**UNIVERSIDAD MAYOR REAL Y PONTIFICIA SAN FRANCISCO XAVIER DE CHUQUISACA**

**FACULTAD DE CIENCIAS Y TECNOLOGÍA**

**INGENIERIA DE SISTEMAS**



**GRUPO**

**MODELO DE ANÁLISIS (FRONTEND CONTROL COLAS JAVASCRIPT)**

**Universitarios:**

Escobar Zarate Julio Cesar  
 Paredes Paredes Naomy Yailin  
 Rojas Torrico Zahira

**Docente:** Ing. Ramiro Duran

**Materia:** SIS324 – Ingeniería de software

**Sucre – Bolivia**

Índice

[Introducción 1](#_Toc152206575)

[1.1 Propósito de la Aplicación 1](#_Toc152206576)

[1.2 Alcance 1](#_Toc152206577)

[1.3 Definiciones, siglas y abreviaturas 1](#_Toc152206578)

[1.4 Referencias 2](#_Toc152206579)

[1.5 Visión general 2](#_Toc152206580)

[Visión de Requerimientos 2](#_Toc152206581)

[2.1 Descripción General del Proyecto 2](#_Toc152206582)

[2.2 Objetivos del Sistema 2](#_Toc152206583)

[2.3 Beneficios Esperados 3](#_Toc152206584)

[2.4 Restricciones y Limitaciones 3](#_Toc152206585)

[2.5 Usuarios y Actores Clave 3](#_Toc152206586)

[2.6 Resumen de los Principales Requisitos Funcionales y No Funcionales 4](#_Toc152206587)

[2.6.1 Requisitos funcionales 4](#_Toc152206588)

[2.6.2 Requisitos no funcionales 4](#_Toc152206589)

[2.7 Modelo de Casos de Uso 4](#_Toc152206590)

[2.8 DESCRIPCIÓN DETALLADA DE LOS CASOS DE USO 9](#_Toc152206591)

[2.9 Arquitectura de casos de uso 13](#_Toc152206592)

[Glosario de Términos 15](#_Toc152206593)

**Modelo de Análisis (Frontend Control Colas JavaScript)**

**Introducción**

**1.1 Propósito de la Aplicación**

El propósito de esta aplicación es gestionar y controlar colas de usuarios en una interfaz frontend utilizando JavaScript. La aplicación busca optimizar la atención al cliente mediante un sistema eficiente de gestión de colas.

**1.2 Alcance**

El alcance de la aplicación incluye:

* Implementación de una interfaz de usuario para la gestión de colas.
* Funcionalidades de encolado y desencolado de usuarios.
* Visualización del estado de la cola en tiempo real.
* Integración con sistemas backend para la actualización de datos.

**1.3 Definiciones, siglas y abreviaturas**

* **Frontend**: Parte del desarrollo web que se ocupa de la interfaz de usuario.
* **Control de colas**: Sistema que gestiona el orden de atención de usuarios.
* **JavaScript (JS)**: Lenguaje de programación utilizado principalmente para el desarrollo web del lado del cliente.

**1.4 Referencias**

* Documentación oficial de JavaScript.
* Guías y buenas prácticas de desarrollo frontend.
* Frameworks y bibliotecas de JavaScript como React o Vue.js.

**1.5 Visión general**

Este documento presenta el modelo de análisis para una aplicación de control de colas en frontend usando JavaScript. Incluye la descripción de los requerimientos, casos de uso, y arquitectura del sistema.

**Visión de Requerimientos**

**2.1 Descripción General del Proyecto**

El proyecto tiene como objetivo desarrollar una aplicación web que permita la gestión de colas de usuarios de manera eficiente y en tiempo real. La aplicación estará desarrollada con tecnologías frontend modernas y se integrará con servicios backend para la persistencia de datos.

**2.2 Objetivos del Sistema**

* Facilitar la gestión de colas de usuarios.
* Mejorar la eficiencia en la atención al cliente.
* Proveer una interfaz intuitiva y fácil de usar.
* Garantizar la actualización en tiempo real del estado de la cola.

**2.3 Beneficios Esperados**

* Reducción de tiempos de espera.
* Mejora en la experiencia del usuario.
* Optimización de recursos en la atención al cliente.
* Mayor transparencia en el proceso de atención.

**2.4 Restricciones y Limitaciones**

* Dependencia de la conectividad a internet para la actualización en tiempo real.
* Limitaciones de rendimiento según la capacidad del servidor backend.
* Compatibilidad con diferentes navegadores web y dispositivos.

**2.5 Usuarios y Actores Clave**

* **Administradores**: Usuarios con permisos para gestionar y configurar la aplicación.
* **Clientes**: Usuarios que ingresan a la cola y esperan ser atendidos.
* **Operadores**: Personal que atiende a los clientes según el orden de la cola.

**2.6 Resumen de los Principales Requisitos Funcionales y No Funcionales**

**2.6.1 Requisitos funcionales**

* **Registro y autenticación de usuarios**: Permitir el acceso seguro a la aplicación.
* **Gestión de colas**: Funcionalidades para encolar y desencolar usuarios.
* **Visualización en tiempo real**: Mostrar el estado actual de la cola.
* **Notificaciones**: Informar a los usuarios sobre el estado de la cola y su posición.

**2.6.2 Requisitos no funcionales**

* **Rendimiento**: La aplicación debe ser rápida y eficiente.
* **Escalabilidad**: Capacidad para manejar un número creciente de usuarios.
* **Seguridad**: Protección de los datos de los usuarios y del sistema.
* **Usabilidad**: Interfaz intuitiva y fácil de usar.

**2.7 Modelo de Casos de Uso**

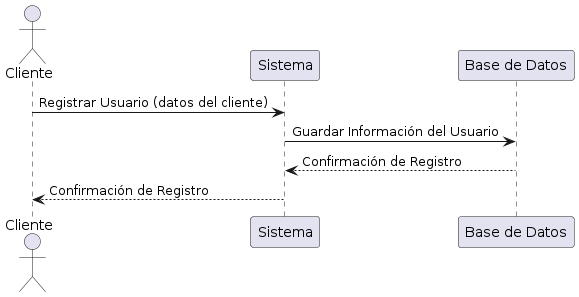
El modelo de casos de uso describe las interacciones entre los usuarios y el sistema, detallando cómo cada actor utiliza las funcionalidades de la aplicación.

**2.8 Descripción Detallada de los Casos de Uso**

1. **Registro de Usuario**:
   * **Actor**: Cliente
   * **Descripción**: El cliente se registra en la aplicación proporcionando sus datos básicos.
   * **Precondiciones**: El cliente debe tener acceso a internet.
   * **Postcondiciones**: El cliente queda registrado en el sistema.
2. **Inicio de Sesión**:
   * **Actor**: Cliente/Administrador
   * **Descripción**: El usuario ingresa sus credenciales para acceder a la aplicación.
   * **Precondiciones**: El usuario debe estar registrado.
   * **Postcondiciones**: El usuario accede a su cuenta.
3. **Encolar Usuario**:
   * **Actor**: Cliente
   * **Descripción**: El cliente se añade a la cola para ser atendido.
   * **Precondiciones**: El cliente debe estar autenticado.
   * **Postcondiciones**: El cliente aparece en la lista de espera.
4. **Desencolar Usuario**:
   * **Actor**: Operador
   * **Descripción**: El operador atiende a un cliente, removiéndolo de la cola.
   * **Precondiciones**: El cliente debe estar en la cola.
   * **Postcondiciones**: El cliente es removido de la lista de espera.
5. **Visualización del Estado de la Cola**:
   * **Actor**: Cliente/Operador
   * **Descripción**: El usuario visualiza el estado actual de la cola.
   * **Precondiciones**: El usuario debe estar autenticado.
   * **Postcondiciones**: El usuario ve el estado actualizado de la cola.
6. **Configuración del Sistema**:
   * **Actor**: Administrador
   * **Descripción**: El administrador configura los parámetros de la aplicación.
   * **Precondiciones**: El administrador debe estar autenticado.
   * **Postcondiciones**: Los cambios se guardan en el sistema.

**DIAGRAMA UML DE LOS CASO DE USO**

1. **CU01 - Registro de Usuario:**

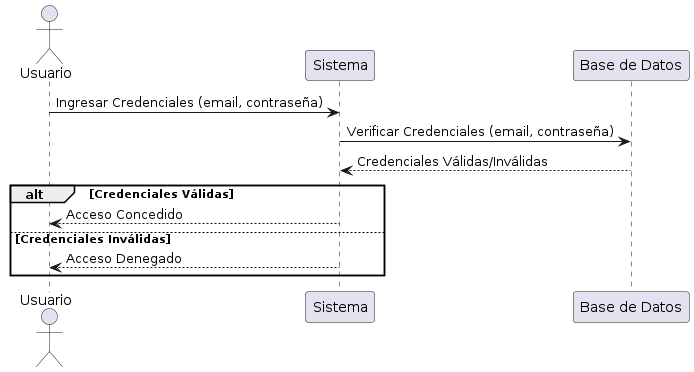


**Descripción del Diagrama de Secuencia**

1. **Cliente**: Es el actor que inicia el proceso de registro proporcionando sus datos básicos.
2. **Sistema**: Es la aplicación que recibe los datos del cliente.
3. **Base de Datos**: Es donde se almacenan los datos del cliente.

**Pasos del Diagrama**

1. El cliente envía una solicitud de registro con sus datos básicos al sistema.
2. El sistema recibe los datos y envía una solicitud para guardar esta información en la base de datos.
3. La base de datos confirma que los datos han sido guardados correctamente y envía una confirmación de registro al sistema.
4. El sistema, a su vez, envía una confirmación de registro al cliente.
5. **CU02 - Registro de Usuario:**

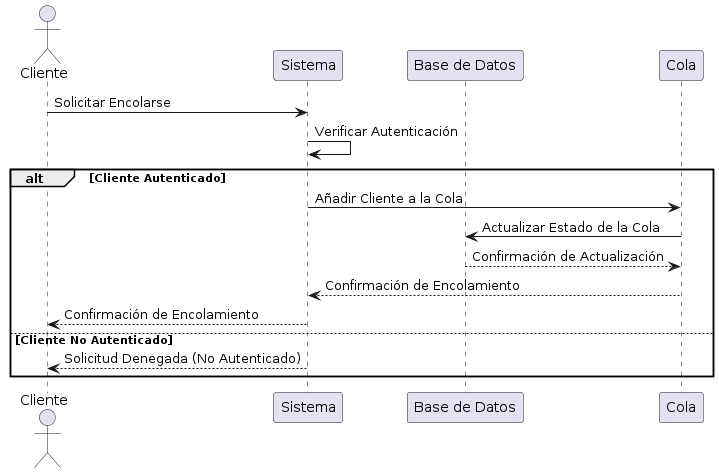


### Descripción del Diagrama de Secuencia

1. **Usuario**: Es el actor que puede ser un cliente o un administrador que intenta iniciar sesión en la aplicación.
2. **Sistema**: Es la aplicación que recibe y verifica las credenciales del usuario.
3. **Base de Datos**: Es donde se almacenan las credenciales del usuario.

### Pasos del Diagrama

1. El usuario envía sus credenciales (email y contraseña) al sistema.
2. El sistema recibe las credenciales y envía una solicitud para verificar estas credenciales en la base de datos.
3. La base de datos verifica las credenciales y devuelve una respuesta al sistema indicando si las credenciales son válidas o no.
4. El sistema evalúa la respuesta de la base de datos:
   * Si las credenciales son válidas, el sistema concede el acceso al usuario y envía una confirmación de acceso concedido.
   * Si las credenciales son inválidas, el sistema deniega el acceso al usuario y envía una notificación de acceso denegado.
5. **CU03- Encolar Usuario:**

****

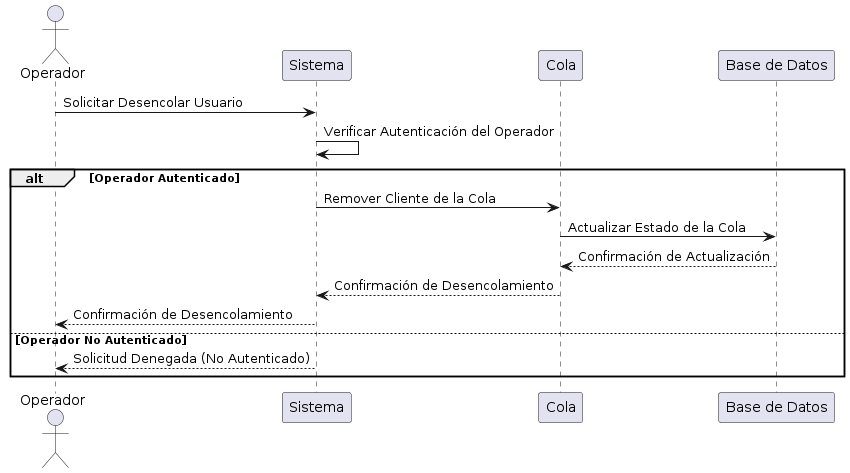
### Descripción del Diagrama de Secuencia

1. **Cliente**: Es el actor que quiere añadirse a la cola para ser atendido.
2. **Sistema**: Es la aplicación que gestiona el encolamiento de los clientes.
3. **Base de Datos**: Es donde se almacena el estado de la cola.
4. **Cola**: Es la estructura que gestiona la lista de espera de los clientes.

### Pasos del Diagrama

1. El cliente envía una solicitud de encolarse al sistema.
2. El sistema verifica si el cliente está autenticado.
3. Si el cliente está autenticado:
   1. El sistema envía una solicitud para añadir al cliente a la cola.
   2. La cola actualiza su estado en la base de datos.
   3. La base de datos confirma que el estado de la cola ha sido actualizado.
   4. La cola confirma al sistema que el cliente ha sido encolado.
   5. El sistema envía una confirmación de encolamiento al cliente.
4. Si el cliente no está autenticado:
   1. El sistema envía una respuesta de solicitud denegada al cliente indicando que no está autenticado.

**4.CU04- Desencolar Usuario**:



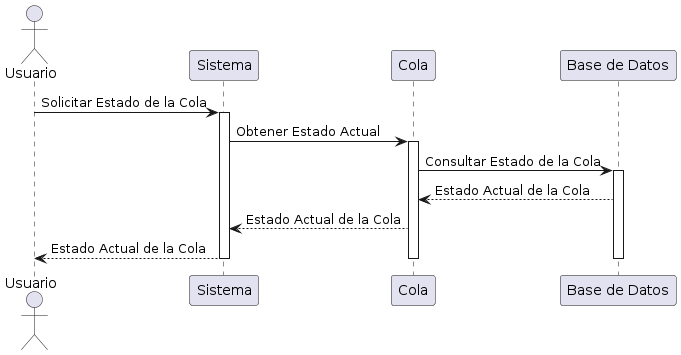
**Descripción del Diagrama de Secuencia**

1. **Operador**: Es el actor que desea atender a un cliente y removerlo de la cola.
2. **Sistema**: Es la aplicación que gestiona el proceso de desencolamiento.
3. **Cola**: Es la estructura que contiene la lista de espera de los clientes.
4. **Base de Datos**: Es donde se almacena el estado de la cola.

**Pasos del Diagrama**

1. El operador envía una solicitud para desencolar a un cliente al sistema.
2. El sistema verifica si el operador está autenticado.
3. Si el operador está autenticado:
   1. El sistema envía una solicitud para remover al cliente de la cola.
   2. La cola actualiza su estado en la base de datos.
   3. La base de datos confirma la actualización.
   4. La cola confirma al sistema que el cliente ha sido desencolado.
   5. El sistema confirma al operador que el cliente ha sido desencolado.
4. Si el operador no está autenticado:
   1. El sistema envía una respuesta de solicitud denegada al operador indicando que no está autenticado.

**5.CU05- Visualización del Estado de la Cola**



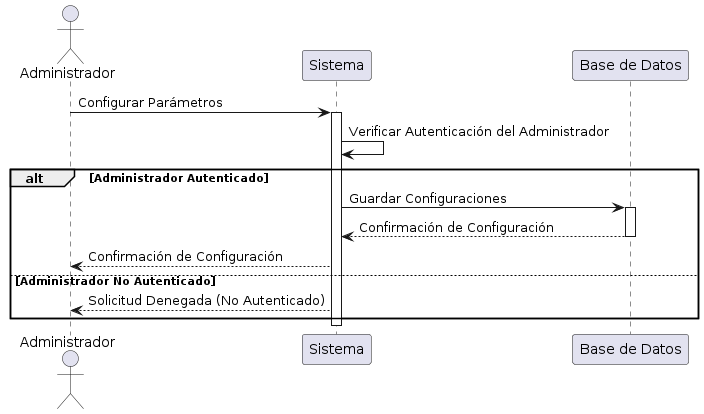
### Descripción del Diagrama de Secuencia

1. **Cliente/Operador**: Es el actor que desea visualizar el estado actual de la cola.
2. **Sistema**: Es la aplicación que gestiona la visualización del estado de la cola.
3. **Cola**: Es la estructura que contiene la lista de espera de los clientes.
4. **Base de Datos**: Es donde se almacena el estado actual de la cola.

### Pasos del Diagrama

1. El cliente u operador envía una solicitud para visualizar el estado de la cola al sistema.
2. El sistema verifica si el usuario está autenticado.
3. Si el usuario está autenticado:
   1. El sistema envía una solicitud para obtener el estado actual de la cola.
   2. La cola consulta su estado actual en la base de datos.
   3. La base de datos devuelve el estado actual de la cola.
   4. La cola confirma al sistema el estado actual.
   5. El sistema envía el estado actual de la cola al cliente u operador.
4. Si el usuario no está autenticado:
   1. El sistema envía una respuesta de solicitud denegada al usuario indicando que no está autenticado.

**6.CU06- Configuración del Sistema:**



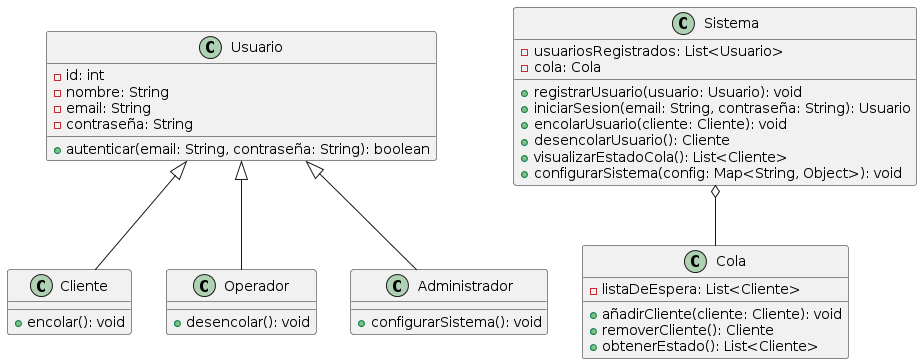
### Descripción del Diagrama de Secuencia

1. **Administrador**: Es el actor que desea configurar los parámetros del sistema.
2. **Sistema**: Es la aplicación que gestiona la configuración de los parámetros.
3. **Base de Datos**: Es donde se guardan los cambios en la configuración del sistema.

### Pasos del Diagrama

1. El administrador envía una solicitud para configurar los parámetros del sistema al sistema.
2. El sistema verifica si el administrador está autenticado.
3. Si el administrador está autenticado:
   1. El sistema guarda las configuraciones en la base de datos.
   2. La base de datos confirma que las configuraciones han sido guardadas correctamente.
   3. El sistema envía una confirmación de configuración al administrador.
4. Si el administrador no está autenticado:
   1. El sistema envía una respuesta de solicitud denegada al administrador indicando que no está autenticado.

DIAGRAMA DE CLASES:



### Explicación del Diagrama de Clases

#### Clase Usuario

* Representa un usuario genérico del sistema.
* Contiene atributos como id, nombre, email y contraseña para identificar y autenticar a los usuarios.

#### Clase Cliente

* Hereda de la clase Usuario.
* Tiene un método encolar() que permite a un cliente añadirse a la cola para ser atendido.

#### Clase Operador

* También hereda de la clase Usuario.
* Contiene un método desencolar() que permite a un operador atender a un cliente y removerlo de la cola.

#### Clase Administrador

* Hereda de la clase Usuario.
* Tiene un método configurarSistema() que permite al administrador ajustar los parámetros de la aplicación.

#### Clase Cola

* Gestiona la lista de espera de clientes.
* Tiene métodos para añadir un cliente a la cola (añadirCliente()), remover un cliente de la cola (removerCliente()) y obtener el estado actual de la cola (obtenerEstado()).

#### Clase Sistema

* Es la clase principal que coordina todas las operaciones del sistema.
* Contiene métodos para registrar usuarios, iniciar sesión, encolar y desencolar usuarios, visualizar el estado de la cola y configurar el sistema.

**2.9 Arquitectura de casos de uso**

La arquitectura de casos de uso proporciona una visión estructurada de cómo se implementarán y se relacionarán los casos de uso dentro del sistema. Esta arquitectura incluye diagramas UML para representar visualmente las interacciones y flujos entre los diferentes actores y casos de uso.

**Glosario de Términos**

* **API**: Interfaz de Programación de Aplicaciones, un conjunto de definiciones y protocolos para construir e integrar software.
* **Frontend**: La parte del desarrollo web que interactúa directamente con los usuarios.
* **Backend**: La parte del desarrollo web que se encarga del procesamiento de datos y la lógica del negocio en el servidor.
* **Encolar**: Añadir un usuario a la lista de espera.
* **Desencolar**: Remover un usuario de la lista de espera una vez ha sido atendido.