabilities:

Gracias a la arquitectura implementada en la solución , Nuestro aplicativo cuenta con los siguientes atributos de calidad:

- Disponibilidad : al desplegar el aplicativo en dos servidores gestionados por un balanceador de cargas , el servicio ofrecido podrá ser accedido por cualquier cliente siempre y cuando esté en ejecución la aplicación en los dos servidores , lo cual se debe garantizar que como primer y primordial escenario el servicio esté funcionando correctamente con sus broker y su caché REDIS activado , y también se debe ejecutar antes las correctas pruebas pertinentes para que el servicio esté funcionando correctamente
- Modificabilidad: Al utilizar la arquitectura REST la implementación del software da la posibilidad del desarrollo de nuevos recursos que puedan ser accedidos muy fácilmente por medio del protocolo HTTP al realizar peticiones GET, POST, PUT y DELETE. Además que la estructura que se utilizó para el empaquetamiento de las diferentes clases por funcionalidades, da facilidad al desarrollador en la interpretación del código y agregar nuevas funcionalidades.
- Escalabilidad: Esta aplicación es escalable horizontalmente gracias al nodo del balanceo de cargas agregado que permite gestionar las peticiones realizadas por los usuarios en dos diferentes servidores, esto para no sobrecargar a un servidor si llega a haber un gran volumen de usuarios interactuando con la aplicación, que este puede ser un escenario considerable al momento de tener demasiados usuarios conectados a nuestro aplicativo
- Seguridad: la aplicación cuenta con un nivel alto en vulnerabilidades de seguridad puesto que no se ha manejado ningún tipo de seguridad como lo puede ser Shiro Login bean lo cual se debe considerar a un futuro a implementar para la información que se maneja de materias , profesores y estudiantes, Para mitigar el impacto que tiene la falta de seguridad en la aplicación, lo mejor es implementar software seguro haciendo uso de software ya construido como el de OWASP, el cual nos va a garantizar que la aplicación o será vulnerable a ciertos ataques por partes de los crackers.

#### 5.3 Risk analysis outputs Quality attributes

**IMPACTO:**

**1 → BAJO**

**3 → MEDIO**

**5 → ALTO**

- **1. Vulnerabilidad :** Fallo en la tercerización del Servicio  
**Impacto :** 4  
**Descripción :** Caída en el servicio AWS

## System Architecture Description of SyncBoard

Doc #

Version: 2.0

Page 14 / 15

- **2.Vulnerabilidad:** Baja seguridad en la plataforma donde se ejecuta la aplicación , puertos inseguros , información no cifrada  
**Impacto:** 5  
**Descripción:** Carencia de un sistema de seguridad en la aplicación
- **3. Vulnerabilidad :** Combinación de las dos vulnerabilidades encontradas anteriormente, incluyendo los fallos en la plataforma y servicio AWS  
**Impacto:** 1  
**Descripción:** Servicio de Caché Fuera de disponibilidad es decir de servicio.

## 6 Requirements traceability

Requirement	Component	Comment
Crear Tablero	Lógico	Es una clase que se encuentra en la parte lógica con sus atributos y sus métodos la cual tiene una representación gráfica html.
Registrar Usuario	Lógico	Es una clase con los atributos de usuario e implementa otras dos clases Profesor y estudiante que se diferencian por sus privilegios.
Modificación Colaborativa	JavaScript	Para que esté esta funcionalidad se usó STOMP y Websockets para lograr una comunicación bidireccional entre el cliente y el servidor
Despliegue de Redis	Servidor Redis	Una plataforma Ubuntu en AWS con el servicio del servidor redis instalado.
Despliegue NGINX	Servidor NGINX	Una plataforma Ubuntu en AWS con el servicio del servidor NGINX instalado.
Despliegue de Broker	Servidor "ActiveMQ"	Una plataforma Ubuntu en AWS con el servicio del servidor activeMQ instalado.

<b>System Architecture Description of SyncBoard</b>		
<b>Doc #</b>	<b>Version: 2.0</b>	<b>Page 15 / 15</b>