

Reporte Técnico del Software

2023

I - Student

Sistema Automatizado para el Control de Asistencia en
Escuelas, mediante el uso de Escáneres Auxiliares.

Líder de proyecto

Irving Rafael Conde Marín

Integrantes del equipo 5



Líder de proyecto

Irving Rafael Conde Marín



Aaron Emiliano Torres Núñez



Adriel Eduardo Peregrina Soto



Magdyel Aguilar Cid



Fermín Del Rosario Antonio



Aldo Torres Ramírez

Universidad Veracruzana

Facultad:

USBI Ixtaczoquitlán

Licenciatura:

Ingeniería de software

Experiencia Educativa:

Desarrollo de Software

NRC

91059

Grupo:

601

Docente:

Jorge Ernesto Gonzales
Díaz

Contenido

Tema 1.- PLANTEAMIENTO DEL PROBLEMA	4
1.1.- Identificación de la problemática.....	4
1.2.- Descripción del cliente objetivo.....	4
1.3.- Breve propuesta de solución (Conceptual y tecnológica).....	4
Tema 2.- PROCESO DE INGENIERÍA DE LA SOLUCIÓN.....	6
2.1 Alcance de la solución.....	6
2.2 Clases de usuarios y características.....	7
2.3 Características del producto	8
2.4 Ambiente operativo	10
2.5 Requerimientos	11
2.5.1 Aplicación Web para Docentes	11
2.5.2 Aplicación Web para Alumnos	21
2.5.3 API para interfaz web de docentes	26
2.5.4 API para interfaz web de alumnos	33
2.5.5 Sistema para lectura de códigos QR.....	38
2.6 Análisis y diseño	51
2.6.2 Especificación de casos de uso.....	54
2.6.3 Mockups.....	73
2.6.4 Arquitectura de software de la solución	91
2.6.5 Modelo de datos	100
2.7 Implementación.....	108
2.7.1 Descripción de metodología/ ciclo de vida empleado para la implementación de la solución	108
2.7.2 Evidencias de trabajo colaborativo en repositorio de código fuente	114
2.7.3 Capturas de pantalla con la descripción de las partes del código más relevante.....	121
2.8 Repositorio de la solución	135
2.8.1 URL.....	135
2.9 Administración del proyecto	135
2.9.1 Roles que desempeño cada miembro del equipo.....	135
2.9.2 Metodología empleada para el desarrollo de la solución	136
2.9.3 Evidencias de la gestión del proyecto	148
2.10 Conclusiones	161

Tema 1.- PLANTEAMIENTO DEL PROBLEMA

1.1.- Identificación de la problemática

La problemática identificada fue planteada por los docentes de la facultad de negocios y tecnologías, esta problemática es con respecto a los inconvenientes presentados en el pase de lista tradicional, como lo es el seguimiento y el control de las asistencias de los alumnos, ya que se les ha convertido en un problema de la vida cotidiana la cual requiere de una solución eficiente.

Esto debido a que la toma de asistencia toma un tiempo y esfuerzo del docente que reduce el tiempo de la clase, y puede haber cierta confusión al llevar un control sobre quiénes están realmente presentes en el salón, por lo cual se ha solicitado a nuestro equipo de trabajo buscar una solución y presentarla ante las personas correspondientes para aprobar dicha solución planteada.

1.2.- Descripción del cliente objetivo

La propuesta ideada por el equipo de desarrollo tiene como cliente objetivo a los usuarios Docentes y a los usuarios Alumnos.

Se ayudará a los usuarios Docentes a llevar un mejor control eficiente sobre sus materias correspondientes, así como también llevar seguimiento de la toma de asistencias de los alumnos en sus materias asignadas.

A los usuarios Alumnos se ayudará a consultar su información registrada en el sistema, así como la generación de códigos QR para llevar a cabo su toma de asistencia.

1.3.- Breve propuesta de solución (Conceptual y tecnológica)

Para la solución presentada por los docentes de la facultad de negocios y tecnologías, el equipo de desarrollo ha ideado la propuesta de un sistema automatizado para la toma de asistencia que pueda satisfacer las necesidades conocidas de los docentes. Dicha propuesta tiene como nombre Sistema Automatizado para el Control de Asistencia en Escuelas, Mediante el uso de Escáneres Auxiliares, dicho nombre se abreviará con *SACAEMEA*

Propuesta Conceptual:

El SACAEMEA tiene como objetivo principal el agilizar el proceso de la toma de asistencia escolar de los alumnos mediante un software especializado para dicha necesidad que utilizará un dispositivo periférico (el cual será remplazado con el simulador de códigos QR), el cual será capaz de registrar la asistencia de los alumnos en su aula de clases correspondiente mediante el escaneo de un código QR generado por una aplicación móvil, este código contendrá los datos necesarios sobre el alumno (matrícula y nombre), así también como lo es la hora de registro en que el código fue leído, lo cual facilitará un seguimiento preciso de la asistencia. Esta solución permitirá a los docentes tener un control más preciso sobre la asistencia de los alumnos y realizar este proceso de una forma más sencilla y ágil, así como también podrá permitir eliminar los errores y las demoras asociadas al registro manual de asistencias, permitiendo a los docentes contar con información actualizada y confiable.

Propuesta Tecnológica:

El SACAEMEA contará con 3 interfaces de usuario web (*Front-end*) que serán desarrolladas utilizando el Framework Angular, 2 interfaces de programación de aplicaciones (APIs) que serán desarrolladas con el entorno de desarrollo NodeJS y la utilización de 2 nubes, las cuales serán Firebase y Google Cloud, la cual Firebase utiliza una base de datos no relacional, y Google Cloud utilizará una base de datos Postgres.

La primera interfaz de usuario será para los docentes, dicha interfaz contará con un Login y podrá ayudar a llevar un control y seguimiento de la toma de asistencia de los alumnos, así como también permitirá al usuario consultar sus materias correspondientes, y solicitar un alta o baja de una materia a los encargados del sistema, además, será capaz de subir una lista de asistencia si el sistema no cuenta con una, o en caso contrario, poder visualizar dicha lista de asistencia almacenada.

La segunda interfaz de usuario será para los alumnos, esta interfaz permitirá al alumno visualizar su información registrada por el sistema, así como sus materias asignadas por el sistema, además, al seleccionar una materia podrá generar un código QR, con el cual podrá realizar el proceso de toma de asistencia.

La tercera interfaz de usuario será un simulador de lectura de códigos QR, esta permitirá realizar el proceso de toma de asistencia, debido a que leerá los códigos QR y validará que los datos sean correctos y pertenezcan a las materias correspondientes, además, podrá ser configurada con el salón y el edificio deseado para realizar dicho proceso.

Las interfaces de programación de aplicaciones (APIs) ayudarán a las interfaces de usuarios a llevar a cabo su funcionamiento de forma correcta, así como también deberán ser capaces de conectarse a las nubes utilizadas o a las bases de datos correspondientes.

Las nubes utilizadas podrán almacenar la información necesaria para que el sistema lleve a cabo su funcionamiento de forma correcta y eficiente, así como lo son los datos de los alumnos, docentes, de las listas de asistencia, materias, etc.

Tema 2.- PROCESO DE INGENIERÍA DE LA SOLUCIÓN

2.1 Alcance de la solución

El alcance de la solución se centra en agilizar y automatizar el proceso de registro de asistencia escolar, ofreciendo a los docentes un control más preciso y facilitando la consulta de información por parte de los alumnos. La solución propuesta busca mejorar la eficiencia y eliminar errores asociados al proceso manual de toma de asistencia, brindando un proceso más organizado y eficiente.

Con respecto a la solución propuesta por el equipo de desarrollo: SACAEMEA (Sistema Automatizado para el Control de Asistencia en Escuelas Mediante el uso de Escáneres Auxiliares) tiene como alcance los puntos siguientes:

1. Desarrollo de software especializado: El proyecto tiene como alcance crear software compuesto por distintos módulos para llevar a cabo el proceso de toma de asistencia de los alumnos de una forma automatizada. Dicho software estará compuesto por distintas interfaces de usuario web que serán codificados en el framework Angular, así como la creación y utilización de APIs codificadas en NodeJS, además, el software deberá utilizar nubes para el almacenamiento de la información necesaria para su funcionamiento correcto y eficiente.
 - 1.1. Interfaz de usuario web para docentes: Se desarrollará una interfaz de usuario web la cual fue ideada para ayudar a los docentes a satisfacer sus necesidades. Esta interfaz web les permitirá a los docentes a llevar un control y seguimiento de la toma de asistencia de los alumnos de una forma más eficiente y sencilla. Además, dicha interfaz permitirá a los docentes consultar sus materias asignadas por el sistema, así como también se les permitirá solicitar altas o bajas de materias, subir listas de asistencia y visualizar listas de asistencia almacenadas.
 - 1.2. Interfaz de usuario web para alumnos: Se desarrollará una interfaz de usuario web para auxiliar a los alumnos en su proceso de toma de asistencia automatizada. Esta interfaz permitirá a los alumnos a consultar su información registrada en el sistema, así como también la información de sus materias asignadas. Además, al seleccionar una materia mostrada en la interfaz web se podrán generar códigos QR para realizar el proceso de toma de asistencia.
 - 1.3. API para docentes: Se desarrollará una API para el funcionamiento de la interfaz web de los docentes. En esta API se encontrarán las funciones necesarias para realizar los procesos necesarios o solicitados por el cliente, como lo son Registro de cuentas de usuario, inicio de sesión, consulta de información del docente, consulta de información de materias, entre otros funcionamientos.
 - 1.4. API para alumnos: Se desarrollará una API para el funcionamiento de la interfaz web de los docentes. En esta API se encontrarán las funciones necesarias para realizar los procesos necesarios o solicitados por el cliente, como lo son Registro de cuentas de

usuario, inicio de sesión, consulta de información de los alumnos, materias asignadas por el sistema.

2. Utilización de dispositivos periféricos: El software deberá funcionar con una pantalla que permitirá la configuración del equipo, como lo es el internet, edificio, salón, etc..; Además, dicho dispositivo deberá contar con una cámara para la lectura de códigos QR, pero debido a la falta de presupuesto del equipo de desarrollo se simularán los dispositivos con una interfaz web para la lectura de códigos QR, con los cuales será posible llevar a cabo el proceso de toma de asistencia automático utilizando su información contenida.
 - 2.1. Simulador de lectura de códigos QR: Debido a nuestra falta de presupuesto se desarrollará una interfaz que simulará la lectura de los códigos QR escaneados por los dispositivos periféricos. Este simulador validará los datos y asegurará que correspondan a las materias adecuadas, además de permitir configuraciones como el salón y el edificio para el proceso de toma de asistencia, además, deberá hacer uso de la cámara web del dispositivo utilizado para llevar a cabo la simulación lo más real posible.
3. Utilización de nubes y bases de datos: Se utilizarán las nubes Firebase y Google Cloud para almacenar la información necesaria, como datos de alumnos, docentes, listas de asistencia, y materias.
 - 3.1. La utilización del Storage de firebase le permitirá al sistema almacenar las imágenes de las cuentas creadas en los registros de cuentas, obteniendo además sus url para su posterior almacenado en la base de datos.
 - 3.2. La utilización del Firestore de firebase permitirá al sistema almacenar la información correspondiente a las listas de asistencia de las materias registradas por el sistema, así como la información de las materias con su respectiva hora, edificio y salón, los cuales serán utilizados por el simulador de escáner de códigos QR.
 - 3.3. La utilización de Google Cloud será para crear una instancia y alojar una base de datos relacional de tipo Postgres, allí se almacenarán los datos de los alumnos, docentes y cuentas creadas en el software.

2.2 Clases de usuarios y características

Para este producto resultante del proyecto se tendrán en consideración tres clases de usuarios que estarán presentes en las aplicaciones web a desarrollar, los cuales son: Usuario Estudiante, Usuario Docente. Estos usuarios desempeñarán distintas funciones en la utilización de las aplicaciones y podrán utilizar distintas herramientas en las aplicaciones disponibles para estos, a continuación, se describe de forma detallada lo anteriormente mencionado:

Usuario Estudiante:

Este tipo de usuario es el que estará usando uno de los softwares desarrollados en este proyecto, la cual será una aplicación web que podrá ser accedida por medio de los navegadores de sus smartphones.

Este tipo de usuarios podrán realizar por medio de la aplicación lo siguiente:

- Registrarse al sistema
- Iniciar Sesión en la aplicación
- Cerrar Sesión en la aplicación
- Generar códigos QR que servirá para el proceso de su toma de asistencia

Usuario Docente:

Este tipo de usuario es el que estará usando uno de los softwares desarrollados en este proyecto, la cual será una aplicación web que podrá ser accedida por medio de los navegadores de los dispositivos de cómputo encontrados en cada aula de la FNT, o por medio de sus dispositivos de cómputo personales.

Este tipo de usuarios podrán realizar por medio de la aplicación lo siguiente:

- Registrarse en el sistema
- Iniciar sesión en la aplicación
- Cerrar sesión en la aplicación
- Modificar las listas de asistencia de los alumnos
- Consultar las listas de asistencia de los alumnos
- Consultar sus materias correspondientes
- Solicitar alta de materias
- Solicitar baja de materias

2.3 Características del producto

Aplicación Web para docentes

1. Contará con un apartado de registro
2. Contará con un apartado de login
3. Deberá contar con un menú para visualizar sus materias correspondientes
4. Deberá contar con un apartado para solicitar una baja de materia
5. Deberá contar con un apartado para solicitar un alta de materia
6. Deberá contar con una opción para subir su formato de asistencia en formato .xls
7. Deberá contar con un apartado para visualizar la lista de las asistencias de los alumnos
8. Contará con un apartado de cierre de sesión

Aplicación Web para Alumnos

1. Contará con un apartado de registro
2. Contará con un apartado de login
3. Tendrá un apartado para mostrar y verificar que los datos del alumno son totalmente correctos
4. Deberá contar con un apartado para generar un código QR, el cual será necesario para el proceso de toma de asistencia del alumno
5. Contará con un apartado de cierre de sesión

Sensor para la toma de asistencia (simulado por aplicación Web)

1. Deberá tener un botón sobre el encendido del equipo
2. Deberá tener un botón sobre el apagado del equipo
3. El simulador deberá tener un apartado para la configuración del equipo
4. Podrá realizar la lectura de códigos QR
5. Deberá mostrar la información contenida en los códigos QR leídos
6. Guardará temporalmente el historial de los códigos QR leídos
7. Podrá realizar el proceso del registro de asistencia del alumno

API para docentes

1. Deberá contar con un apartado para la configuración de conexión a la bd
2. Deberá ser capaz de realizar el proceso de creación de cuentas de Docentes
3. Deberá ser capaz de realizar el proceso de inicio de sesión de los Docentes
4. Deberá ser capaz de realizar el proceso de alta de materia
5. Deberá ser capaz de realizar el proceso de baja de materia
6. Será capaz de consultar la información personal del docente
7. Podrá consultar las materias correspondientes al docente

API para alumnos

1. Deberá contar con un apartado para la configuración de conexión a la bd
2. Podrá consultar las materias correspondientes al alumno
3. Deberá ser capaz de realizar el proceso de inicio de sesión de los Deberá ser capaz de realizar el proceso de consulta de alumnos
4. Deberá ser capaz de realizar el proceso de consulta de alumnos
5. Deberá ser capaz de realizar el proceso de creación de cuentas de estudiantes

2.4 Ambiente operativo

Las aplicaciones web planteadas previamente se desarrollarán para funcionar en cualquier navegador web, permitiendo que sean de fácil acceso en cualquier dispositivo.

Front-End: Angular

Para nuestra parte del Front-End de las aplicaciones web se utilizará el framework conocido como Angular, debido a que nuestros desarrolladores están familiarizados con este y su funcionamiento en base a componentes permite un mejor diseño de su arquitectura de funcionamiento, permitiendo que sea más adaptable a cambios y a la facilidad de mantenimiento.

Back-End: NodeJS

La parte del Back-End se llevará a cabo utilizando la tecnología de Nodejs, ya que el encargado de esta parte está capacitado totalmente para llevar el desarrollo utilizando esta tecnología.

Base de datos: PostgreSQL

Esta base de datos nos permite el almacenado de los datos acerca de las cuentas creadas para la utilización de los usuarios, tanto para los usuarios estudiantes, usuarios docentes como para los usuarios administradores, además, aquí estará encontrada toda la información necesaria para el correcto funcionamiento de nuestro sistema. Otro punto para tomar sobre esto es que se trata de una base de datos de código abierto, es multiplataforma, es segura y permite el alto volumen de datos, por lo que ofrecerá las ventajas de no tener costo alguno y permitirá el manejo de grandes cantidades de datos de forma segura y poder ser administrada desde cualquier sistema operativo.

2.5 Requerimientos

2.5.1 Aplicación Web para Docentes

Requerimientos Funcionales del Software

I - STUDENT Aplicación Web para Docentes		
ID: Requerimiento Funcional 001	Nombre del Requerimiento: Registro de docente	
Tipo (Deseable / Necesario): Necesario	Categoría: Front - End	Prioridad de desarrollo: Alta
Entrada: <ol style="list-style-type: none">Nombre del docente <i>Aceptará una cadena de texto de 30 caracteres, y aceptará entrada de caracteres especiales.</i>Número de personal <i>Aceptará una cadena de 10 números, sin caracteres especiales</i>Correo Electrónico <i>Aceptará una cadena de texto de 30 caracteres, y aceptará entrada de caracteres especiales</i>Contraseña <i>Aceptará una cadena de texto de 30 caracteres, y aceptará entrada de caracteres especiales</i>Foto <i>Aceptará un archivo de tipo .jpg con un peso máximo de 2mb</i>		Salida: El proceso de registro de la cuenta del docente se realizará de forma correcta
Descripción: Precondición: El docente debe contar con su número de personal proporcionada por la facultad para completar su proceso de registro. Descripción: El sistema validará que las credenciales ingresadas son las solicitadas, verificará que cumplen con las restricciones dadas y que el usuario rellene todos los campos presentados. Postcondición: El sistema realizará correctamente el proceso de registro de la cuenta del docente		
Manejo de situaciones anormales: <ol style="list-style-type: none">El usuario ya se ha registrado previamente (Se mostrará un mensaje en pantalla de que la cuenta ya existe)Los campos no han sido llenados completamente (No permitirá realizar el proceso)Error al mandar los datos por mala conexión (Mostrar un mensaje en pantalla)		
Criterios de aceptación: <ul style="list-style-type: none">Las credenciales ingresadas al sistema al momento de realizar el registro al sistema son válidas en cuestión de las restricciones, permitiendo realizar el registro exitoso del usuario		

I - STUDENT Aplicación Web para Docentes				
ID: Requerimiento Funcional 002	Nombre del Requerimiento: Login de docente			
Tipo (Deseable / Necesario): Necesario	Categoría: Front - End	Prioridad de desarrollo: Alta		
Entrada: 1. Correo Electrónico <i>Aceptará una cadena de texto de 30 caracteres, aceptará la entrada de caracteres especiales.</i> 2. Contraseña <i>Aceptará una cadena de texto de 30 caracteres, aceptará la entrada de caracteres especiales.</i>	Salida: Inicio de sesión de forma correcta en la aplicación			
Descripción:				
Precondición: El docente debe estar previamente registrado en el sistema Descripción: El sistema validará que las credenciales ingresadas son correctas Postcondición: El sistema permitirá el acceso al sistema en caso de ser válidas				
Manejo de situaciones anormales: <ol style="list-style-type: none"> El docente no se ha registrado previamente al sistema (Se mostrará en pantalla un mensaje que informará al usuario que no está registrado en el sistema) Los campos no han sido llenados completamente (No permitirá realizar el proceso) 				
Criterios de aceptación: <ul style="list-style-type: none"> Las credenciales ingresadas al sistema al momento de iniciar sesión son correctas, por lo que el usuario podrá acceder al resto del sistema 				

I - STUDENT Aplicación Web para Docentes		
ID: Requerimiento Funcional 003	Nombre del Requerimiento: Materias del docente	
Tipo (Deseable / Necesario): Necesario	Categoría: Front - End	Prioridad de desarrollo: Alta
Entrada: 1. Validación de login del docente <i>Se tendrá que validar que la cuenta del docente exista</i>		Salida: Se mostrará un menú en la aplicación web donde se mostrará una lista de botones que representen las materias pertenecientes al docente. Los botones deberán contener: <ul style="list-style-type: none"> • Nombre de la experiencia educativa • Grupo Impartido • NRC de la materia • Turno • Licenciatura • Cantidad Alumnos
Descripción:		
Precondición: El docente deberá iniciar sesión en el sistema de forma correcta Descripción: El sistema mostrará un menú con las experiencias educativas pertenecientes a cada docente Postcondición: El sistema permitirá seleccionar alguna opción presentada, con la que podrá acceder a las listas de asistencia de los alumnos pertenecientes a dicha experiencia educativa		
Manejo de situaciones anormales: <ol style="list-style-type: none"> 1. El docente no cuenta con experiencias educativas para impartir (Se mostrará un mensaje en pantalla de que no cuenta con experiencias educativas) 2. La experiencia educativa no cuenta con una lista de asistencia oficial (Se mostrará un mensaje en pantalla de que no se ha completado el proceso de registro por parte de los estudiantes) 3. Una materia del docente no se encuentra en el sistema (El docente podrá solicitar el alta de esta materia) 4. Una materia que no pertenece al docente se muestra en el sistema (El docente podrá solicitar la baja de esta materia) 		
Criterios de aceptación: <ul style="list-style-type: none"> • Todas las experiencias educativas pertenecientes al docente se muestran de una forma correcta, y los botones funcionan sin problema alguno 		

I - STUDENT Aplicación Web para Docentes		
ID: Requerimiento Funcional 004	Nombre del Requerimiento: Alta Materia	
Tipo (Deseable / Necesario): Necesario	Categoría: Front - End	Prioridad de desarrollo: Alta
Entrada: 1. NRC de la materia <i>Se aceptará únicamente una cadena de cinco números enteros, sin ningún signo o carácter especial.</i>		Salida: Se mostrará un mensaje de que la petición del alta de materias se ha realizado con éxito. Si se aprueba la petición la materia aparecerá tiempo después en el apartado de materias Si se rechaza la petición, se mostrará un mensaje en pantalla sobre el rechazo
Descripción:		
<p>Precondición: El Docente deberá contar con el NRC de la materia deseada y su número de personal</p> <p>Descripción: El Docente solicitará la asignación de una materia a su disposición, si el encargado aprueba la solicitud se le asignará y podrá realizar los procesos estipulados en esta. Si se rechaza deberá mostrarse un mensaje en la pantalla cuando este inicie sesión posteriormente, informando que su solicitud fue rechazada.</p> <p>Postcondición: Ninguna</p>		
<p>Manejo de situaciones anormales:</p> <ol style="list-style-type: none"> 1. El NRC ingresado no coincide con ninguna materia registrada en el sistema (Se mostrará un mensaje por pantalla) 2. El horario de la materia solicitada coincide con el horario de otra materia asociada al Docente, por lo que se rechazará automáticamente la solicitud (Se muestra mensaje) 3. La materia solicitada ya está asociada a un docente, por lo que se rechazará automáticamente la solicitud (Se mostrará mensaje por pantalla) 		
Criterios de aceptación:		
<ul style="list-style-type: none"> • El proceso se realiza con éxito 		

I - STUDENT Aplicación Web para Docentes		
ID: Requerimiento Funcional 005	Nombre del Requerimiento: Baja Materia	
Tipo (Deseable / Necesario): Necesario	Categoría: Front - End	Prioridad de desarrollo: Alta
Entrada: 1. NRC de la materia <i>Se aceptará únicamente una cadena de cinco números enteros, sin ningún signo o carácter especial.</i>		Salida: Se mostrará un mensaje de que la petición de la baja de materia se ha realizado con éxito. Si se aprueba la petición la materia desaparecerá tiempo después en el apartado de materias Si se rechaza la petición, se mostrará un mensaje en pantalla sobre el rechazo
Descripción:		
<p>Precondición: El Docente deberá contar con el NRC de la materia deseada y su número de personal</p> <p>Descripción: El Docente solicitará la eliminación de una materia a su disposición, si el encargado aprueba la solicitud se le eliminará y dejará de aparecer en su apartado de materias. Si se rechaza deberá mostrarse un mensaje en la pantalla cuando este inicie sesión posteriormente, informando que su solicitud fue rechazada.</p> <p>Postcondición: Ninguna</p>		
<p>Manejo de situaciones anormales:</p> <ol style="list-style-type: none"> El NRC ingresado no coincide con ninguna materia registrada en el sistema (Se mostrará un mensaje por pantalla) El número de personal no coincide con ningún dato almacenado en el sistema (Se mostrará un mensaje por pantalla) <p>Criterios de aceptación:</p> <ul style="list-style-type: none"> El proceso se realiza con éxito 		

I - STUDENT Aplicación Web para Docentes		
ID: Requerimiento Funcional 006	Nombre del Requerimiento: Extraer información de archivo	
Tipo (Deseable / Necesario): Necesario	Categoría: Front - End	Prioridad de desarrollo: Alta
Entrada: 1. Archivo .XLS cargado al sistema <i>Se extraerá el formato del archivo de tipo Excel, obteniendo la información necesaria para realizar el proceso de toma de asistencia</i>		Salida: Clonación de la información del.XLS y almacenarlo a la base de datos.
Descripción:		
Precondición: El docente deberá cargar su archivo .XLS al sistema Descripción: El sistema clonará la información del archivo .XLS subido al sistema, almacenando estos datos en la base de datos para su posterior uso en el proceso de la toma de asistencia y su descarga por parte del usuario Postcondición: El docente deberá presionar el botón de guardar para finalizar el proceso		
Manejo de situaciones anormales: <ol style="list-style-type: none"> El archivo no se encuentra en formato .XLS (Se mostrará un mensaje por pantalla) El archivo no cumple con el formato predefinido por el cliente (Se mostrará un mensaje por pantalla) 		
Criterios de aceptación: <ul style="list-style-type: none"> El sistema clona y almacena correctamente los datos obtenidos por el archivo .XLS 		

I - STUDENT Aplicación Web para Docentes		
ID: Requerimiento Funcional 007	Nombre del Requerimiento: Tabla de asistencia	
Tipo (Deseable / Necesario): Necesario	Categoría: Front - End	Prioridad de desarrollo: Alta
Entrada: 2. Información del .XLS <i>Se extraerá la información almacenada en el sistema</i>		Salida: <p>Se mostrará una pantalla con un formato de tabla, donde se mostrará la información acerca de la lista de asistencia de la materia</p> <p>La información deberá contener:</p> <ul style="list-style-type: none"> • Nombre del curso • NRC • Matricula del Estudiante • Nombre del Estudiante • Estatus de inscripción del Estudiante • Carrera del Estudiante
Descripción:		
<p>Precondición: El docente debió subir su archivo .XLS que deseaba guardar en el sistema</p> <p>Descripción: El sistema mostrará una pestaña, en donde se encontrará una tabla con información desplegada, la cual será necesaria para que el docente consulte la información requerida sobre las asistencias de los alumnos</p> <p>Postcondición: Ninguna</p>		
<p>Manejo de situaciones anormales:</p> <p>1. El archivo almacenado en el sistema cuenta con el formato, pero no hay registros en el (Se mostrará un mensaje por pantalla)</p>		
<p>Criterios de aceptación:</p> <ul style="list-style-type: none"> • Toda la información acerca de los alumnos es correcta, así como el correcto pase de lista de la asistencia. 		

I - STUDENT Aplicación Web para Docentes		
ID: Requerimiento Funcional 008	Nombre del Requerimiento: Cierre de sesión	
Tipo (Deseable / Necesario): Necesario	Categoría: Front - End	Prioridad de desarrollo: Alta
Requisito 1 Entrada: <i>Botón de cerrar sesión</i>		Salida: La sesión del profesor se ha cerrado correctamente
Descripción: <p>Precondición: El docente debe haber iniciado sesión en el sistema Descripción: El docente podrá cerrar su sesión actual en el sistema Postcondición: El sistema deberá regresar al apartado de Login</p>		
Manejo de situaciones anormales:		
Criterios de aceptación: <ul style="list-style-type: none"> • La sesión del docente se ha cerrado correctamente 		

Requerimientos No Funcionales del Software

I - STUDENT Aplicación Web para Docentes		
ID: Requerimiento No Funcional 001	Nombre del Requerimiento: Diseño del registro de docente	
Tipo (Deseable / Necesario): Deseado	Requerimiento que lo utiliza Requerimiento Funcional 001	Prioridad: Media
Descripción:		
<p>Requisito 1: Tener un recuadro para ingresar los datos solicitados</p> <p>Requisito 2: Mostrar como título “Registrar Cuenta”</p> <p>Requisito 3: Contener iconos relacionados a los inputs mostrados</p> <p>Requisito 4: Contener un botón para registrar la cuenta</p> <p>Requisito 5: Contener un botón para cambiar la pestaña a Login del docente</p> <p>Requisito 6: Los caracteres ingresados en el input de contraseña deben ser intercambiados por asteriscos para evitar su visualización</p>		
Criterios de aceptación: <ul style="list-style-type: none"> Los diseños estipulados se han cumplido correctamente en el diseño del sistema 		

I - STUDENT Aplicación Web para Docentes		
ID: Requerimiento No Funcional 002	Nombre del Requerimiento: Diseño del login de docente	
Tipo (Deseable / Necesario): Deseado	Requerimiento que lo utiliza Requerimiento Funcional 002	Prioridad: Media
Descripción:		
<p>Requisito 1: Tener un logo de la universidad veracruzana</p> <p>Requisito 2: Tener un recuadro para ingresar los datos solicitados</p> <p>Requisito 3: Mostrar como título “Login Cuenta”</p> <p>Requisito 4: Contener iconos relacionados a los inputs mostrados</p> <p>Requisito 5: Contener un botón para Iniciar Sesión</p> <p>Requisito 6: Contener un botón para cambiar pestaña a Registro</p> <p>Requisito 7: Los caracteres ingresados en el input de contraseña deben ser intercambiados por asteriscos para evitar su visualización</p>		
Criterios de aceptación: <ul style="list-style-type: none"> Los diseños estipulados se han cumplido correctamente en el diseño del sistema 		

I - STUDENT Aplicación Web para Docentes		
ID: Requerimiento No Funcional 003	Nombre del Requerimiento: Diseño de apartado de materias	
Tipo (Deseable / Necesario): Deseado	Requerimiento que lo utiliza Requerimiento Funcional 003	Prioridad: Media
Descripción:		
<p>Requisito 1: El menú debe mostrar un listado vertical, con una serie de opciones en forma de botones</p> <p>Requisito 2: El contenido del listado debe ir centrado en la pantalla para una mejor visualización</p> <p>Requisito 3: La letra de los botones debe ser totalmente clara</p>		
Criterios de aceptación: <ul style="list-style-type: none"> Los diseños estipulados se han cumplido correctamente en el diseño del sistema 		

I - STUDENT Aplicación Web para Docentes		
ID: Requerimiento No Funcional 004	Nombre del Requerimiento: Diseño de dashboard de asistencia	
Tipo (Deseable / Necesario): Deseado	Requerimiento que lo utiliza Requerimiento Funcional 004	Prioridad: Media
Descripción:		
<p>Requisito 1: La pestaña debe mostrar una tabla donde actualizará únicamente la información solicitada para optimizar su desempeño</p> <p>Requisito 2: La información deberá estar separada por columnas</p>		
Criterios de aceptación: <ul style="list-style-type: none"> Los diseños estipulados se han cumplido correctamente en el diseño del sistema 		

2.5.2 Aplicación Web para Alumnos

Requerimientos Funcionales del Software

I - STUDENT Aplicación Web para Alumnos		
ID: Requerimiento Funcional 001	Nombre del Requerimiento: Registro de alumno	
Tipo (Deseable / Necesario): Necesario	Categoría: Front - End	Prioridad de desarrollo: Alta
Entrada: <ol style="list-style-type: none"> Nombre del alumno <i>Aceptará una cadena de texto de 30 caracteres, y aceptará entrada de caracteres especiales.</i> Apellidos del alumno <i>Aceptará una cadena de texto de 30 caracteres y aceptará entrada de caracteres especiales</i> Matrícula Escolar <i>Aceptará una cadena de texto de 10 caracteres, el formato de la matrícula será “zS” seguida de 8 números</i> Correo Electrónico <i>Aceptará una cadena de texto de 30 caracteres, y aceptará entrada de caracteres especiales</i> Contraseña <i>Aceptará una cadena de texto de 30 caracteres, y aceptará entrada de caracteres especiales</i> Foto <i>Aceptará un archivo de tipo .jpg con un peso máximo de 2mb</i> 	Salida: El proceso de registro de la cuenta del alumno se realizará de forma correcta	
Descripción:		
<p>Precondición: El alumno debe contar con su matrícula escolar proporcionada por la facultad para completar su proceso de registro.</p> <p>Descripción: El sistema validará que las credenciales ingresadas son las solicitadas, verificará que cumplen con las restricciones dadas y que el usuario rellene todos los campos presentados.</p> <p>Postcondición: El sistema realizará correctamente el proceso de registro de la cuenta del alumno</p>		
<p>Manejo de situaciones anormales:</p> <ol style="list-style-type: none"> El usuario ya se ha registrado previamente (Se mostrará un mensaje en pantalla de que la cuenta ya existe) La matrícula estudiantil del usuario ya ha sido registrada previamente (Se mostrará un mensaje en pantalla de que la cuenta ya existe) Los campos no han sido llenados completamente (No permitirá realizar el proceso) <p>Criterios de aceptación:</p> <ul style="list-style-type: none"> Las credenciales ingresadas al sistema al momento de realizar el registro al sistema son válidas en cuestión de las restricciones, permitiendo realizar el registro exitoso del usuario 		

I - STUDENT Aplicación Web para Alumnos				
ID: Requerimiento Funcional 002	Nombre del Requerimiento: Login de alumno			
Tipo (Deseable / Necesario): Necesario	Categoría: Front - End	Prioridad de desarrollo: Alta		
Entrada: 1. Correo Electrónico <i>Aceptará una cadena de texto de 30 caracteres, aceptará la entrada de caracteres especiales.</i> 2. Contraseña <i>Aceptará una cadena de texto de 30 caracteres, aceptará la entrada de caracteres especiales.</i>	Salida: Inicio de sesión de forma correcta en la aplicación			
Descripción:				
Precondición: El alumno debe estar previamente registrado en el sistema Descripción: El sistema validará que las credenciales ingresadas son correctas Postcondición: El sistema permitirá el acceso al sistema en caso de ser válidas				
Manejo de situaciones anormales: <ol style="list-style-type: none"> El alumno no se ha registrado previamente al sistema (Se mostrará en pantalla un mensaje que informará al usuario que no está registrado en el sistema) Los campos no han sido llenados completamente (No permitirá realizar el proceso) 				
Criterios de aceptación: <ul style="list-style-type: none"> Las credenciales ingresadas al sistema al momento de iniciar sesión son correctas, por lo que el usuario podrá acceder al resto del sistema 				

I - STUDENT Aplicación Web para Alumnos		
ID: Requerimiento Funcional 003	Nombre del Requerimiento: Menú de Inicio	
Tipo (Deseable / Necesario): Necesario	Categoría: Front - End	Prioridad de desarrollo: Alta
Entrada: <ul style="list-style-type: none"> Validación de login del alumno <i>Se tendrá que validar que la cuenta del docente exista</i> 	Salida: <p>Se mostrará un menú donde se mostrará una lista de información acerca del alumno</p> <p>La información que deberá mostrar es:</p> <ul style="list-style-type: none"> • Nombre del alumno • Matricula del alumno • Email del alumno <p>Deberá contener botones para:</p> <ul style="list-style-type: none"> • Generar código QR 	
Descripción: <p>Precondición: El alumno debe haber iniciado sesión previamente</p> <p>Descripción: El sistema permitirá al usuario verificar que su información es válida y poder generar un código QR para realizar su proceso de toma de asistencia</p> <p>Postcondición: El sistema permitirá al estudiante realizar su toma de asistencia</p>		
Manejo de situaciones anormales:		
Criterios de aceptación: <ul style="list-style-type: none"> • El usuario ha validado que su información es correcta • El sistema ha permitido al usuario generar su código QR de forma correcta 		

I - STUDENT Aplicación Web para Alumnos		
ID: Requerimiento Funcional 004	Nombre del Requerimiento: Cierre de sesión	
Tipo (Deseable / Necesario): Necesario	Categoría: Front - End	Prioridad de desarrollo: Alta
Entrada: <i>1. Botón de cerrar sesión</i>		Salida: La sesión del estudiante se ha cerrado correctamente
Descripción:		
<p>Precondición: El estudiante debe haber iniciado sesión en el sistema Descripción: El estudiante podrá cerrar su sesión actual en el sistema Postcondición: El sistema deberá regresar al apartado de Login</p>		
Manejo de situaciones anormales:		
<p>Criterios de aceptación:</p> <ul style="list-style-type: none"> • La sesión del estudiante se ha cerrado correctamente 		

Requerimientos No Funcionales del Software

I - STUDENT Aplicación Web para Alumnos		
ID: Requerimiento No Funcional 001	Nombre del Requerimiento: Diseño del registro de alumno	
Tipo (Deseable / Necesario): Deseado	Requerimiento que lo utiliza Requerimiento Funcional 001	Prioridad: Media
Descripción:		
<p>Requisito 1: Tener un recuadro para ingresar los datos solicitados Requisito 2: Mostrar como título “Registrar Cuenta” Requisito 3: Contener iconos relacionados a los inputs mostrados Requisito 4: Contener un botón para registrar la cuenta Requisito 5: Contener un botón para cambiar la pestaña a Login del alumno Requisito 6: Los caracteres ingresados en el input de contraseña deben ser intercambiados por asteriscos para evitar su visualización</p>		
<p>Criterios de aceptación:</p> <ul style="list-style-type: none"> • Los diseños estipulados se han cumplido correctamente en el diseño del sistema 		

I - STUDENT Aplicación Web para Alumnos		
ID: Requerimiento No Funcional 002	Nombre del Requerimiento: Diseño del login de alumnos	
Tipo (Deseable / Necesario): Deseado	Requerimiento que lo utiliza Requerimiento Funcional 002	Prioridad: Media
Descripción:		
<p>Requisito 1: Tener un recuadro para ingresar los datos solicitados</p> <p>Requisito 2: Mostrar como título “Login Cuenta”</p> <p>Requisito 3: Contener iconos relacionados a los inputs mostrados</p> <p>Requisito 4: Contener un botón para Iniciar Sesión</p> <p>Requisito 5: Contener un botón para cambiar pestaña a Registro</p> <p>Requisito 6: Los caracteres ingresados en el input de contraseña deben ser intercambiados por asteriscos para evitar su visualización</p>		
Criterios de aceptación:		
<ul style="list-style-type: none"> Los diseños estipulados se han cumplido correctamente en el diseño del sistema 		

I - STUDENT Aplicación Web para Alumnos		
ID: Requerimiento No Funcional 003	Nombre del Requerimiento: Diseño del menú de alumnos	
Tipo (Deseable / Necesario): Deseado	Requerimiento que lo utiliza Requerimiento Funcional 002	Prioridad: Media
Descripción:		
<p>Requisito 1: La información debe estar centrada</p> <p>Requisito 2: Utilizar una fuente clara</p> <p>Requisito 3: La imagen del alumno deberá estar recortada en forma circular</p> <p>Requisito 4: Deberá tener el nombre de la información a desplegar (Nombre del alumno, Apellido del alumno, Matricula, etc)</p> <p>Requisito 5: La información desplegada deberá ser de color negro</p>		
Criterios de aceptación:		
<ul style="list-style-type: none"> Los diseños estipulados se han cumplido correctamente en el diseño del sistema 		

2.5.3 API para interfaz web de docentes

Requerimientos Funcionales del Software

Servidor del sistema		
ID: Requerimiento Funcional 001	Nombre del Requerimiento: Conexión a la base de datos	
Tipo (Deseable / Necesario): Necesario	Categoría: Back - End	Prioridad de desarrollo: Alta
Entrada:		Salida:
<ol style="list-style-type: none"> 1. Type Se necesita especificar a qué base de datos se conectará mediante las credenciales 2. Host Se ingresará el host en el que estará alojada la base de datos del sistema 3. Port Se ingresará el puerto para poder acceder a la base de datos del sistema 4. Username Se ingresará el usuario correspondiente para el manejo de la base de datos del sistema 5. Password Se ingresará la contraseña del usuario correspondiente para el manejo de la base de datos del sistema 6. Database <i>Se ingresará el nombre de la base de datos que se estará utilizando para el funcionamiento del sistema</i> 	El servidor se deberá conectar correctamente a la base de datos y realizar los procesos solicitados por el sistema.	
Descripción:		
<p>Precondición: La base de datos deberá estar lista previamente a esto</p> <p>Descripción: El desarrollador deberá configurar correctamente el servidor para su conexión a la base de datos</p> <p>Postcondición: Ninguna</p>		
<p>Manejo de situaciones anormales:</p> <ul style="list-style-type: none"> • Los datos ingresados son erróneos con respecto a las credenciales del sistema, por lo que se mostrará un mensaje por consola • 		
<p>Criterios de aceptación:</p> <ul style="list-style-type: none"> • El servidor se conecta correctamente a la base de datos 		

Servidor del sistema		
ID: Requerimiento Funcional 002	Nombre del Requerimiento: Registro de cuenta Docente	
Tipo (Deseable / Necesario): Necesario	Categoría: Back - End	Prioridad de desarrollo: Alta
<p>Entrada:</p> <ol style="list-style-type: none"> Nombre del usuario Su tipo de dato será varchar, con un límite de 30 caracteres y aceptará caracteres especiales. Apellidos del usuario Su tipo de dato será varchar, con un límite de 30 caracteres y aceptará caracteres especiales. Número del personal Su tipo de dato será int, con un límite de 10 caracteres, teniendo una restricción con el siguiente formato: [0-9] [0-9] [0-9] [0-9] [0-9] [0-9] [0-9] [0-9] Correo Electrónico Su tipo de dato será varchar, con un límite de 30 caracteres y aceptará caracteres especiales. Contraseña Su tipo de dato será varchar, con un límite de 200 caracteres, debido a que se cifrará la contraseña con algoritmos HASH SHA-256 Foto <i>Su tipo de dato será varchar, con un límite de 200 caracteres, debido a que se recibirá una cadena de tipo base64</i> 	<p>Salida:</p> <p>Las cuentas se han registrado correctamente</p>	
<p>Descripción:</p> <p>Precondición: El servidor deberá estar conectado correctamente a la base de datos del sistema</p> <p>Descripción: El servidor deberá poder realizar el registro de las cuentas de docentes</p> <p>Postcondición: Ninguna</p> <p>Manejo de situaciones anormales:</p> <ul style="list-style-type: none"> Los datos ingresados son erróneos con respecto a las restricciones planteadas <p>Criterios de aceptación:</p> <ul style="list-style-type: none"> El registro de las cuentas se realiza correctamente 		

Servidor del sistema		
ID: Requerimiento Funcional 003	Nombre del Requerimiento: Validación de Inicio de Sesión Docente	
Tipo (Deseable / Necesario): Necesario	Categoría: Back - End	Prioridad de desarrollo: Alta
Entrada: 1. Correo Electrónico <i>Su tipo de dato será varchar, con un límite de 30 caracteres y aceptará caracteres especiales.</i> 2. Contraseña <i>Su tipo de dato será varchar, con un límite de 200 caracteres</i>		Salida: La base de datos permitirá la validación del inicio de sesión de las cuentas de Docentes en el sistema
Descripción:		
Precondición: El docente debe tener una cuenta registrada en el sistema Descripción: El procedimiento almacenado permitirá la validación del inicio de sesión, a través de la aceptación de dos valores, los cuales serán Correo Electrónico y la contraseña. Este procedimiento almacenado realizará la consulta de la verificación directamente sobre la tabla Cuentas Docente Postcondición: Ninguna		
Manejo de situaciones anormales: <ul style="list-style-type: none"> • El correo electrónico no coincide con ninguno registrado en el sistema (No permitirá realizar el proceso) • La contraseña no coincide con la almacenada en el sistema (No permitirá realizar el proceso) 		
Criterios de aceptación: <ul style="list-style-type: none"> • La verificación del inicio de sesión se realiza correctamente 		

Servidor del sistema		
ID: Requerimiento Funcional 004	Nombre del Requerimiento: Alta Materias-Docentes	
Tipo (Deseable / Necesario): Necesario	Categoría: Back - End	Prioridad de desarrollo: Alta
Entrada: 1. Número del personal del Docente que la impartirá <i>Su tipo de dato será int, con un límite de 10 caracteres, teniendo como llave foránea el número del personal de la tabla Docentes</i> 2. NRC de la materia <i>Su tipo de dato será int, con un límite de 5 caracteres, teniendo como llave foránea el NRC de la tabla Materias</i>	Salida: La base de datos permitirá la relación de una materia con un docente en una tabla de la base de datos	
Descripción:		
Precondición: La materia y el Docente deben estar registrados previamente en el sistema Descripción: La base de datos deberá contar con una tabla que permita la relación de las materias con los Docentes a través de los dos campos ingresados. Postcondición: Ninguna		
Manejo de situaciones anormales: <ul style="list-style-type: none"> El NRC de la materia no coincide con ningún dato almacenado en el sistema (No permitirá realizar el proceso) El número del personal del docente no coincide con ningún dato almacenado en el sistema (No permitirá realizar el proceso) La materia ya está relacionada con un Docente (No permitirá realizar el proceso) El horario de la materia coincide con algún horario del docente (No permitirá realizar el proceso) 		
Criterios de aceptación:		
<ul style="list-style-type: none"> La alta materias se realiza correctamente 		

Servidor del sistema		
ID: Requerimiento Funcional 005	Nombre del Requerimiento: Baja Materias-Docentes	
Tipo (Deseable / Necesario): Necesario	Categoría: Back - End	Prioridad de desarrollo: Alta
Entrada: 1. Número del personal del Docente <i>Su tipo de dato será int, con un límite de 10 caracteres, teniendo como llave foránea el número del personal de la tabla Docentes</i> 2. NRC de la materia <i>Su tipo de dato será int, con un límite de 5 caracteres, teniendo como llave foránea el NRC de la tabla Materias</i>		Salida: La base de datos permitirá eliminar la relación existente entre una materia y un Docente
Descripción:		
Precondición: La materia y el Docente deben estar registrados previamente en el sistema Descripción: La base de datos deberá contar con una tabla que permita la relación de las materias con los Docentes a través de los dos campos ingresados. Postcondición: Ninguna		
Manejo de situaciones anormales: <ul style="list-style-type: none"> El NRC de la materia no coincide con ningún dato almacenado en el sistema (No permitirá realizar el proceso) El número del personal del docente no coincide con ningún dato almacenado en el sistema (No permitirá realizar el proceso) Criterios de aceptación: <ul style="list-style-type: none"> La baja de materias se realiza correctamente 		

Servidor del sistema		
ID: Requerimiento Funcional 006	Nombre del Requerimiento: Información Docente	
Tipo (Deseable / Necesario): Necesario	Categoría: Back - End	Prioridad de desarrollo: Alta
Entrada: 1) Número del personal del Docente <i>Su tipo de dato será int, con un límite de 10 caracteres, teniendo como llave foránea el número del personal de la tabla Docentes</i>	Salida: La API consultará a la base de datos para poder obtener la información relacionada al docente, la cual será: 1. Nombre del docente 2. Numero del personal 3. Facultad asignada 4. Imagen del docente	
Descripción:		
Precondición: El Docente debe estar registrados previamente en el sistema Descripción: La base de datos deberá contar con una tabla que permita la consulta de la información del Docente Postcondición: Ninguna		
Manejo de situaciones anormales:		
<ul style="list-style-type: none"> • El número del personal del docente no coincide con ningún dato almacenado en el sistema (No permitirá realizar el proceso) 		
Criterios de aceptación:		
<ul style="list-style-type: none"> • La consulta de información se realiza de forma correcta 		

Servidor del sistema		
ID: Requerimiento Funcional 007	Nombre del Requerimiento: Materias-Docentes	
Tipo (Deseable / Necesario): Necesario	Categoría: Back - End	Prioridad de desarrollo: Alta
<p>Entrada:</p> <p>1. Número del personal del Docente <i>Su tipo de dato será int, con un límite de 10 caracteres, teniendo como llave foránea el número del personal de la tabla Docentes</i></p>		<p>Salida:</p> <p>La API consultará a la base de datos para poder obtener la información relacionada al docente, la cual será:</p> <ol style="list-style-type: none"> 1. Nombre de la materia 2. NRC de la materia 3. Licenciatura asignada 4. Grupo asignado 5. Turno asignado
Descripción:		
<p>Precondición: La materia y el Docente deben estar registrados previamente en el sistema</p> <p>Descripción: La base de datos deberá contar con una tabla que permita la relación de las materias con los Docentes a través de los dos campos ingresados.</p> <p>Postcondición: Ninguna</p>		
<p>Manejo de situaciones anormales:</p> <ul style="list-style-type: none"> • El número del personal del docente no coincide con ningún dato almacenado en el sistema (No permitirá realizar el proceso) <p>Criterios de aceptación:</p> <ul style="list-style-type: none"> • La consulta de información se realiza correctamente 		

2.5.4 API para interfaz web de alumnos

Requerimientos Funcionales del Software

Servidor del sistema		
ID: Requerimiento Funcional 001	Nombre del Requerimiento: Conexión a la base de datos	
Tipo (Deseable / Necesario): Necesario	Categoría: Back - End	Prioridad de desarrollo: Alta
Entrada:		Salida: El servidor se deberá conectar correctamente a la base de datos y realizar los procesos solicitados por el sistema.
1. Type Se necesita especificar a qué base de datos se conectará mediante las credenciales 2. Host Se ingresará el host en el que estará alojada la base de datos del sistema 3. Port Se ingresará el puerto para poder acceder a la base de datos del sistema 4. Username Se ingresará el usuario correspondiente para el manejo de la base de datos del sistema 5. Password Se ingresará la contraseña del usuario correspondiente para el manejo de la base de datos del sistema 6. Database <i>Se ingresará el nombre de la base de datos que se estará utilizando para el funcionamiento del sistema</i>		
Descripción:		
Precondición: La base de datos deberá estar lista previamente a esto Descripción: El desarrollador deberá configurar correctamente el servidor para su conexión a la base de datos Postcondición: Ninguna		
Manejo de situaciones anormales: <ul style="list-style-type: none"> Los datos ingresados son erróneos con respecto a las credenciales del sistema, por lo que se mostrará un mensaje por consola 		
Criterios de aceptación: <ul style="list-style-type: none"> El servidor se conecta correctamente a la base de datos 		

Servidor del sistema		
ID: Requerimiento Funcional 002	Nombre del Requerimiento: Registro de cuenta Alumno	
Tipo (Deseable / Necesario): Necesario	Categoría: Back - End	Prioridad de desarrollo: Alta
Entrada: <ol style="list-style-type: none"> Nombre del usuario Su tipo de dato será varchar, con un límite de 30 caracteres y aceptará caracteres especiales. Apellidos del usuario Su tipo de dato será varchar, con un límite de 30 caracteres y aceptará caracteres especiales. Matrícula Su tipo de dato será varchar, con un límite de 10 caracteres, teniendo una restricción con el siguiente formato: zS[0-9] [0-9] [0-9] [0-9] [0-9] [0-9] [0-9] Correo Electrónico Su tipo de dato será varchar, con un límite de 30 caracteres y aceptará caracteres especiales. Contraseña Su tipo de dato será varchar, con un límite de 200 caracteres, debido a que se cifrará la contraseña con algoritmos HASH SHA-256 Foto <i>Su tipo de dato será varchar, con un límite de 200 caracteres, debido a que se recibirá una cadena de tipo base64</i> 	Salida: Las cuentas se han registrado correctamente	
Descripción:		
Precondición: El servidor deberá contar con una conexión a la base de datos Descripción: La base de datos deberá poder realizar el registro de las cuentas de alumnos Postcondición: Ninguna		
Manejo de situaciones anormales:		
<ul style="list-style-type: none"> Los datos ingresados son erróneos con respecto a las restricciones planteadas 		
Criterios de aceptación:		
<ul style="list-style-type: none"> El registro de las cuentas se realiza correctamente 		

Servidor del sistema		
ID: Requerimiento Funcional 003	Nombre del Requerimiento: Validación de Inicio de Sesión Alumno	
Tipo (Deseable / Necesario): Necesario	Categoría: Back - End	Prioridad de desarrollo: Alta
Entrada: 1. Correo Electrónico <i>Su tipo de dato será varchar, con un límite de 30 caracteres y aceptará caracteres especiales.</i> 2. Contraseña <i>Su tipo de dato será varchar, con un límite de 200 caracteres</i>		Salida: La base de datos permitirá la validación del inicio de sesión de las cuentas de Alumnos en el sistema
Descripción:		
Precondición: El Alumno debe tener una cuenta registrada en el sistema Descripción: El procedimiento almacenado permitirá la validación del inicio de sesión, a través de la aceptación de dos valores, los cuales serán Correo Electrónico y la contraseña. Este procedimiento almacenado realizará la consulta de la verificación directamente sobre la tabla Cuentas Alumnos Postcondición: Ninguna		
Manejo de situaciones anormales: <ul style="list-style-type: none"> • El correo electrónico no coincide con ninguno registrado en el sistema (No permitirá realizar el proceso) • La contraseña no coincide con la almacenada en el sistema (No permitirá realizar el proceso) 		
Criterios de aceptación: <ul style="list-style-type: none"> • La verificación del inicio de sesión se realiza correctamente 		

Servidor del sistema		
ID: Requerimiento Funcional 004	Nombre del Requerimiento: Información Alumno	
Tipo (Deseable / Necesario): Necesario	Categoría: Back - End	Prioridad de desarrollo: Alta
Entrada: 1) Matricula del estudiante <i>Su tipo de dato será int, con un límite de 10 caracteres, teniendo como llave foránea el número del personal de la tabla Docentes</i>		Salida: La API consultará a la base de datos para poder obtener la información relacionada al docente, la cual será: 1. Nombre del alumno 2. Matricula del alumno 3. Email del alumno
Descripción:		
Precondición: El alumno debe estar registrados previamente en el sistema Descripción: La base de datos deberá contar con una tabla que permita la consulta de la información del alumno Postcondición: Ninguna		
Manejo de situaciones anormales:		
<ul style="list-style-type: none"> • La matrícula del alumno no coincide con ningún dato almacenado en el sistema (No permitirá realizar el proceso) 		
Criterios de aceptación:		
<ul style="list-style-type: none"> • La consulta de información se realiza de forma correcta 		

Servidor del sistema		
ID: Requerimiento Funcional 005	Nombre del Requerimiento: Materias-Alumnos	
Tipo (Deseable / Necesario): Necesario	Categoría: Back - End	Prioridad de desarrollo: Alta
Entrada: 1. Matricula del alumno <i>Su tipo de dato será int, con un límite de 10 caracteres, teniendo como llave foránea el número del personal de la tabla Docentes</i>	Salida: La API consultará a la base de datos para poder obtener la información relacionada al docente, la cual será: 1. Nombre de la materia 2. NRC de la materia 3. Licenciatura asignada 4. Grupo asignado 5. Turno asignado	
Descripción: Precondición: La materia y el alumno deben estar registrados previamente en el sistema Descripción: La base de datos deberá contar con una tabla que permita la relación de las materias con los alumnos a través de los dos campos ingresados. Postcondición: Ninguna		
Manejo de situaciones anormales: <ul style="list-style-type: none"> La matrícula del alumno no coincide con ningún dato almacenado en el sistema (No permitirá realizar el proceso) Criterios de aceptación: <ul style="list-style-type: none"> La consulta de información se realiza correctamente 		

2.5.5 Sistema para lectura de códigos QR

Requerimientos Funcionales del Software

I - STUDENT Aplicación Web para lectura de QR		
ID: Requerimiento Funcional 001	Nombre del Requerimiento: Encendido del equipo	
Tipo (Deseable / Necesario): Necesario	Categoría: Front - End	Prioridad de desarrollo: Alta
Entrada: 1. Botón Encendido <i>Se mostrará en pantalla un botón para el encendido de la cámara del equipo</i>	Salida: El encendido de la cámara del dispositivo se iniciará mostrando una previalización	
Descripción:		
<p>Precondición: El usuario debe encontrarse en la aplicación web</p> <p>Descripción: El sistema permitirá su iniciación de la cámara para su utilización</p> <p>Postcondición: Ninguna</p>		
Manejo de situaciones anormales:		
<ul style="list-style-type: none"> • 		
Criterios de aceptación:		
<ul style="list-style-type: none"> • La cámara del dispositivo se inicia correctamente 		

I - STUDENT Aplicación Web para lectura de QR		
ID: Requerimiento Funcional 002	Nombre del Requerimiento: Apagado del equipo	
Tipo (Deseable / Necesario): Necesario	Categoría: Front - End	Prioridad de desarrollo: Alta
Entrada: 2. Botón Apagado <i>Se deberá presionar el botón de apagado</i>	Salida: La cámara del dispositivo se apagará correctamente	
Descripción:		
<p>Precondición: El usuario debe encontrarse en la aplicación web</p> <p>Descripción: El sistema permitirá el apagado correcto de su cámara</p> <p>Postcondición: Ninguna</p>		
Manejo de situaciones anormales:		
<ul style="list-style-type: none"> • 		
Criterios de aceptación:		
<ul style="list-style-type: none"> • La cámara se apaga correctamente 		

I - STUDENT Aplicación Web para lectura de QR		
ID: Requerimiento Funcional 003	Nombre del Requerimiento: Configuración del equipo	
Tipo (Deseable / Necesario): Necesario	Categoría: Front - End	Prioridad de desarrollo: Alta
Entrada: 1. Salón <i>Se mostrará un checkbox con una serie de opciones que tenga el sistema para poder ser configurado por parte del usuario</i> 2. Edificio <i>Se mostrará un checkbox con una serie de opciones que tenga el sistema para poder ser configurado por parte del usuario</i> 3. Botón de Aceptar <i>Se deberá presionar un botón para que el usuario pueda aceptar la configuración dada al sistema</i>	Salida: El sistema extraerá la lista de materias que corresponden a la configuración dada por el usuario, y funcionará en conjunto con los horarios registrados en cada una con la hora extraída por el sistema.	
Descripción:		
Precondición: El usuario debe realizar el encendido del sistema Descripción: El sistema permitirá la configuración de este para que funcione correctamente y como espera el usuario Postcondición: El usuario deberá presionar el botón de aceptación de la configuración para su posterior utilización en el proceso de la toma de asistencia		
Manejo de situaciones anormales: <ul style="list-style-type: none"> • El salón no ha sido cargado con materias para el proceso de toma de asistencia Criterios de aceptación: <ul style="list-style-type: none"> • El sistema se configura correctamente 		

I - STUDENT Aplicación Web para lectura de QR		
ID: Requerimiento Funcional 004	Nombre del Requerimiento: Opciones Disponibles	
Tipo (Deseable / Necesario): Necesario	Categoría: Front - End	Prioridad de desarrollo: Alta
Entrada: <i>1. Botones sobre las opciones disponibles del sistema</i>		Salida: El sistema mostrará en pantalla una serie de botones para escoger, los cuales contendrán las opciones disponibles para el uso del usuario, como lo son: <ul style="list-style-type: none">• Escaneo de códigos QR para el proceso del pase de lista• Configuración del dispositivo• Apagado del dispositivo
Descripción: Precondición: El usuario debe haber iniciado el sistema Descripción: El sistema deberá mostrar la serie de opciones disponibles para el uso del usuario Postcondición: El usuario deberá seleccionar alguna de las opciones disponibles por el sistema		
Manejo de situaciones anormales: <ul style="list-style-type: none">• 		
Criterios de aceptación: <ul style="list-style-type: none">• La elección de una opción se ejecuta de manera correcta		

I - STUDENT Aplicación Web para lectura de QR		
ID: Requerimiento Funcional 005	Nombre del Requerimiento: Escaneo de códigos QR	
Tipo (Deseable / Necesario): Necesario	Categoría: Front - End	Prioridad de desarrollo: Alta
Entrada: 1. Código QR <i>El sistema podrá leer los códigos QR generados por la aplicación de Estudiantes, extrayendo la información contenida en estos.</i>	Salida: El sistema podrá descifrar la información contenida en los códigos QR, la cual será la siguiente: <ul style="list-style-type: none"> • Matricula • Nombre • Status El sistema mostrará un mensaje al terminar de realizar el proceso de asistencia, mostrando la información extraída anteriormente.	
Descripción:		
Precondición: El usuario debe haber habilitado la opción de escaneo de códigos QR Descripción: El sistema podrá realizar el escaneo de los códigos QR Postcondición: Ninguna		
Manejo de situaciones anormales: <ul style="list-style-type: none"> • No se ha podido leer correctamente el código QR, por lo que no realizará el proceso de toma de asistencia y no mostrará la información del estudiante • El estudiante que realiza el proceso de asistencia no se encuentra registrado en la materia, por lo que mostrará un mensaje en pantalla • El sistema no puede realizar el proceso de toma de asistencia debido a problemas de internet, mostrando un mensaje por pantalla Criterios de aceptación: <ul style="list-style-type: none"> • El sistema realizar correctamente el proceso de la toma de asistencia 		

I - STUDENT Aplicación Web para lectura de QR		
ID: Requerimiento Funcional 006	Nombre del Requerimiento: Historial de escaneo de códigos QR	
Tipo (Deseable / Necesario): Necesario	Categoría: Front - End	Prioridad de desarrollo: Alta
Entrada: 1. Código QR <i>Por medio de un dispositivo de hardware se escaneará un código QR para su lectura de datos</i> 2. Hora de lectura <i>A través del uso de bibliotecas JavaScript se almacenará la hora en que se escaneo un el código para su posterior consulta</i>		Salida: El sistema deberá ser capaz de almacenar temporalmente todos los códigos que han sido leídos, junto a sus horas de lectura
Descripción:		
Precondición: El sistema debe haber realizado la lectura del código QR Descripción: Se almacenará el registro del código en la base de datos Postcondición: Se podrán consultar todos los registros de los códigos QR escaneados durante el día		
Manejo de situaciones anormales: <ul style="list-style-type: none"> • No se han tomado lecturas de código QR, por lo que mostrará un mensaje en pantalla • Los códigos QR solo se almacenarán por un periodo temporal, por lo que no se podrán consultar registros que pasen de este límite 		
Criterios de aceptación: <ul style="list-style-type: none"> • El historial se almacena perfectamente de forma temporal 		

I - STUDENT Aplicación Web para lectura de QR		
ID: Requerimiento Funcional 007	Nombre del Requerimiento: Realizar proceso de asistencia	
Tipo (Deseable / Necesario): Necesario	Categoría: Front - End	Prioridad de desarrollo: Alta
Entrada: 1. Matricula del alumno <i>Se extraerá la matrícula del alumno por medio de la lectura del código QR</i> 2. Nombre del alumno <i>Se extraerá el nombre del alumno por medio de la lectura del código QR</i> 3. Status del alumno <i>Se extraerá el status del alumno por medio de la lectura del código QR</i> 4. Hora del escaneo <i>Se extraerá la hora exacta del momento en que fue escaneado el código QR del alumno</i>	Salida: El sistema deberá ser capaz de realizar correctamente el proceso de asistencia del alumno con los dos datos ingresados.	
Descripción:		
<p>Precondición: El sistema debe haber realizado la lectura del código QR</p> <p>Descripción: El sistema hará uso de la hora obtenida, el salón y el edificio en el que fue configurado previamente, para poder saber en qué materia se encuentra la hora del registro, pudiendo así realizar dicha asistencia en la materia con la matrícula del alumno obtenida por el escaneo del código QR</p> <p>Postcondición: El sistema deberá mostrar un mensaje en pantalla de que la asistencia se realizó correctamente</p>		
<p>Manejo de situaciones anormales:</p> <ul style="list-style-type: none"> • No se ha tomado correctamente la lectura del código QR, por lo que mostrará un mensaje en pantalla • El alumno no está registrado en la materia para su toma de asistencia, por lo que se mostrará un mensaje en pantalla • El sistema no ha podido realizar la toma de asistencia debido a un problema de conexión a internet, por lo que mostrará un mensaje en pantalla 		
Criterios de aceptación:		
<ul style="list-style-type: none"> • El sistema realiza la toma de asistencia correctamente 		

Requerimientos No Funcionales del Software

I - STUDENT Aplicación Web para lectura de QR		
ID: Requerimiento No Funcional 001	Nombre del Requerimiento: Mostrar información	
Tipo (Deseable / Necesario): Deseable	Requerimiento que lo utiliza Requerimiento Funcional 001	Prioridad: Media
Descripción:		
<ul style="list-style-type: none"> • El sistema debe ser capaz de mostrar la información codificada del código QR al usuario, de una forma clara y legible 		
Criterios de aceptación:		
<ul style="list-style-type: none"> • Los puntos estipulados se han cumplido correctamente en el diseño del sistema 		

I - STUDENT Aplicación Web para lectura de QR		
ID: Requerimiento No Funcional 002	Nombre del Requerimiento: Escaneo de código QR	
Tipo (Deseable / Necesario): Deseable	Requerimiento que lo utiliza Requerimiento Funcional 001	Prioridad: Media
Descripción:		
<ul style="list-style-type: none"> • La decodificación del código QR debe ser de manera rápida y eficiente para garantizar una experiencia de usuario sin interrupciones 		
Criterios de aceptación:		
<ul style="list-style-type: none"> • Los puntos estipulados se han cumplido correctamente en el diseño del sistema 		

I - STUDENT		
Aplicación Web para lectura de QR		
ID: Requerimiento No Funcional 003	Nombre del Requerimiento: Escaneo de código QR	
Tipo (Deseable / Necesario): Deseable	Requerimiento que lo utiliza Requerimiento Funcional 001	Prioridad: Media
Descripción:		
<ul style="list-style-type: none"> • El sistema debe ser capaz de escanear códigos QR en tiempo real 		
Criterios de aceptación:		
<ul style="list-style-type: none"> • Los puntos estipulados se han cumplido correctamente en el diseño del sistema 		

I - STUDENT		
Aplicación Web para lectura de QR		
ID: Requerimiento No Funcional 004	Nombre del Requerimiento: Actualizar información	
Tipo (Deseable / Necesario): Deseable	Requerimiento que lo utiliza Requerimiento Funcional 001	Prioridad: Media
Descripción:		
<ul style="list-style-type: none"> • Debe actualizar la información a medida que se realiza la lectura. 		
Criterios de aceptación:		
<ul style="list-style-type: none"> • Los puntos estipulados se han cumplido correctamente en el diseño del sistema 		

I - STUDENT Aplicación Web para lectura de QR		
ID: Requerimiento No Funcional 005	Nombre del Requerimiento: Historial de escaneos	
Tipo (Deseable / Necesario): Deseable	Requerimiento que lo utiliza Requerimiento Funcional 002	Prioridad: Media
Descripción:		
<ul style="list-style-type: none"> • Debe contener un historial temporal acerca de la información de los códigos QR escaneados • Solo debe almacenarse la matrícula del estudiante junto a la hora del registro • La eliminación de los registros debe ser de manera automática • El guardado de los registros debe ser de manera automática 		
Criterios de aceptación:		
<ul style="list-style-type: none"> • Los puntos estipulados se han cumplido correctamente en el diseño del sistema 		

I - STUDENT Aplicación Web para lectura de QR		
ID: Requerimiento No Funcional 007	Nombre del Requerimiento: Seguridad de la información	
Tipo (Deseable / Necesario): Deseable	Requerimiento que lo utiliza Requerimiento Funcional 001	Prioridad: Media
Descripción:		
<ul style="list-style-type: none"> • Se debe garantizar la seguridad de los datos obtenidos a través de los códigos QR escaneados y proteger la privacidad de la información del usuario 		
Criterios de aceptación:		
<ul style="list-style-type: none"> • Los puntos estipulados se han cumplido correctamente en el diseño del sistema 		

I - STUDENT Aplicación Web para lectura de QR		
ID: Requerimiento No Funcional 008	Nombre del Requerimiento: Mensaje de estado	
Tipo (Deseable / Necesario): Deseable	Requerimiento que lo utiliza Requerimiento Funcional 001	Prioridad: Media
Descripción:		
<ul style="list-style-type: none"> • Se debe proporcionar retroalimentación al usuario para indicar que el escaneo fue exitoso o si sufrió de un problema durante el proceso. 		
Criterios de aceptación: <ul style="list-style-type: none"> • Los puntos estipulados se han cumplido correctamente en el diseño del sistema 		

I - STUDENT Aplicación Web para lectura de QR		
ID: Requerimiento No Funcional 009	Nombre del Requerimiento: Usabilidad	
Tipo (Deseable / Necesario): Deseable	Requerimiento que lo utiliza Requerimiento Funcional 001	Prioridad: Media
Descripción:		
<ul style="list-style-type: none"> • El sistema debe ser fácil de usar y comprender para los usuarios de diferentes niveles de habilidad 		
Criterios de aceptación: <ul style="list-style-type: none"> • Los puntos estipulados se han cumplido correctamente en el diseño del sistema 		

I - STUDENT Aplicación Web para lectura de QR		
ID: Requerimiento No Funcional 010	Nombre del Requerimiento: Disponibilidad	
Tipo (Deseable / Necesario): Deseable	Requerimiento que lo utiliza Requerimiento Funcional 001	Prioridad: Media
Descripción:		
<ul style="list-style-type: none"> • El sistema debe estar disponible en todo momento y ser capaz de manejar un alto volumen de solicitudes de escaneo 		
Criterios de aceptación:		
<ul style="list-style-type: none"> • Los puntos estipulados se han cumplido correctamente en el diseño del sistema 		

I - STUDENT Aplicación Web para lectura de QR		
ID: Requerimiento No Funcional 011	Nombre del Requerimiento: Aumento de las solicitudes	
Tipo (Deseable / Necesario): Deseable	Requerimiento que lo utiliza Requerimiento Funcional 001	Prioridad: Media
Descripción:		
<ul style="list-style-type: none"> • El sistema debe ser capaz de escalar para manejar un aumento en la demanda sin comprometer la calidad o la velocidad del servicio 		
Criterios de aceptación:		
<ul style="list-style-type: none"> • Los puntos estipulados se han cumplido correctamente en el diseño del sistema 		

I - STUDENT Aplicación Web para lectura de QR		
ID: Requerimiento No Funcional 012	Nombre del Requerimiento: Portabilidad del sistema	
Tipo (Deseable / Necesario): Deseable	Requerimiento que lo utiliza Requerimiento Funcional 001	Prioridad: Media
Descripción:		
<ul style="list-style-type: none"> • El sistema debe ser portátil y capaz de funcionar en diferentes plataformas y dispositivos. 		
Criterios de aceptación:		
<ul style="list-style-type: none"> • Los puntos estipulados se han cumplido correctamente en el diseño del sistema 		

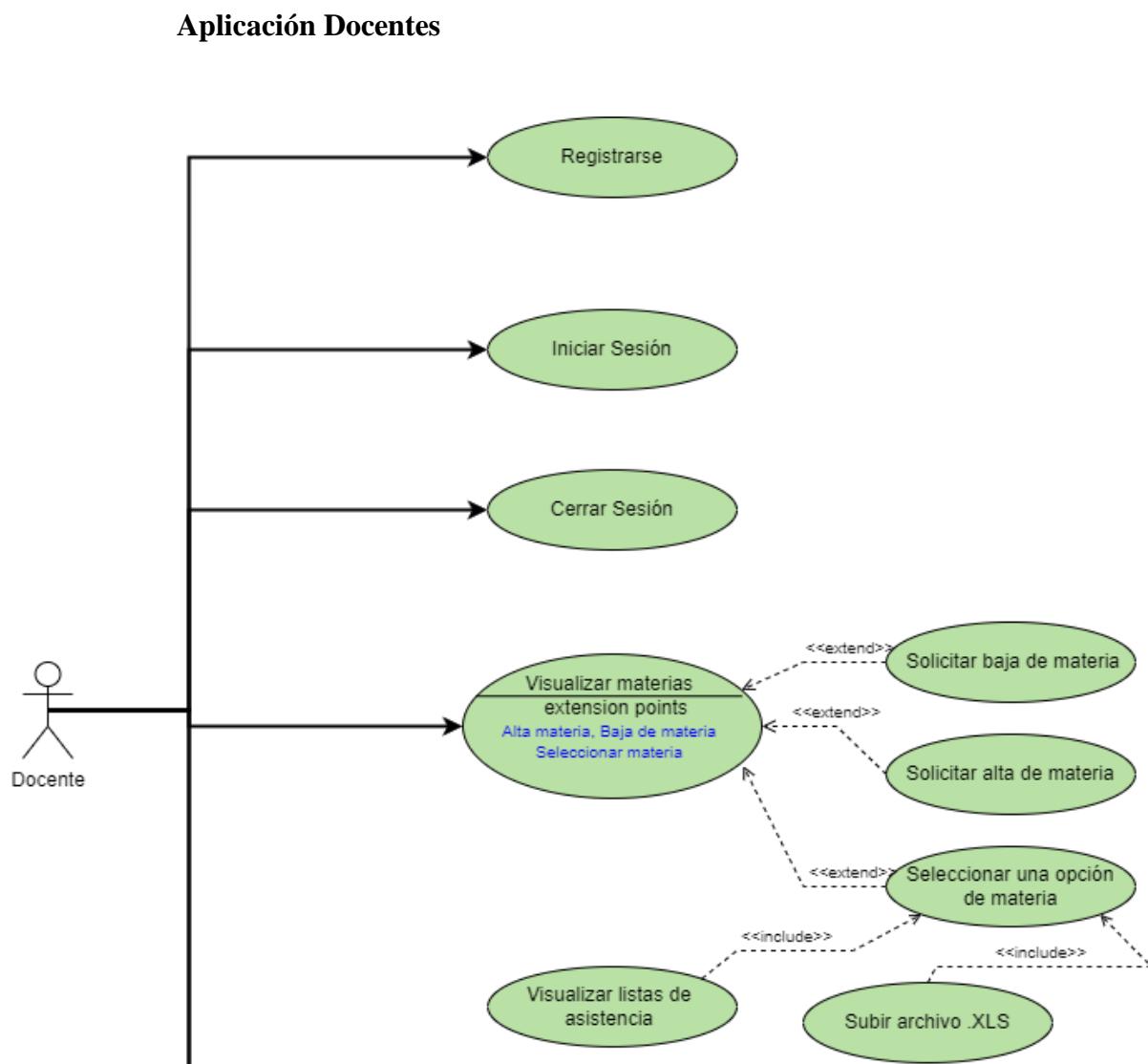
I - STUDENT Aplicación Web para lectura de QR		
ID: Requerimiento No Funcional 013	Nombre del Requerimiento: Mantenimiento del sistema	
Tipo (Deseable / Necesario): Deseable	Requerimiento que lo utiliza Requerimiento Funcional 001	Prioridad: Media
Descripción:		
<ul style="list-style-type: none"> • El sistema debe ser fácil de mantener y actualizar para corregir errores y mejorar el rendimiento 		
Criterios de aceptación:		
<ul style="list-style-type: none"> • Los puntos estipulados se han cumplido correctamente en el diseño del sistema 		

I - STUDENT Aplicación Web para lectura de QR		
ID: Requerimiento No Funcional 014	Nombre del Requerimiento: Adaptabilidad del sistema	
Tipo (Deseable / Necesario): Deseable	Requerimiento que lo utiliza Requerimiento Funcional 001	Prioridad: Media
Descripción:		
<ul style="list-style-type: none"> • El sistema debe ser capaz de adaptarse a diferentes entornos y situaciones de uso, como la iluminación baja o códigos QR borrosos 		
Criterios de aceptación:		
<ul style="list-style-type: none"> • Los puntos estipulados se han cumplido correctamente en el diseño del sistema 		

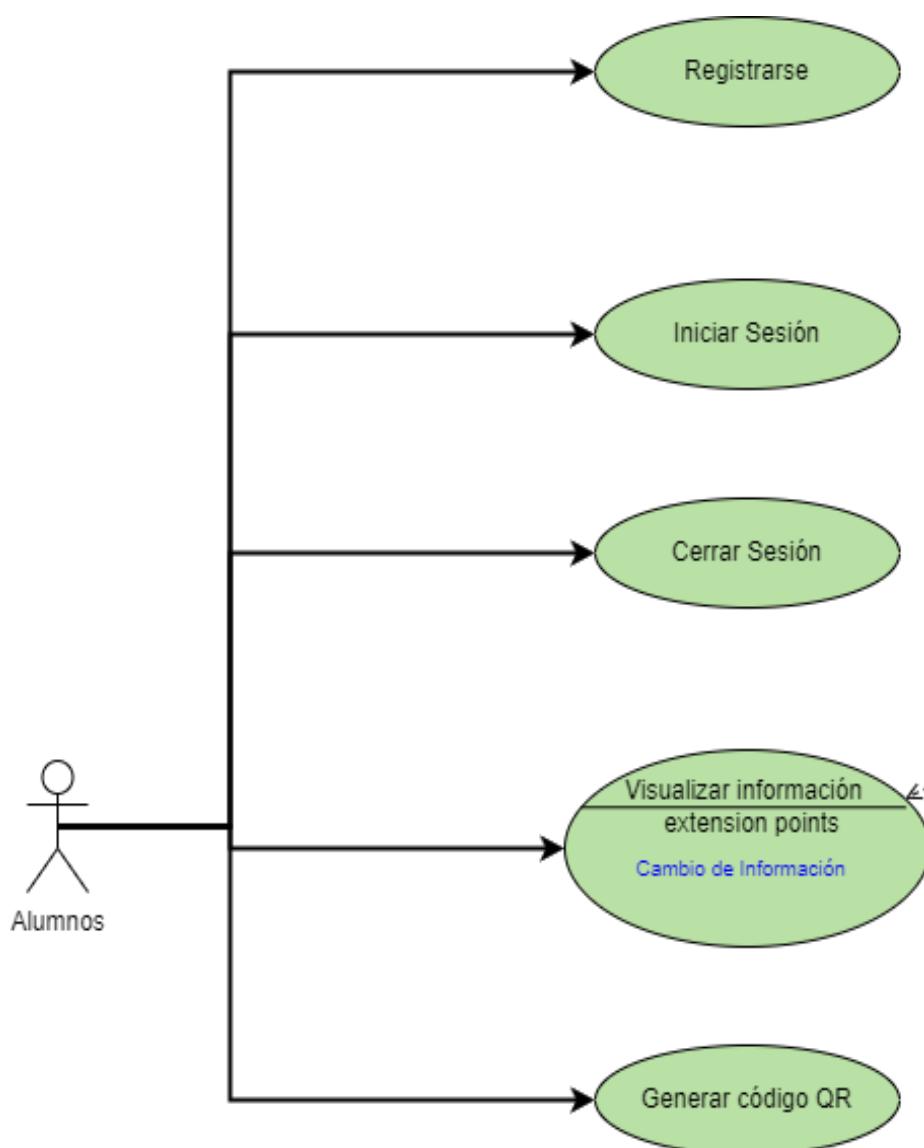
2.6 Análisis y diseño

La especificación del diseño del sistema es un proceso de detallar cómo será el sistema visualmente, así como también mostrar cómo será su funcionamiento a través de diferentes diagramas UML.

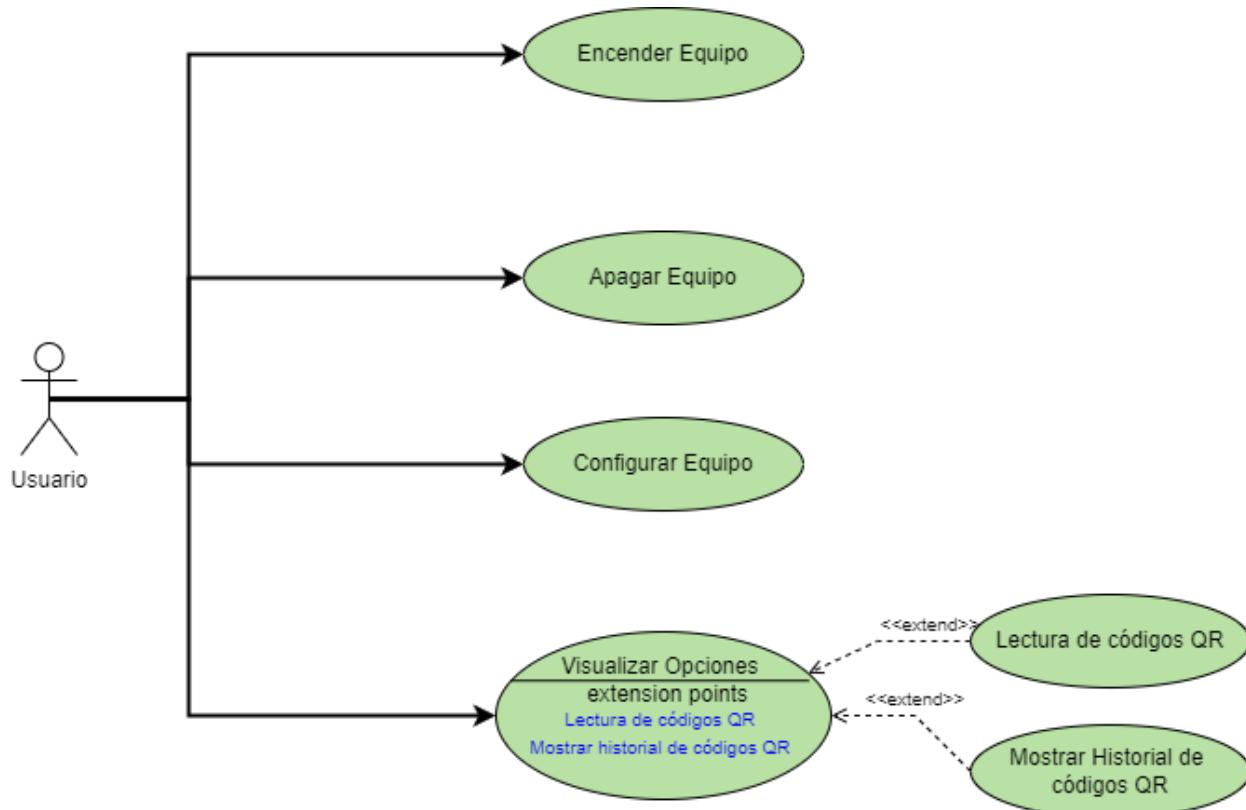
2.6.1 Casos de uso



Aplicación Alumnos



Simulador Escáner



2.6.2 Especificación de casos de uso

Aplicación web para Docentes

I - Student			
1.		Proyecto:	Aplicación web para Docentes
2.	Caso de uso:	CU01	Registro de cuenta
3.	Destinatario:	Desarrolladores	
4.	Descripción:	Este caso de uso representa al docente realizar su registro de cuenta en el sistema	
5.	Autores:	Irving R. Conde Marín	
6.	Actores:	Docente	
7.	Requerimientos:	Registro de docente	
8.	Verificado por:		
9.	Precondiciones:	Contar con su número de personal	
10.	Postcondiciones:	Ninguna	
11.	Flujo principal:	Paso	Acción
		1	Verificar que todos los campos estén llenados por el usuario
		2	Mandar los datos para el proceso de registro
		3	Verificar que los datos cumplen con las restricciones establecidas en los requerimientos
		4	Verificar que el número de docente no ha sido registrado con anterioridad en el sistema
		5	Guardar datos de la cuenta
		6	Mostrar mensaje: "La cuenta se ha registrado exitosamente"
		7	Finalizar
12.	Flujo secundario:		
13.	Excepciones:	Pasos	Acción
		1	El usuario ya se ha registrado con anterioridad, por lo cual se mostrará un mensaje por pantalla: "El usuario ya se ha registrado anteriormente, inicia sesión con tu correo electrónico ingresado"
		2	El usuario no ha llenado todos los campos del registro, por lo cual el sistema mostrará un mensaje por pantalla: "Por favor llene todos los campos para realizar el proceso de registro"
		3	El número de personal ingresado en el campo no cumple con el formato estipulado en los requerimientos del sistema, por lo cual el sistema mostrará un mensaje por pantalla: "El formato del número del personal ingresado no corresponde con el estipulado"
		4	La contraseña ingresada en el campo no cumple con el formato estipulado en los requerimientos del sistema, por lo cual mostrará un mensaje por pantalla: "La contraseña no coincide con las restricciones dadas, debe contener al menos 8 caracteres, entre ellos 1 mayúscula, 1 número y un carácter especial"
		5	El peso de la imagen a subir es mayor al permitido por el sistema, por lo que se mostrará un mensaje por pantalla: "No se ha podido realizar el registro debido a que el peso de la imagen es mayor al permitido"
		6	Si el usuario presenta problemas de conexión a internet, y esto provoca que no puedan mandarse los datos para su registro, se mostrará un mensaje por pantalla: "No se ha podido realizar el proceso de registro, verifique que tenga una conexión a internet"
		7	El correo electrónico ingresado en el campo no existe, por lo que el registro no podrá llevarse a cabo, mostrando un mensaje en pantalla: "Ingrese un correo electrónico existente"
		8	El número de personal ingresado no coincide con ninguno almacenado en la base de datos, mostrando un mensaje en pantalla: "El número del personal ingresado no corresponde a ningún dato almacenado, por favor ingrese un dato válido"
14.	Prioridad:	Necesario	

I - Student			
1.		Proyecto:	Aplicación web para Docentes
2.	Caso de uso:	CU02	Inicio de Sesión
3.	Destinatario:	Desarrolladores	
4.	Descripción:	Este caso de uso representa al docente iniciar sesión en el sistema con su cuenta registrada	
5.	Autores:	Irving R. Conde Marín	
6.	Actores:	Docente	
7.	Requerimientos:	Login de docente	
8.	Verificado por:		
9.	Precondiciones:	Contar con una cuenta registrada en el sistema	
10.	Postcondiciones:	Ninguna	
11.	Flujo principal:	Paso	Acción
		1	Verificar que todos los campos estén llenados por el usuario
		2	Mandar los datos para el proceso de inicio de sesión
		3	Verificar que el correo electrónico ingresado esté registrado en el sistema
		4	Verificar que la contraseña concuerde con la contraseña almacenada en el sistema
		5	Iniciar Sesión en el sistema
		6	Finalizar
12.	Flujo secundario:		
13.	Excepciones:	Pasos	Acción
		1	El usuario no se ha registrado previamente a esto, por lo que el sistema mostrará un mensaje por pantalla: "La cuenta ingresada no existe, por favor registre esta cuenta o ingrese una cuenta existente"
		2	La contraseña ingresada no corresponde con la contraseña asociada al correo electrónico ingresado, por lo que el sistema deberá mostrar un mensaje por pantalla: "La contraseña ingresada es incorrecta, por favor verifique nuevamente su contraseña"
		3	Los campos del formulario no están llenos, por lo que mostrará un mensaje por pantalla: "Por favor llene todos los campos para iniciar sesión en el sistema"
		4	El usuario presenta problemas con su conexión a Internet, por lo que no puede realizar el envío de los datos para el proceso de Inicio de Sesión, mostrando un mensaje en pantalla: "Error al iniciar sesión, verifique que cuente con una conexión a internet"
14.	Prioridad:	Necesario	

I - Student			
1.		Proyecto:	Aplicación web para Docentes
2.	Caso de uso:	CU03	Cerrar Sesión
3.	Destinatario:	Desarrolladores	
4.	Descripción:	Este caso de uso representa al docente cerrar sesión en el sistema	
5.	Autores:	Irving R. Conde Marín	
6.	Actores:	Docente	
7.	Requerimientos:	Cerrar Sesión	
8.	Verificado por:		
9.	Precondiciones:	Tener la sesión iniciada	
10.	Postcondiciones:	El sistema deberá redirigir al usuario al apartado de inicio de sesión	
11.	Flujo principal:	Paso	Acción
		1	Cerrar la sesión actual
		2	Redirigir al usuario al apartado de inicio de sesión
		3	Finalizar
12.	Flujo secundario:		
13.	Excepciones:	Pasos	Acción
		1	El usuario presenta problemas con su conexión a Internet, por lo que no puede realizar el proceso de Cierre de Sesión, mostrando un mensaje en pantalla: "Error al cerrar sesión, verifique que cuente con una conexión a internet"
14.	Prioridad:	Necesario	

I - Student			
1.		Proyecto:	Aplicación web para Docentes
2.	Caso de uso:	CU04	Visualizar materias
3.	Destinatario:	Desarrolladores	
4.	Descripción:	Este caso de uso representa al docente visualizar sus materias correspondientes al iniciar el sistema	
5.	Autores:	Irving R. Conde Marín	
6.	Actores:	Docente	
7.	Requerimientos:	Materias del docente	
8.	Verificado por:		
9.	Precondiciones:	El docente deberá iniciar sesión en el sistema	
10.	Postcondiciones:	Ninguna	
11.	Flujo principal:	Paso	Acción
		1	Extraer el número del personal de la sesión iniciada
		2	Mandar el número del personal al servidor para iniciar una consulta
		4	Extraer la información recibida por el servidor para su uso
		5	Generar una serie de botones con los datos extraídos de cada materia
		6	Mostrar los botones en la página, además de habilitar dichos botones para su posterior selección
		7	Finalizar
12.	Flujo secundario:	Paso	Acción
		1	El servidor realizará una consulta a la base de datos para extraer las materias que estén relacionadas al docente a través de su número del personal
		2	El servidor mandará la información recopilada al sistema para su posterior uso, la cual será: Nombre de la experiencia educativa, grupo impartido, NRC de la materia
13.	Excepciones:	Pasos	Acción
		1	El docente no cuenta con experiencias educativas para impartir, por lo que el sistema deberá mostrar un mensaje por pantalla: "No hay materias por mostrar, por favor solicite a su facultad la asignación de materias para impartir"
		2	Una materia del docente no se encuentra en el sistema, por lo que el docente deberá solicitar el alta de la materia a través de la opción "Alta Materia"
		3	Una materia que no pertenece al docente se muestra en el sistema, o el docente ya no será el encargado de impartir dicha materia, por lo que el docente deberá solicitar la baja de materia a través de la opción "Baja Materia"
14.	Prioridad:	Necesario	

I - Student			
1.		Proyecto:	Aplicación web para Docentes
2.	Caso de uso:	CU05	Solicitar alta de materia
3.	Destinatario:	Desarrolladores	
4.	Descripción:	Este caso de uso representa al docente solicitando al sistema un alta de materia	
5.	Autores:	Irving R. Conde Marín	
6.	Actores:	Docente	
7.	Requerimientos:	Alta Materia	
8.	Verificado por:		
9.	Precondiciones:	El docente deberá iniciar sesión en el sistema El docente deberá contar con el NRC de la materia que desea dar de alta El docente no deberá estar asociado a la materia que desea dar de alta	
10.	Postcondiciones:	Ninguna	
11.	Flujo principal:	Paso	Acción
		1	Seleccionar el botón de “Alta Materia”
		2	Extraer el NRC de la materia solicitada para su alta
		4	Extraer el número del personal de la sesión iniciada
		5	Mandar los datos recopilados al servidor para su proceso correspondiente
		6	Mostrar un mensaje por pantalla sobre la respuesta proveniente del servidor
		7	Finalizar
12.	Flujo secundario:	Paso	Acción
		1	El servidor deberá consultar a la base de datos para validar que el número del personal existe realmente
		2	El servidor deberá consultar a la base de datos para validar que el NRC de la materia solicitada existe realmente
		3	Se ejecutará el procedimiento almacenado correspondiente de la base de datos para la asignación de la materia solicitada al docente
		4	Si la solicitud de alta se acepta, la materia deberá mostrarse en el apartado de materias del docente, además se enviará un mensaje del servidor a la aplicación para su visualización, el cual dirá: “Su solicitud de alta de materia se ha aprobado correctamente”
		5	Si la solicitud de alta se rechaza, la materia no se mostrará en el apartado de materias del docente, además, se enviará un mensaje del servidor a la aplicación para su visualización, el cual dirá: “Su solicitud de alta de materia se ha rechazado”
13.	Excepciones:	Pasos	Acción
		1	El NRC ingresado de la materia solicitada no coincide con ninguna materia registrada en el sistema, por lo que se mostrará un mensaje por pantalla: “El NRC ingresado no corresponde a ninguna materia registrada, favor de ingresar uno existente”
		2	El horario de la materia solicitada coincide con el horario de otra materia asociada al docente, por lo que se rechazará la petición y se mostrará un mensaje por pantalla: “El horario de la materia coincide con otra materia asociada al usuario, por lo que no es posible realizar el alta de materia correctamente”
		3	La materia ya está asociada a otro docente, por lo que se rechazará la petición y se mostrará un mensaje por pantalla: “La materia solicitada ya está asignada a otro docente”
14.	Prioridad:	Necesario	

I - Student			
1.		Proyecto:	Aplicación web para Docentes
2.	Caso de uso:	CU06	Solicitar baja de materia
3.	Destinatario:	Desarrolladores	
4.	Descripción:	Este caso de uso representa al docente solicitando al sistema una baja de materia	
5.	Autores:	Irving R. Conde Marín	
6.	Actores:	Docente	
7.	Requerimientos:	Baja Materia	
8.	Verificado por:		
9.	Precondiciones:	El docente deberá iniciar sesión en el sistema El docente deberá estar asociado a la materia que se desea dar de baja El docente deberá contar con el NRC de la materia que desea dar de baja	
10.	Postcondiciones:	Ninguna	
11.	Flujo principal:	Paso	Acción
		1	Seleccionar el botón de “Baja Materia”
		2	Extraer el NRC de la materia solicitada para su baja
		4	Extraer el número del personal de la sesión iniciada
		5	Mandar los datos recopilados al servidor para su proceso correspondiente
		6	Mostrar un mensaje por pantalla sobre la respuesta proveniente del servidor
		7	Finalizar
12.	Flujo secundario:	Paso	Acción
		1	El servidor deberá consultar a la base de datos para validar que el número del personal existe realmente
		2	El servidor deberá consultar a la base de datos para validar que el NRC de la materia solicitada existe realmente
		3	Se ejecutará el procedimiento almacenado correspondiente de la base de datos para la baja de la materia asociada al docente
		4	Si la solicitud de alta se acepta, la materia deberá dejar de mostrarse en el apartado de materias del docente, además se enviará un mensaje del servidor a la aplicación para su visualización, el cual dirá: “Su solicitud de baja de materia se ha aprobado correctamente”
		5	Si la solicitud de baja se rechaza, la materia se seguirá mostrando en el apartado de materias del docente, además, se enviará un mensaje del servidor a la aplicación para su visualización, el cual dirá: “Su solicitud de baja de materia se ha rechazado”
13.	Excepciones:	Pasos	Acción
		1	El NRC ingresado de la materia solicitada no coincide con ninguna materia registrada en el sistema, por lo que se mostrará un mensaje por pantalla: “El NRC ingresado no corresponde a ninguna materia registrada, favor de ingresar uno existente”
		2	El NRC de la materia no está asociado al docente que ha realizado la solicitud de baja de materia, por lo que se mostrará un mensaje por pantalla: “La materia que desea dar de baja no está asociada al usuario”
14.	Prioridad:	Necesario	

I - Student			
1.	(ovalo)	Proyecto:	Aplicación web para Docentes
2.	Caso de uso:	CU07	Seleccionar una opción de materia
3.	Destinatario:	Desarrolladores	
4.	Descripción:	Este caso de uso representa al docente seleccionando una opción del apartado visualizar materias	
5.	Autores:	Irving R. Conde Marín	
6.	Actores:	Docente	
7.	Requerimientos:	Seleccionar Materia	
8.	Verificado por:		
9.	Precondiciones:	El docente deberá iniciar sesión en el sistema El docente debe contar con materias para visualizar	
10.	Postcondiciones:	El Docente debe presionar la opción de la materia deseada	
11.	Flujo principal:	Paso	Acción
		1	Seleccionar una opción de materia
		2	Redirigir a la pestaña correspondiente a la vista de lista de asistencia
		3	Finalizar
12.	Flujo secundario:	Paso	Acción
13.	Excepciones:	Pasos	Acción
14.	Prioridad:	Necesario	

I - Student			
1.		Proyecto:	Aplicación web para Docentes
2.	Caso de uso:	CU08	Visualizar listas de asistencia
3.	Destinatario:	Desarrolladores	
4.	Descripción:	Este caso de uso representa al docente visualizando la lista de asistencia de la materia correspondiente	
5.	Autores:	Irving R. Conde Marín	
6.	Actores:	Docente	
7.	Requerimientos:	Tabla de asistencia	
8.	Verificado por:		
9.	Precondiciones:	La materia debe contar con una lista de asistencia almacenada	
10.	Postcondiciones:	Ninguna	
11.	Flujo principal:	Paso	Acción
		1	Se deberá cargar la pestaña correspondiente a la vista de lista de asistencia
		2	Se deberá verificar si el sistema cuenta con una lista de asistencia almacenada
		3	El sistema deberá trazar gráficamente el formato de la tabla de asistencia en su apartado correspondiente
		4	Se deberán cargar los datos obtenidos de la lista almacenada en la tabla graficada por el sistema
		5	Se deberán habilitar los botones para la modificación de la lista de asistencia por parte del docente
		6	Se deberá mostrar un apartado para la consulta de un alumno en específico del lado derecho
		7	Se deberán recibir los datos enviados por el servidor respecto a la toma de asistencia automática de los alumnos y reflejar inmediatamente en la tabla graficada por el sistema
		8	Finalizar
12.	Flujo secundario:	Paso	Acción
		1	El sistema de lectura de QR deberá escanear los códigos de los estudiantes para la toma de asistencia
		2	El sistema de lectura de QR deberá extraer la información de los códigos QR escaneados y mandar la información necesaria al servidor para este proceso
		3	El servidor deberá realizar la validación de que el alumno está registrado en la experiencia educativa en la que se tomará su asistencia
		4	El servidor deberá realizar los cambios necesarios sobre la lista de asistencia correspondiente al proceso
13.	Excepciones:	Pasos	Acción
		1	No existe una lista de asistencia almacenada en el sistema, por lo que la lista graficada estará sin datos y aparecerá un mensaje sobre pantalla: "La lista de asistencia no cuenta con los datos necesarios para realizar el proceso de asistencia"
14.	Prioridad:	Necesario	

I - Student			
1.		Proyecto:	Aplicación web para Docentes
2.	Caso de uso:	CU09	Subir archivo .XLS
3.	Destinatario:	Desarrolladores	
4.	Descripción:	Este caso de uso representa al docente subiendo una lista de asistencia al sistema para su posterior clonación y almacenado de este	
5.	Autores:	Irving R. Conde Marín	
6.	Actores:	Docente	
7.	Requerimientos:	Clonado de archivo	
8.	Verificado por:		
9.	Precondiciones:	El docente deberá cargar su archivo .XLS correspondiente a la lista de asistencia al sistema	
10.	Postcondiciones:	El docente deberá presionar el botón de guardar para finalizar el proceso	
11.	Flujo principal:	Paso	Acción
		1	Se deberá cargar la pestaña correspondiente a la vista de lista de asistencia
		2	El docente deberá presionar el botón “Subir archivo”
		3	El sistema deberá verificar que se recibe un archivo .XLS
		4	Se deberá verificar que el archivo escaneado cumpla con el formato estipulado para una lista de asistencia correcta
		5	Se deberá guardar el archivo .XLS cargado por el usuario en el sistema
		6	Se mostrará una vista previa de cómo luce el archivo escaneado y se le solicitará al usuario presionar el botón “Guardar”
		7	Se mostrará un mensaje en pantalla: “El archivo ha sido almacenado correctamente en el sistema para su futuro uso”
		8	Finalizar
12.	Flujo secundario:	Paso	Acción
		1	El sistema deberá escanear el archivo cargado por el usuario para su futuro almacenado.
		2	El sistema deberá clonar el archivo .XLS en un nuevo archivo .XLS para el funcionamiento del sistema
13.	Excepciones:	Pasos	Acción
		1	El archivo cargado por el usuario no se encuentra en formato .XLS, mostrando un mensaje por pantalla: “El archivo no se encuentra en formato .XLS, por favor suba un archivo con el formato correcto”
		2	El archivo escaneado no cumple con el formato predefinido al de una lista de asistencia, mostrando un mensaje por pantalla: “El contenido del archivo no corresponde con el formato de una lista de asistencia, verifique que el documento sea el correcto”
14.	Prioridad:	Necesario	

2.4.4.2 Aplicación web para Alumnos

I - Student			
1.		Proyecto:	Aplicación web para Alumnos
2.	Caso de uso:	CU01	Registro de cuenta
3.	Destinatario:	Desarrolladores	
4.	Descripción:	Este caso de uso representa al alumno realizar su registro de cuenta en el sistema	
5.	Autores:	Irving R. Conde Marín	
6.	Actores:	Alumnos	
7.	Requerimientos:	Registro de alumno	
8.	Verificado por:		
9.	Precondiciones:	Contar con su matrícula escolar	
10.	Postcondiciones:	Ninguna	
11.	Flujo principal:	Paso	Acción
		1	Verificar que todos los campos estén llenados por el usuario
		2	Mandar los datos para el proceso de registro
		3	Verificar que los datos cumplen con las restricciones establecidas en los requerimientos
		4	Verificar que la matrícula del alumno no ha sido registrado con anterioridad en el sistema
		5	Guardar datos de la cuenta
		6	Mostrar mensaje: "La cuenta se ha registrado exitosamente"
		7	Finalizar
12.	Flujo secundario:		
13.	Excepciones:	Pasos	Acción
		1	El usuario ya se ha registrado con anterioridad, por lo cual se mostrará un mensaje por pantalla: "El usuario ya se ha registrado anteriormente, inicia sesión con tu correo electrónico ingresado"
		2	El usuario no ha llenado todos los campos del registro, por lo cual el sistema mostrará un mensaje por pantalla: "Por favor llene todos los campos para realizar el proceso de registro"
		3	La matrícula ingresada en el campo no cumple con el formato estipulado en los requerimientos del sistema, por lo cual el sistema mostrará un mensaje por pantalla: "El formato de la matrícula escolar ingresada no corresponde con el estipulado"
		4	La contraseña ingresada en el campo no cumple con el formato estipulado en los requerimientos del sistema, por lo cual mostrará un mensaje por pantalla: "La contraseña no coincide con las restricciones dadas, debe contener al menos 8 caracteres, entre ellos 1 mayúscula, 1 número y un carácter especial"
		5	El peso de la imagen a subir es mayor al permitido por el sistema, por lo que se mostrará un mensaje por pantalla: "No se ha podido realizar el registro debido a que el peso de la imagen es mayor al permitido"
		6	Si el usuario presenta problemas de conexión a internet, y esto provoca que no puedan mandarse los datos para su registro, se mostrará un mensaje por pantalla: "No se ha podido realizar el proceso de registro, verifique que tenga una conexión a internet"
		7	El correo electrónico ingresado en el campo no existe, por lo que el registro no podrá llevarse a cabo, mostrando un mensaje en pantalla: "Ingrese un correo electrónico existente"
		8	La matrícula escolar no coincide con ninguna almacenado en la base de datos, mostrando un mensaje en pantalla: "La matrícula escolar ingresada no corresponde a ningún dato almacenado, por favor ingrese un dato válido"
14.	Prioridad:	Necesario	

I - Student			
1.		Proyecto:	Aplicación web para Alumnos
2.	Caso de uso:	CU02	Inicio de Sesión
3.	Destinatario:	Desarrolladores	
4.	Descripción:	Este caso de uso representa al alumno iniciar sesión en el sistema con su cuenta registrada	
5.	Autores:	Irving R. Conde Marín	
6.	Actores:	Alumnos	
7.	Requerimientos:	Login de Alumnos	
8.	Verificado por:		
9.	Precondiciones:	Contar con una cuenta registrada en el sistema	
10.	Postcondiciones:	Ninguna	
11.	Flujo principal:	Paso	Acción
		1	Verificar que todos los campos estén llenados por el usuario
		2	Mandar los datos para el proceso de inicio de sesión
		3	Verificar que el correo electrónico ingresado esté registrado en el sistema
		4	Verificar que la contraseña concuerde con la contraseña almacenada en el sistema
		5	Iniciar Sesión en el sistema
		6	Finalizar
12.	Flujo secundario:		
13.	Excepciones:	Pasos	Acción
		1	El usuario no se ha registrado previamente a esto, por lo que el sistema mostrará un mensaje por pantalla: "La cuenta ingresada no existe, por favor registre esta cuenta o ingrese una cuenta existente"
		2	La contraseña ingresada no corresponde con la contraseña asociada al correo electrónico ingresado, por lo que el sistema deberá mostrar un mensaje por pantalla: "La contraseña ingresada es incorrecta, por favor verifique nuevamente su contraseña"
		3	Los campos del formulario no están llenos, por lo que mostrará un mensaje por pantalla: "Por favor llene todos los campos para iniciar sesión en el sistema"
		4	El usuario presenta problemas con su conexión a Internet, por lo que no puede realizar el envío de los datos para el proceso de Inicio de Sesión, mostrando un mensaje en pantalla: "Error al iniciar sesión, verifique que cuente con una conexión a internet"
14.	Prioridad:	Necesario	

I - Student			
1.		Proyecto:	Aplicación web para Alumnos
2.	Caso de uso:	CU03	Cerrar Sesión
3.	Destinatario:	Desarrolladores	
4.	Descripción:	Este caso de uso representa al alumno cerrar sesión en el sistema	
5.	Autores:	Irving R. Conde Marín	
6.	Actores:	Alumnos	
7.	Requerimientos:	Cerrar Sesión	
8.	Verificado por:		
9.	Precondiciones:	Tener la sesión iniciada	
10.	Postcondiciones:	El sistema deberá redirigir al usuario al apartado de inicio de sesión	
11.	Flujo principal:	Paso	Acción
		1	Cerrar la sesión actual
		2	Redirigir al usuario al apartado de inicio de sesión
		3	Finalizar
12.	Flujo secundario:		
13.	Excepciones:	Pasos	Acción
		1	El usuario presenta problemas con su conexión a Internet, por lo que no puede realizar el proceso de Cierre de Sesión, mostrando un mensaje en pantalla: "Error al cerrar sesión, verifique que cuente con una conexión a internet"
14.	Prioridad:	Necesario	

I - Student			
1.		Proyecto:	Aplicación web para Alumno
2.	Caso de uso:	CU04	Visualizar información del usuario
3.	Destinatario:	Desarrolladores	
4.	Descripción:	Este caso de uso representa al alumno visualizando su información registrada en el sistema	
5.	Autores:	Irving R. Conde Marín	
6.	Actores:	Alumnos	
7.	Requerimientos:	Menú de inicio	
8.	Verificado por:		
9.	Precondiciones:	El alumno debe tener la sesión iniciada en el sistema	
10.	Postcondiciones:	Ninguna	
11.	Flujo principal:	Paso	Acción
		1	El alumno deberá ingresar al sistema
		2	El sistema deberá extraer la matrícula de la sesión inicial
		3	El sistema enviará la matrícula recopilada al servidor para la consulta de los datos sobre el alumno
		4	El sistema recibirá los datos enviados por el servidor sobre el alumno
		5	Se deberán mostrar los datos obtenidos para la visualización del alumno
		6	Finalizar
12.	Flujo secundario:	Paso	Acción
		1	El servidor recibirá la matrícula del alumno que desea ser consultado
		2	El servidor recabará la información del alumno
		3	Se enviará la información obtenida al sistema para su visualización del alumno
		4	
13.	Excepciones:	Pasos	Acción
		1	La información del alumno no es correcta, por lo que el alumno podrá solicitar un cambio de información al sistema para una corrección de datos
14.	Prioridad:	Necesario	

I - Student																		
1.	Proyecto:	Aplicación web para Alumno																
2.	Caso de uso:	CU06 Generar código QR																
3.	Destinatario:	Desarrolladores																
4.	Descripción:	Este caso de uso representa al alumno generando un código QR																
5.	Autores:	Irving R. Conde Marín																
6.	Actores:	Alumnos																
7.	Requerimientos:	Generar código QR																
8.	Verificado por:																	
9.	Precondiciones:	El alumno debe tener la sesión iniciada en el sistema																
10.	Postcondiciones:	Ninguna																
11.	Flujo principal:	<table border="1"> <thead> <tr> <th>Paso</th><th>Acción</th></tr> </thead> <tbody> <tr> <td>1</td><td>El alumno deberá ingresar al sistema</td></tr> <tr> <td>2</td><td>El alumno deberá seleccionar el apartado de generar código QR</td></tr> <tr> <td>3</td><td>El sistema deberá extraer la matrícula de la sesión iniciada</td></tr> <tr> <td>4</td><td>El sistema enviará la matriculada al servidor para la consulta de información del alumno</td></tr> <tr> <td>5</td><td>El sistema recibirá la información enviada por el servidor sobre el alumno</td></tr> <tr> <td>6</td><td>Se deberá generar un código QR con la información recopilada del alumno</td></tr> <tr> <td>7</td><td>Finalizar</td></tr> </tbody> </table>	Paso	Acción	1	El alumno deberá ingresar al sistema	2	El alumno deberá seleccionar el apartado de generar código QR	3	El sistema deberá extraer la matrícula de la sesión iniciada	4	El sistema enviará la matriculada al servidor para la consulta de información del alumno	5	El sistema recibirá la información enviada por el servidor sobre el alumno	6	Se deberá generar un código QR con la información recopilada del alumno	7	Finalizar
Paso	Acción																	
1	El alumno deberá ingresar al sistema																	
2	El alumno deberá seleccionar el apartado de generar código QR																	
3	El sistema deberá extraer la matrícula de la sesión iniciada																	
4	El sistema enviará la matriculada al servidor para la consulta de información del alumno																	
5	El sistema recibirá la información enviada por el servidor sobre el alumno																	
6	Se deberá generar un código QR con la información recopilada del alumno																	
7	Finalizar																	
12.	Flujo secundario:	<table border="1"> <thead> <tr> <th>Paso</th><th>Acción</th></tr> </thead> <tbody> <tr> <td>1</td><td>El servidor recibirá la matrícula del alumno</td></tr> <tr> <td>2</td><td>El servidor recabará la información acerca del alumno con la matrícula recibida por el sistema</td></tr> <tr> <td>3</td><td>Se enviarán los datos recopilados al sistema para su uso</td></tr> </tbody> </table>	Paso	Acción	1	El servidor recibirá la matrícula del alumno	2	El servidor recabará la información acerca del alumno con la matrícula recibida por el sistema	3	Se enviarán los datos recopilados al sistema para su uso								
Paso	Acción																	
1	El servidor recibirá la matrícula del alumno																	
2	El servidor recabará la información acerca del alumno con la matrícula recibida por el sistema																	
3	Se enviarán los datos recopilados al sistema para su uso																	
13.	Excepciones:	<table border="1"> <thead> <tr> <th>Pasos</th><th>Acción</th></tr> </thead> <tbody> <tr> <td></td><td></td></tr> <tr> <td></td><td></td></tr> </tbody> </table>	Pasos	Acción														
Pasos	Acción																	
14.	Prioridad:	Necesario																

2.4.4.2 Aplicación web para simular escáner

I - Student														
1.	Proyecto:	Sistema para lectura de códigos QR												
2.	Caso de uso:	CU01												
3.	Destinatario:	Desarrolladores												
4.	Descripción:	Este caso de uso representa al usuario encendiendo el equipo												
5.	Autores:	Irving R. Conde Marín												
6.	Actores:	Usuarios												
7.	Requerimientos:	Encendido del equipo												
8.	Verificado por:													
9.	Precondiciones:	El equipo debe estar conectado a una corriente de luz												
10.	Postcondiciones:	Ninguna												
11.	Flujo principal:	<table border="1"> <thead> <tr> <th>Paso</th><th>Acción</th></tr> </thead> <tbody> <tr> <td>1</td><td>El usuario deberá conectar el equipo a una corriente de luz</td></tr> <tr> <td>2</td><td>El equipo deberá mostrar un ícono de botón de encendido</td></tr> <tr> <td>3</td><td>El usuario deberá apretar el botón de encendido para iniciar el sistema</td></tr> <tr> <td>4</td><td>El equipo se encenderá y mostrará el menú principal</td></tr> <tr> <td>5</td><td>Finalizar</td></tr> </tbody> </table>	Paso	Acción	1	El usuario deberá conectar el equipo a una corriente de luz	2	El equipo deberá mostrar un ícono de botón de encendido	3	El usuario deberá apretar el botón de encendido para iniciar el sistema	4	El equipo se encenderá y mostrará el menú principal	5	Finalizar
Paso	Acción													
1	El usuario deberá conectar el equipo a una corriente de luz													
2	El equipo deberá mostrar un ícono de botón de encendido													
3	El usuario deberá apretar el botón de encendido para iniciar el sistema													
4	El equipo se encenderá y mostrará el menú principal													
5	Finalizar													
12.	Flujo secundario:													
13.	Excepciones:	<table border="1"> <thead> <tr> <th>Pasos</th><th>Acción</th></tr> </thead> <tbody> <tr> <td>1</td><td>El equipo no está conectado a una corriente de luz, por lo que no permitirá iniciar el sistema</td></tr> </tbody> </table>	Pasos	Acción	1	El equipo no está conectado a una corriente de luz, por lo que no permitirá iniciar el sistema								
Pasos	Acción													
1	El equipo no está conectado a una corriente de luz, por lo que no permitirá iniciar el sistema													
14.	Prioridad:	Necesario												

I - Student										
1.	Proyecto:	Sistema para lectura de códigos QR								
2.	Caso de uso:	CU02								
3.	Destinatario:	Desarrolladores								
4.	Descripción:	Este caso de uso representa al usuario apagando el equipo								
5.	Autores:	Irving R. Conde Marín								
6.	Actores:	Usuarios								
7.	Requerimientos:	Apagado del equipo								
8.	Verificado por:									
9.	Precondiciones:	El equipo debe estar conectado a una corriente de luz El equipo debe estar encendido								
10.	Postcondiciones:	Ninguna								
11.	Flujo principal:	<table border="1"> <thead> <tr> <th>Paso</th><th>Acción</th></tr> </thead> <tbody> <tr> <td>1</td><td>El usuario debe apretar el botón de apagado</td></tr> <tr> <td>2</td><td>El equipo se apagará correctamente</td></tr> <tr> <td>3</td><td>Finalizar</td></tr> </tbody> </table>	Paso	Acción	1	El usuario debe apretar el botón de apagado	2	El equipo se apagará correctamente	3	Finalizar
Paso	Acción									
1	El usuario debe apretar el botón de apagado									
2	El equipo se apagará correctamente									
3	Finalizar									
12.	Flujo secundario:									
13.	Excepciones:									
14.	Prioridad:	Necesario								

I - Student			
1.		Proyecto:	Sistema para lectura de códigos QR
2.	Caso de uso:	CU03	Configuración del equipo
3.	Destinatario:	Desarrolladores	
4.	Descripción:	Este caso de uso representa al usuario configurando el equipo	
5.	Autores:	Irving R. Conde Marín	
6.	Actores:	Usuarios	
7.	Requerimientos:	Configuración del equipo	
8.	Verificado por:		
9.	Precondiciones:	El equipo debe estar conectado a una corriente de luz El equipo debe estar encendido	
10.	Postcondiciones:	Ninguna	
11.	Flujo principal:	Paso	Acción
		1	El usuario deberá ingresar al apartado de configuración del equipo
		2	El usuario deberá llenar los campos necesarios con la información adecuada para un correcto funcionamiento del sistema
		3	Se deberá presionar el botón de aceptar para guardar la configuración
		4	El equipo iniciará el proceso de configuración correspondiente, solicitando la información necesaria al servidor para su correcto funcionamiento
		5	Finalizar
12.	Flujo secundario:	Pasos	Acción
		1	El equipo enviará los datos ingresados por el usuario para realizar una consulta de materias al servidor
		2	El servidor recibirá los datos enviados por el equipo
		3	El servidor realizará una consulta a la base de datos para recabar las materias que se imparten en el edificio y salón en donde está encontrado el equipo
		4	El servidor enviará la información recopilada al equipo
		5	El equipo recibirá la información enviada por el servidor y se configurará para modificar las listas de asistencia del docente correspondiente
13.	Excepciones:	Pasos	Acción
		1	El salón ingresado al equipo no cuenta con materias cargadas para el proceso de toma de asistencia
14.	Prioridad:	Necesario	

I - Student			
1.		Proyecto:	Sistema para lectura de códigos QR
2.	Caso de uso:	CU04	Visualizar Opciones
3.	Destinatario:	Desarrolladores	
4.	Descripción:	Este caso de uso representa al usuario visualizando las opciones disponibles por el equipo	
5.	Autores:	Irving R. Conde Marín	
6.	Actores:	Usuarios	
7.	Requerimientos:	Opciones Disponibles	
8.	Verificado por:		
9.	Precondiciones:	El equipo debe estar encendido	
10.	Postcondiciones:	Ninguna	
11.	Flujo principal:	Paso	Acción
		1	Al iniciar el equipo deberá presentar al usuario las opciones que están disponibles para su uso
		2	El usuario deberá seleccionar alguna opción presentada por el equipo
		3	El equipo deberá ejecutar la opción seleccionada por el usuario
		4	Finalizar
12.	Flujo secundario:	Pasos	Acción
13.	Excepciones:	Pasos	Acción
14.	Prioridad:	Necesario	

I - Student			
1.		Proyecto:	Sistema para lectura de códigos QR
2.	Caso de uso:	CU05	Lectura de código QR
3.	Destinatario:	Desarrolladores	
4.	Descripción:	Este caso de uso representa al usuario escogiendo la opción de lectura de códigos QR	
5.	Autores:	Irving R. Conde Marín	
6.	Actores:	Usuarios	
7.	Requerimientos:	Lectura de código QR	
8.	Verificado por:		
9.	Precondiciones:	El usuario debe seleccionar la opción de Lectura de código QR	
10.	Postcondiciones:	El equipo debe configurarse para empezar el escaneo de códigos QR	
11.	Flujo principal:	Paso	Acción
		1	El usuario selecciona la opción de leer códigos QR
		2	El equipo ejecuta su función para leer códigos QR
		3	El equipo escanea un código QR generado por un alumno
		4	Se deberá extraer la información contenida en el código previamente escaneado
		5	El sistema enviará al servidor la matrícula del alumno para la toma de asistencia, junto con la hora en que se escaneó el código
		6	Si la toma de asistencia fue correcta, el sistema mostrará en el equipo la información del alumno, junto a un mensaje: "Asistencia tomada"
		7	Si la toma de asistencia fue incorrecta, el sistema mostrará en el equipo un mensaje en pantalla proveniente del servidor
		8	El equipo deberá almacenar el registro de lectura en su historial
		9	Finalizar
12.	Flujo secundario:	Pasos	Acción
		1	El servidor recibirá la matrícula enviada por el equipo para el proceso de toma de asistencia y la hora en que sucedió el escaneo
		2	El servidor ejecutará el procedimiento almacenado en la base de datos para la toma de asistencia, enviando como parámetro la matrícula del alumno y la hora de escaneo
		3	La base de datos deberá validar que el alumno está inscrito en la materia correspondiente al horario en que fue tomado su registro
		4	El servidor deberá realizar el cambio en la lista de asistencia correspondiente al horario
		5	La lista de asistencia del docente deberá recibir los cambios y mostrarlos para su visualización
13.	Excepciones:	Pasos	Acción
		1	El alumno no está registrado en la materia, por lo que se enviará un mensaje al sistema: "El alumno no está registrado en la materia"
		2	No se ha podido leer correctamente el código QR, por lo que no se realizará el proceso de toma de asistencia y no mostrará la información del estudiante
		3	El sistema no puede realizar el proceso de toma de asistencia debido a problemas de internet, mostrando un mensaje por pantalla
14.	Prioridad:	Necesario	

I - Student			
1.	(ovalo)	Proyecto:	Sistema para lectura de códigos QR
2.	Caso de uso:	CU06	Historial de lectura de códigos QR
3.	Destinatario:	Desarrolladores	
4.	Descripción:	Este caso de uso representa al usuario escogiendo la opción de mostrar el historial de lectura de códigos QR	
5.	Autores:	Irving R. Conde Marín	
6.	Actores:	Usuarios	
7.	Requerimientos:	Historial de escaneo de códigos QR	
8.	Verificado por:		
9.	Precondiciones:	El usuario debe seleccionar la opción de historial de lectura de códigos QR	
10.	Postcondiciones:	El equipo debe mostrar el historial de lectura	
11.	Flujo principal:	Paso	Acción
		1	El usuario selecciona la opción de historial de lectura de códigos QR
		2	El equipo ejecuta su función para mostrar el historial de lectura
		3	El equipo mostrará la información sobre el historial de lectura
		4	Finalizar
12.	Flujo secundario:		
13.	Excepciones:	Pasos	Acción
		1	No se han tomado lecturas de códigos QE, por lo que se mostrará un mensaje en pantalla: "No se han tomado lecturas de códigos QR el día de hoy, favor de iniciar la lectura de códigos"
		2	Los códigos QR solo se almacenarán por un periodo temporal, por lo que no se podrán consultar registros que pasen de este límite
14.	Prioridad:	Necesario	

2.6.3 Mockups

Mockups Docentes

Mockup de registro de cuenta

En este mockup tendremos la libertad de poder tener nuestra propia cuenta para poder acceder al sitio web, tendremos que ingresar nuestros datos como tenemos en la imagen: Correo institucional, contraseña, tendremos que confirmar nuestra contraseña, tendremos otro apartado para poder seleccionar una imagen para nuestro usuario.

- Como siguiente daremos en crear cuenta y automáticamente tendremos nuestra cuenta creada.



The mockup shows a registration form titled "Registrar Cuenta" with the sub-instruction "Ingresa los datos solicitados". It includes fields for "@Correo", "Contraseña", "Confirmar Contraseña", and a file upload section showing "0" files selected. Below the file upload is a button "Elegir archivo" and a message "No se eligió ningún archivo". At the bottom are two buttons: "Crear Cuenta" and "Iniciar Sesión!" with a user icon.

Mockup de login

Aquí tenemos nuestro mockup login, el cual tiene como propósito verificar que tengamos una cuenta creada, “si no tenemos una cuenta, tenemos que crear una”.

Tendremos que ingresar los datos que ingresamos a la hora de crear nuestra cuenta en los siguientes apartados: Usuario, contraseña. Y seleccionaremos en iniciar sesión.

Si en dado caso no recordamos nuestra contraseña tendremos un apartado en donde nos especifica “Recuperar contraseña”, el cual nos ayudara a recuperar nuestra contraseña o modificarla.

Tendremos otro apartado que nos indica “Crear cuenta”, crearemos una en caso de no tener ninguna.



Mockup Cargar Lista de Asistencia.

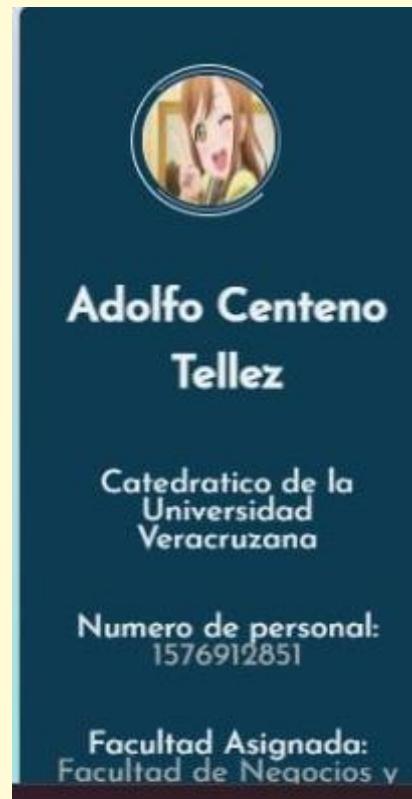
El propósito de este mockup es poder seleccionar una lista que tengamos en nuestro equipo de cómputo, celular, etc.

Al tener el archivo seleccionado daremos en el botón “Cargar archivo”, podremos ver nuestra lista en la página.

The mockup displays a user interface for managing attendance lists. At the top, there is a navigation bar with the Faculty of Engineering logo, the text "FACULTAD DE INGENIERÍA", and links for "Inicio", "Listado", "Alta Materia", "Baja Materia", and "Cerrar Sesión". Below the navigation bar, there is a file selection area with buttons for "Seleccionar archivo", "Sin archivos seleccionados", and "Cargar archivo". A large blue card on the left side contains a profile picture of a person, the name "Adolfo Centeno Tellez", the title "Catedrático de la Universidad Veracruzana", and the personal number "Número de personal: 1576912851". The background of the page is light yellow.

Mockup de perfil de docente

En este apartado se mostrará los datos del docente una tarjeta y también se mostrará como tiene que ser el diseño



Mockup del menú del docente (apartado/inicio)

Aquí tenemos el mockup inicio, que una vez que hayamos iniciado sesión en el mockup “Login”, podremos tener acceso a la página.

Aquí podremos tener acceso a cualquier tipo de información ya sea del docente, alumno, materias, grupo, etc.

Ya dicho lo anterior, aquí podremos ver nuestro grupo y las materias seleccionadas de dichos grupos.

Tendremos diferentes botones: Inicio, Listado, Alta Materia, Baja Materia, Cerrar sesión.

Como podemos observar tenemos 3 botones muy peculiares los cuales son: Listado, Alta Materia, Baja Materia, de los cuales hablaremos más adelante.

The mockup displays the following information:

- Header:** FACULTAD DE INGENIERÍA, Inicio, Listado, Alta Materia, Baja Materia, Cerrar Sesión.
- Profile:** Adolfo Centeno Tellez, Catedrático de la Universidad Veracruzana, Número de personal: 1576912651, Facultad Asignada: Facultad de Negocios y Tecnologías.
- Materias del docente:**
 - Sistemas en Red**: NRC: 98429, Grupo: 601, Estudiantes: 4. [Lista Asistencia](#). Description: Experiencia Educativa perteneciente al programa educativo de ISW en la Facultad de negocios y tecnologías.
 - Desarrollo de Software**: NRC: 98443, Grupo: 601, Estudiantes: 1. [Lista Asistencia](#). Description: Experiencia Educativa perteneciente al programa educativo de ISW en la Facultad de negocios y tecnologías.
 - Principios de Construcción de Software**: NRC: 91235, Grupo: 401, Estudiantes: 0. [Lista Asistencia](#). Description: Experiencia Educativa perteneciente al programa educativo de ISW en la Facultad de negocios y tecnologías.

Mockup del menú del docente (apartado/listado)

Esta sección como su nombre especifica tiene como propósito mostrar las asistencias en tiempo real de los alumnos.

Tenemos los datos del alumno y la hora los cuales son los siguientes: ID, Matricula, Nombre, Estado, Hora.

FACULTAD DE INGENIERÍA				
Inicio	Listado	Alta Materia	Baja Materia	Cerrar Sesión
ID	Matricula	Nombre	Estado	Hora

Mockup del menú del docente (apartado/alta materia)

Aquí tenemos el botón Alta de materia.

Su propósito es poder dar de alta una materia que necesitemos, que queremos decir con esto, por algún motivo de algún fallo, tiempo, fallo técnico, etc. No pudo correctamente solicitar todas sus materias, esta sección le ayudara para poder seleccionar dicha materia restante.

Como podemos observar tenemos varios datos que nos muestra: Grupo y Escuela, Número telefónico, Correo escolar.

Tenemos 2 campos el cual hay que llenar con los datos de dicha materia que solicitemos.

Al terminar si la información es correcta se activará el botón “Enviar”.

Solicitar alta de materia!

Contactanos



602 ISW
USBI



272-259-4250



zS20006735@estudiantes.uv.

Número de personal

NRC de materia

Enviar

Mockup del menú del docente (apartado/baja materia)

Aquí tenemos el botón Baja de Materia.

Su propósito es dar de baja una materia que no solicito o que simplemente no está dispuesto a cursar por algún inconveniente.

Como podemos observar tenemos varios datos que nos muestra: Grupo y Escuela, Número telefónico, Correo escolar.

Tenemos 2 campos el cual hay que llenar con los datos de dicha materia.

Al terminar si la información es correcta se activará el botón “Enviar”.

Solicitar baja de materia!

Contactanos

 602 ISW
USBI

 272-259-4250

 zS20006735@estudiantes.uv.r

Número de personal

NRC de materia

Enviar

Mockup del botón lista de asistencia

Aquí tenemos el botón Listado el cual nos muestra la lista de asistencia del grupo.

Esta lista nos muestra los datos de los alumnos: Matricula, Alumno, Carrera, Status.

Lista de Asistencia

Índice	Alumno	Matricula	Carrera	Status
1	zS20006712	Magdyel Aguilar Cid	ISW	I era inscripción
2	zS20006726	Aaron Emiliano Torres Nuñez	ISW	I era Inscripción
3	zS20006735	Irving Rafael Conde Marín	ISW	I era inscripción
4	zS20006749	Adriel Eduardo Peregrina Soto	ISW	I era inscripción

Mockup del sistema de simulador

Mockups del menú de inicio del simulador

En este mockup se muestra la cámara para poder escanear y se muestran los últimos datos del último escaneo que se realizó

The mockup displays a user interface for a simulator. At the top, there is a blue header bar with the logo of the Faculty of Engineering on the left, followed by the text "FACULTAD DE INGENIERÍA". To the right of the logo are three buttons: "Inicio", "Configuracion", and a green button labeled "Encendido" (On). Below the header is a camera feed showing a room with several long tables and chairs, likely a laboratory or classroom setting. Underneath the camera feed, the text "Último dato escaneado:" is displayed, followed by three lines of data: "Matrícula:", "Nombre: No hay datos", and "Status: No hay datos".

Mockups del menú de configuración del simulador

Al seleccionar el botón configuración nos dirigirá a esta sección.

Tendremos varios botones y campos.

Tendremos 3 botones: Mostrar_Lista, Mostrar_Materias, Guardar configuración.

Y tendremos 2 campos: Edificio asignado, Selecciona un salón, los cuales deberemos llenar.



Mockups del botón mostrar lista

Si seleccionamos el botón Mostrar_Lista, podremos observar nuestra lista de asistencia previamente seleccionada.

Mostrará lista de asistencia de la materia en turno (El escáner se guiará por el Salón, Edificio y Hora. Es decir, El lunes a las 7am en el salón 6 del edificio 1 se imparte Desarrollo de software.

A lo cual arrojara la lista.

The mockup shows a user interface for managing attendance settings and viewing attendance lists. At the top, there are two buttons: "Mostrar_Lista" (highlighted in blue) and "Mostrar_Materias". Below these are dropdown menus for "Edificio asignado" (set to "Edificio 1") and "Salón asignado" (set to "Salón 6"). A "Guardar configuración" button is located below the dropdowns. The main section is titled "Lista de Asistencia" and contains a table with four rows of student information:

Indice	Alumno	Matrícula	Carrera	Status
1	S19004877	MENESES CONDE JESUS SAITH	ISOF	1ra
2	S20006726	TORRES NUÑEZ AARON EMILIANO	ISOF	1ra
3	S20006727	SANCHEZ VASQUEZ ANAHI XIMENA	ISOF	1ra
4	S20006728	JUAREZ ESPINOSA ERICK	ISOF	1ra

Mockup del front de Estudiantes

Mockup del menú de inicio del alumno

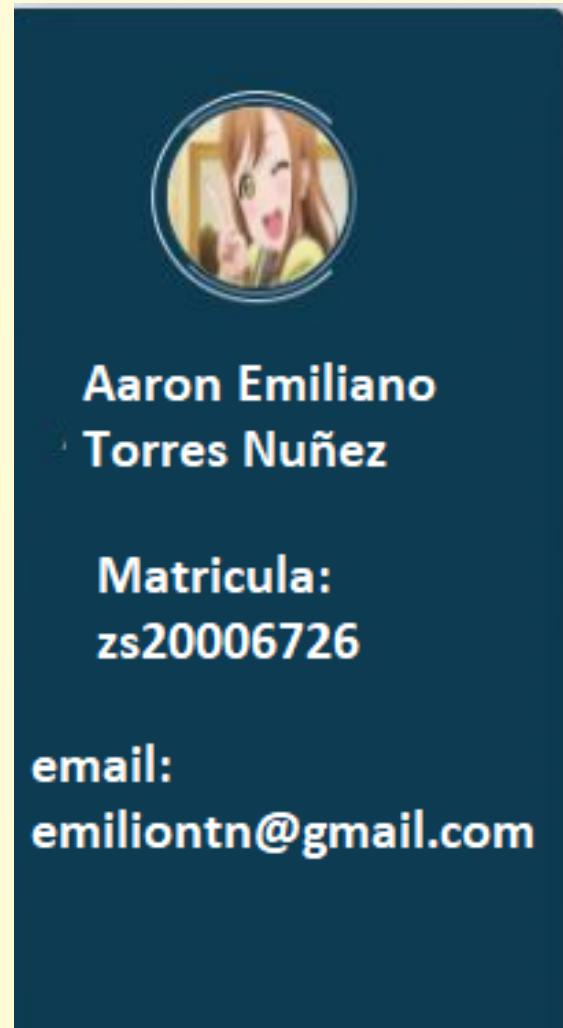
En este mockup se mostrará el diseño y lo que contiene el menú de inicio del alumno

The mockup displays a user interface for a student login. At the top, there is a blue header bar with the university's crest, the text "FACULTAD DE INGENIERÍA", and navigation links for "Inicio", "perfil", and "Cerrar Sesión". Below the header, the main content area has a light purple background. It features a dark blue sidebar on the left containing a profile picture of a student, the name "Aaron Emiliano Torres Nuñez", the matrícula number "zs20006726", and the email "emiliontn@gmail.com". To the right of the sidebar, there is a section titled "Materias del Alumno" (Student Subjects) which lists three subjects: "Sistemas en Red" (NRC: 98429), "Desarrollo de Software" (NRC: 98443), and "Principios de Construcción de Software" (NRC: 91235). Each subject entry includes a "generar qr" button and a brief description of the educational experience.

Materia	NRC	Descripción
Sistemas en Red	98429	Experiencia Educativa perteneciente al programa educativo de ISW en la Facultad de negocios y tecnologías
Desarrollo de Software	98443	Experiencia Educativa perteneciente al programa educativo de ISW en la Facultad de negocios y tecnologías
Principios de Construcción de Software	91235	Experiencia Educativa perteneciente al programa educativo de ISW en la Facultad de negocios y tecnologías

Mockup del menú de perfil del alumno

En este mockup se mostrarán los siguientes datos del alumno como perfil



Mockups del botón mostrar materia

En este mockup se muestra las materias que el salón asignado contiene, el nrc y el horario de la experiencia educativa.

Materias del Alumno

The mockup displays three cards, each representing a subject. Each card has a dark blue header with white text and a light blue footer with a blue button labeled "generar qr".

- Sistemas en Red**
NRC: 98429
generar qr
Experiencia Educativa perteneciente al programa educativo de ISW en la Facultad de negocios y tecnologías
- Desarrollo de Software**
NRC: 98443
generar qr
Experiencia Educativa perteneciente al programa educativo de ISW en la Facultad de negocios y tecnologías
- Principios de Construcción de Software**
NRC: 91235
generar qr
Experiencia Educativa perteneciente al programa educativo de ISW en la Facultad de negocios y tecnologías

Mockups de la materia donde se genera el código qr

En este mockup se mostrará el diseño y el botón de generar el código qr



Datos Recibidos:

Irving Rafael Conde Marín, Irving Rafael Conde Marín 213642 zS20006735

Mockup del registro del alumno

En este mockup tendremos la libertad de poder tener nuestra propia cuenta para poder acceder al sitio web, tendremos que ingresar nuestros datos como tenemos en la imagen: Correo institucional, contraseña, tendremos que confirmar nuestra contraseña, tendremos otro apartado para poder seleccionar una imagen para nuestro usuario.

- Como siguiente daremos en crear cuenta y automáticamente tendremos nuestra cuenta creada.

The mockup shows a registration form titled "Registrar Cuenta" with the subtitle "Ingresa los datos solicitados". The form includes fields for "Correo" (with placeholder "@Correo"), "Contraseña" (with placeholder "Contraseña"), "Confirmar Contraseña" (with placeholder "Confirmar Contraseña"), and a file upload field showing "0" files selected. Below these fields is a button labeled "Elegir archivo" with the subtext "No se eligió ningún archivo". At the bottom of the form are two buttons: "Crear Cuenta" and "Iniciar Sesión!" with a user icon.

Registrar
Cuenta

Ingresa los datos solicitados

@Correo

Contraseña

Confirmar Contraseña

0

Elegir archivo No se eligió ningún archivo

Crear Cuenta

Iniciar Sesión!

Mockup login

En este mockup se muestra el diseño del login y los datos que se necesitan para ingresar



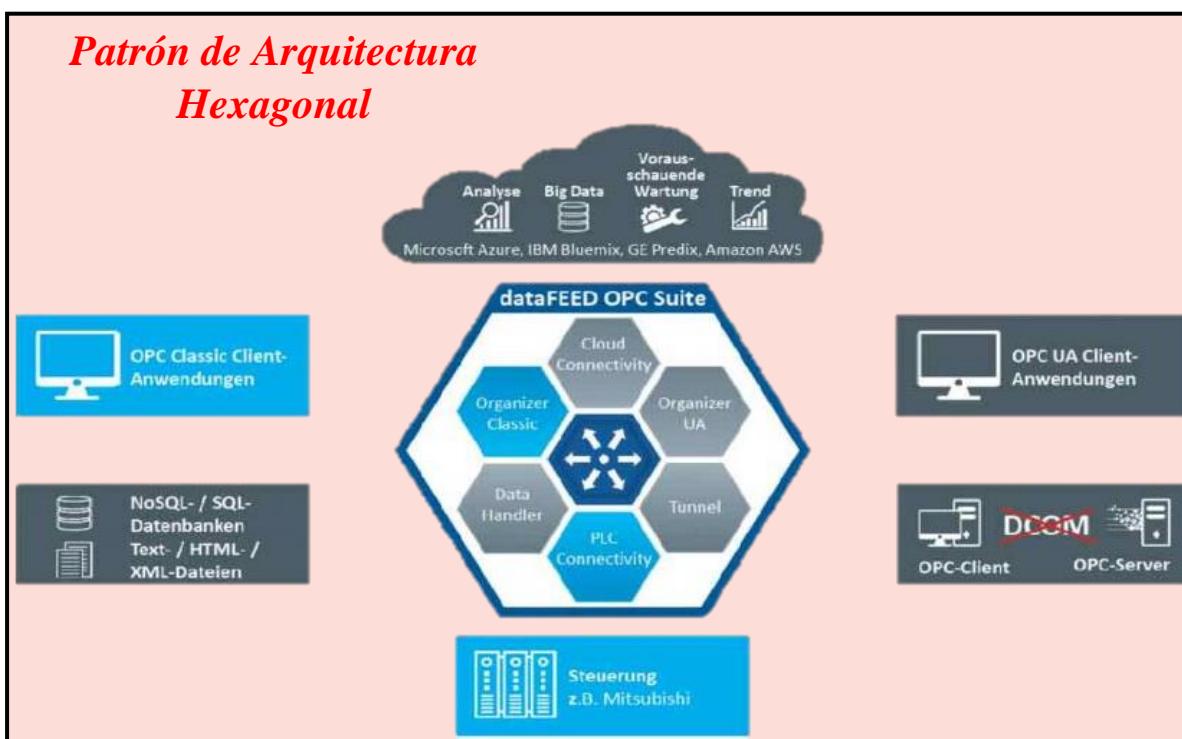
2.6.4 Arquitectura de software de la solución

Patrón arquitectónico: Arquitectura Hexagonal

Para la realización del proyecto I-Student, el equipo de desarrollo ha optado por utilizar el patrón arquitectónico conocido como: Patrón de Arquitectura Hexagonal.

El patrón de arquitectura hexagonal, también conocido como arquitectura de puertos y adaptadores, propone separar nuestra aplicación en distintas capas, cada una con su propia responsabilidad permitiendo así que evolucionen de forma aislada, y que cada una sea independiente a las demás.

Esto nos permitirá realizar cambios en nuestro sistema en caso de ser necesarios, un ejemplo claro sería la anexión de una nueva base de datos, o la inesperada e indeseada migración del front-end a otro framework, esto sin afectar el funcionamiento de las demás capas, haciendo posible realizar estos cambios con tan solo cambiar los puertos del sistema correspondientes.

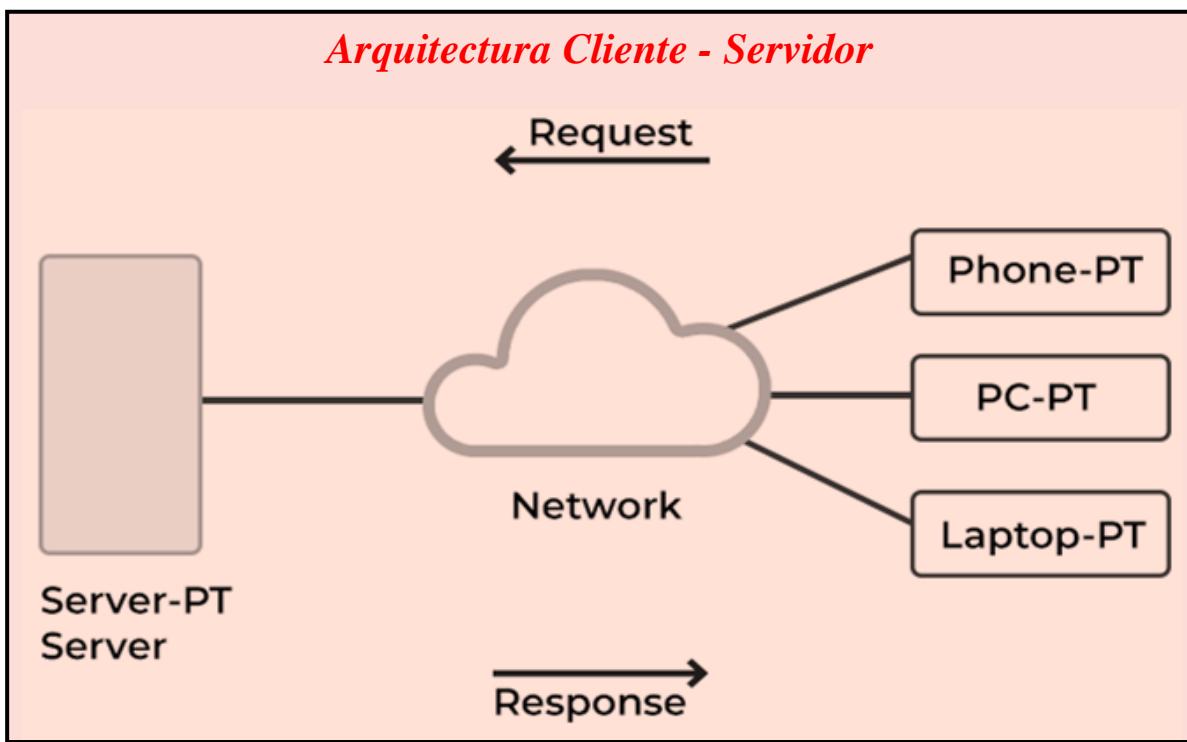


Arquitectura de software: Arquitectura Cliente - Servidor

Al tratarse el proyecto I-Student como un sistema escolar para la toma y gestión de las asistencias de los alumnos, así como una herramienta para el seguimiento de esto por parte de los docentes a través de una aplicación web, el equipo de desarrollo de I-Student ha optado por utilizar una arquitectura de software denominada como Cliente-Servidor.

Se ha elegido esta arquitectura de software debido a que permite una fácil interacción entre el cliente y el servidor, y proporciona una experiencia de usuario sin problemas, dándonos como sus componentes:

<i>Cliente</i>	
<i>Aplicación Web para Docentes</i>	Esta aplicación permitirá a los docentes llevar un seguimiento sobre las materias pertenecientes a este, así como las listas de asistencia de los alumnos con respecto a las dichas materias.
<i>Aplicación Web para Alumnos</i>	Esta aplicación permitirá a los alumnos generar un código QR para realizar el proceso de la toma de asistencia, además, contará con un apartado para verificar que su información almacenada en el sistema está correcta.
<i>Simulador de lectura de QR</i>	Esta aplicación permitirá realizar el proceso de toma de asistencia de forma automatizada, a través de la configuración del sistema con un edificio y un salón correspondiente para su utilización, por lo que posteriormente podrá empezar la lectura de códigos QR y realizará la toma de asistencia correspondiente de los alumnos.
<i>Servidor</i>	
<i>Base de datos</i>	El servidor contará con una conexión a la base de datos correspondiente, permitiendo almacenar toda la información relevante del sistema, como lo es la información de las cuentas creadas, la información acerca de las materias existentes, información acerca de los alumnos, etc.
<i>Servicios Web</i>	El servidor deberá contar con una serie de servicios web disponibles para satisfacer las necesidades del cliente, como lo serían servicios para la gestión de la asistencia de los alumnos, servicios para la consulta de información acerca de un alumno específico, etc.
<i>Reglas de negocio</i>	El servidor deberá tener bien definidas las reglas del negocio propuestas por el cliente, para realizar un esfuerzo óptimo en base a la necesidad presentada, por ejemplo, las reglas estipuladas para la detección de un estudiante con bajas asistencias en sus materias correspondientes, el manejo o formato de las listas de asistencia de las que harán uso los docentes en el sistema, etc.
<i>Seguridad</i>	El servidor deberá ser capaz de poder realizar los procesos solicitados por el usuario de una forma segura, sin exponer los datos manipulados en dichos procesos a un riesgo de perderse en sus caminos, un riesgo de ser robados por terceros, etc.



Componentes del sistema

Aplicación Web para Docentes

Módulo 1: Login del Sistema

Componente 1: Registro de cuenta

Componente 2: Inicio de sesión

Componente 3: Cerrar sesión

Módulo 2: Listado de materias

Componente 1: Menú para mostrar las materias

Módulo 3: Apartado de control de asistencia

<i>Componente 1:</i>	Lista de asistencia
<i>Componente 2:</i>	Toma de asistencia en tiempo real
<i>Componente 3:</i>	Clonación de archivo .XLS

Módulo 4: Manejo de Materias

<i>Componente 1:</i>	Alta de materias
<i>Componente 2:</i>	Baja de materias

Aplicación Web para Estudiantes

Módulo 1: Login del Sistema

<i>Componente 1:</i>	Registro de cuenta
<i>Componente 2:</i>	Inicio de sesión
<i>Componente 3:</i>	Cerrar sesión

Módulo 2: Menú Principal

<i>Componente 1:</i>	Apartado de la información registrada
<i>Componente 2:</i>	Visualización de las materias registradas
<i>Componente 3:</i>	Generación del código QR

Sensor para la toma de asistencia (*Simulado con aplicación web*)

Módulo 1: Escáner

Componente 1:	Conectividad con Hardware
Componente 2:	Configuración del equipo

Módulo 2: Lectura de códigos QR

Componente 1:	Escaneo de códigos QR
Componente 2:	Mostrar información de QR
Componente 3:	Realizar asistencia
Componente 4:	Almacenar la lectura en el historial

Módulo 3: Historial de escaneo

Componente 1:	Mostrar Historial de lectura de códigos QR
----------------------	---

Servidor (Api Docentes)

Módulo 1: Procedimientos a la base de datos

<i>Componente 1:</i>	Crear Cuentas
<i>Componente 2:</i>	Consultar Información Docente
<i>Componente 3:</i>	Validar Inicios de Sesión
<i>Componente 4:</i>	Consultar Materias Docente
<i>Componente 5:</i>	Realizar Alta Materia
<i>Componente 6:</i>	Realizar Baja Materia

Módulo 2: Conexión a la base de datos

<i>Componente 1:</i>	Inicio de sesión
<i>Componente 2:</i>	Cierre de sesión
<i>Componente 3:</i>	Configuración de las credenciales

Módulo 3: Conexión a IAAS

<i>Componente 1:</i>	Configuración de las credenciales
----------------------	--

Servidor (Api Alumnos)

Módulo 1: Procedimientos a la base de datos

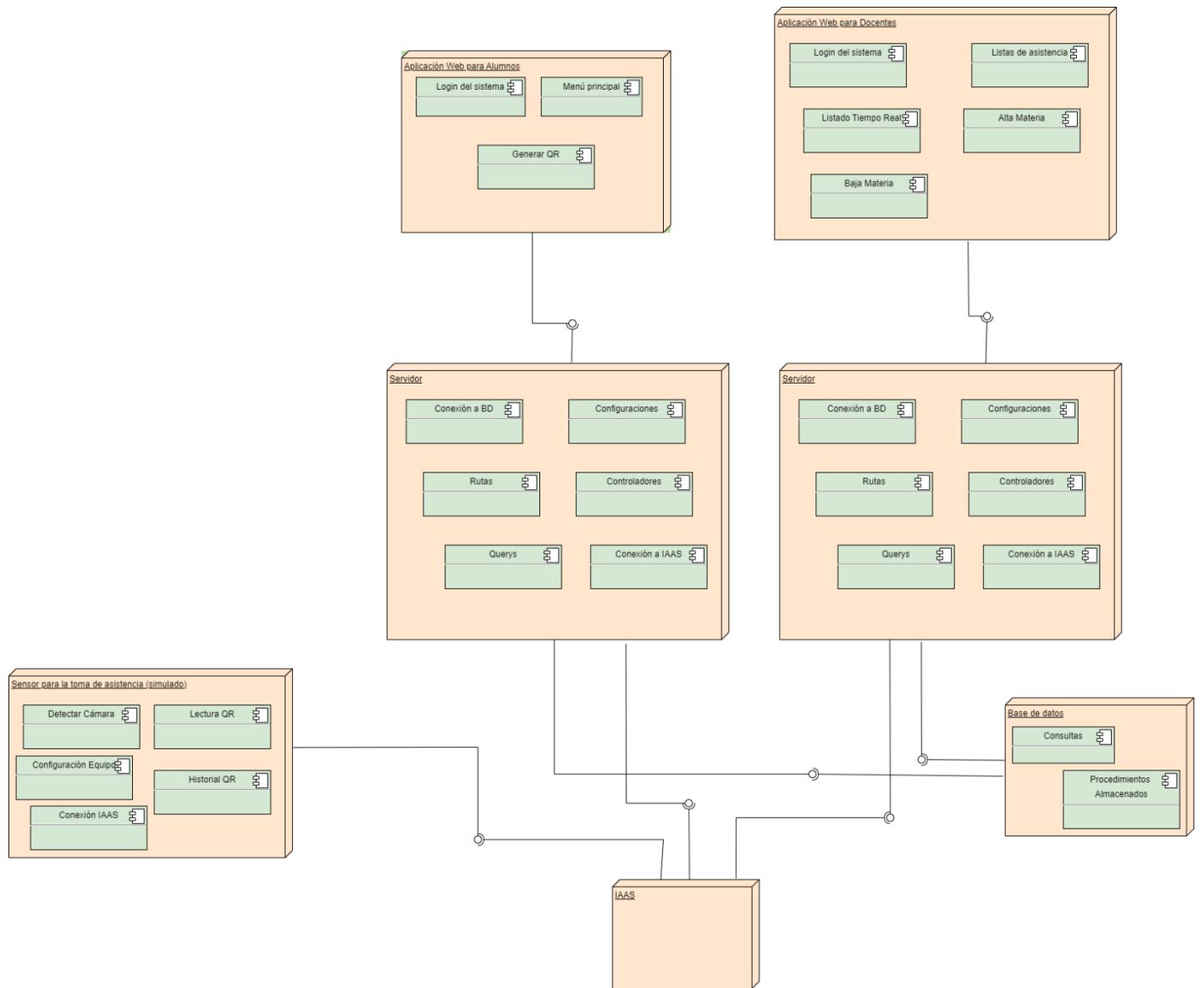
<i>Componente 1:</i>	Crear Cuentas
<i>Componente 2:</i>	Consultar Información Alumno
<i>Componente 3:</i>	Validar Inicios de Sesión
<i>Componente 4:</i>	Consultar Materias Alumno

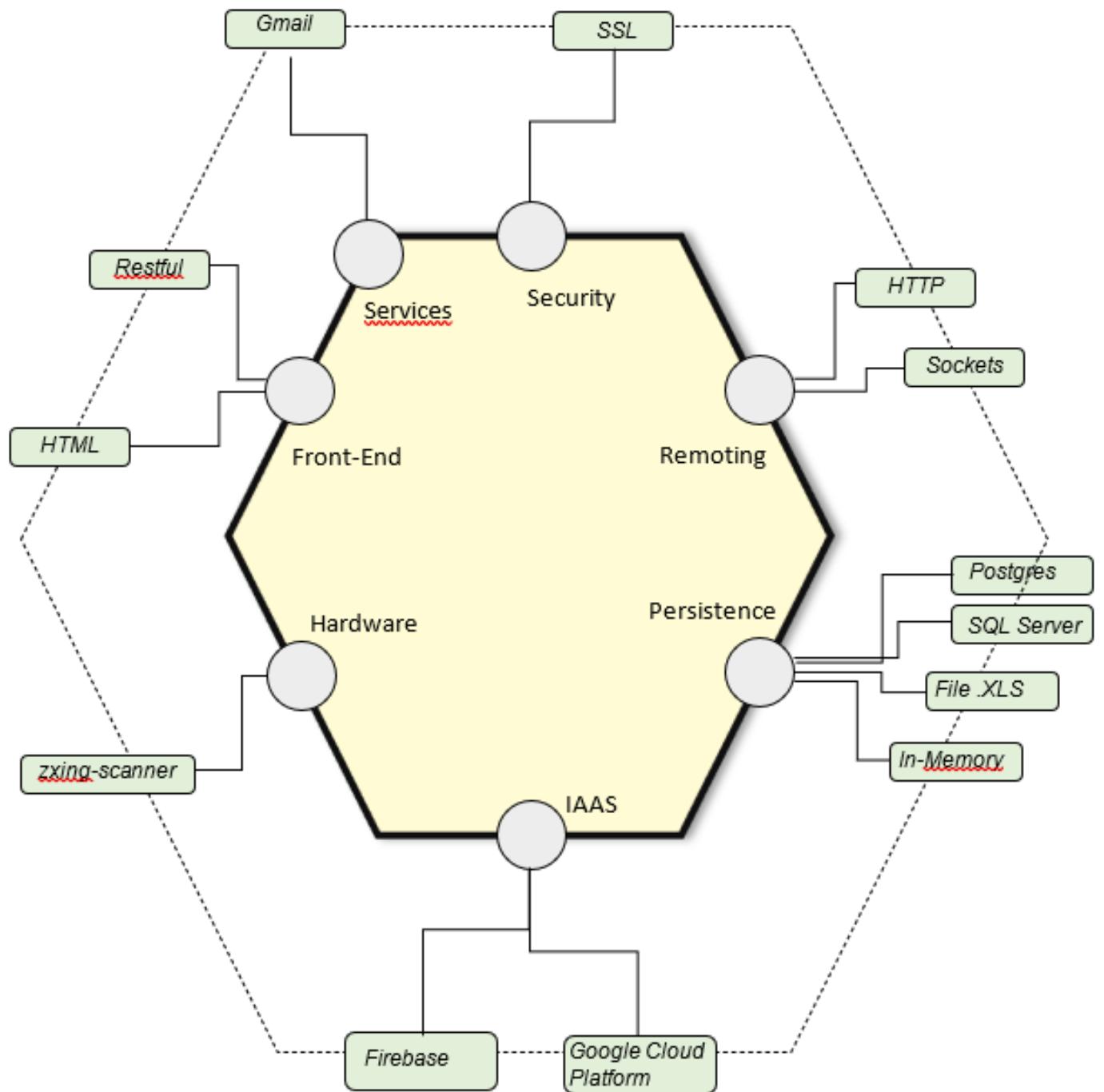
Módulo 2: Conexión a la base de datos

<i>Componente 1:</i>	Inicio de sesión
<i>Componente 2:</i>	Cierre de sesión
<i>Componente 3:</i>	Configuración de las credenciales

Módulo 3: Conexión a IAAS

<i>Componente 1:</i>	Configuración de las credenciales
----------------------	--

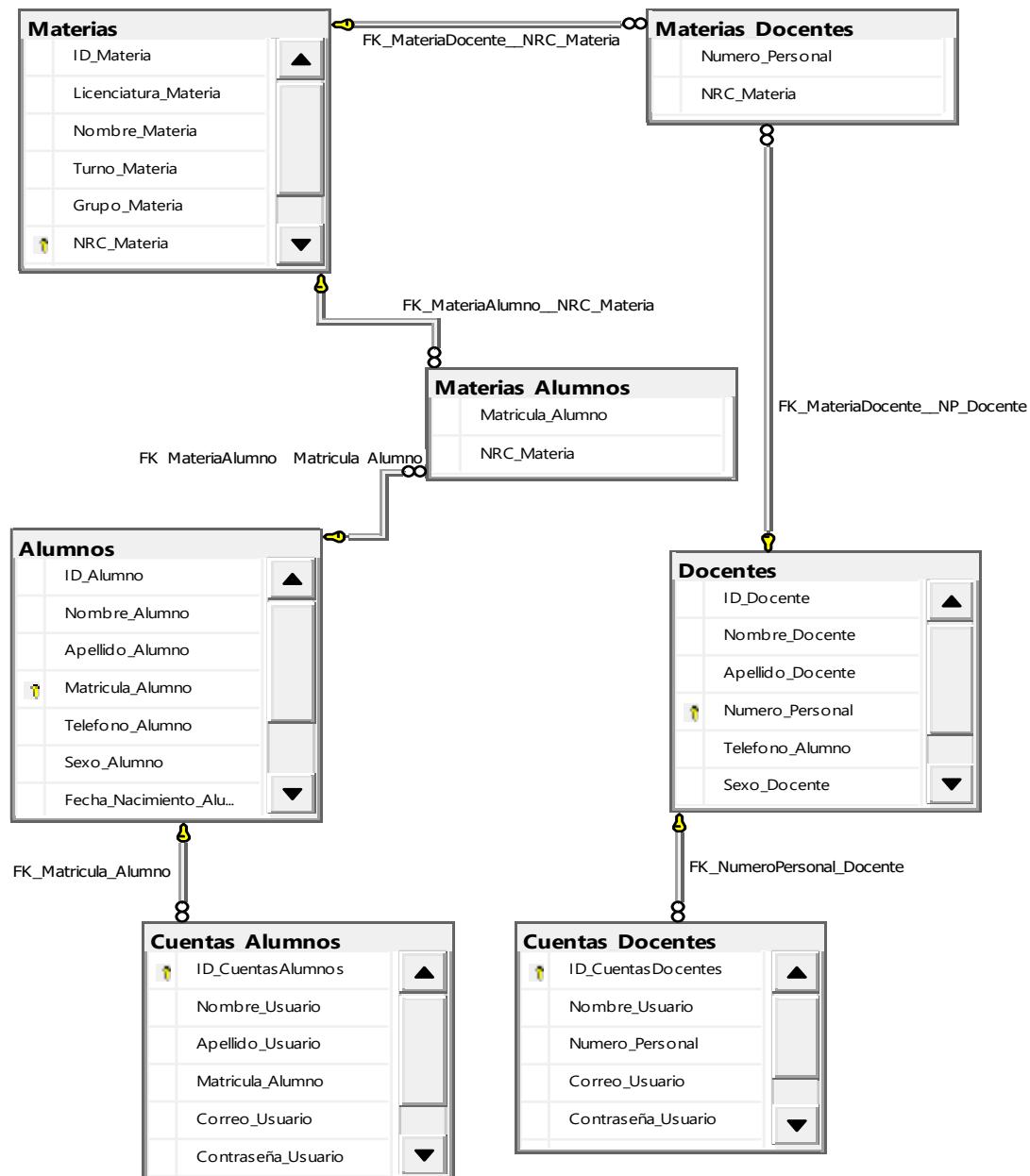




2.6.5 Modelo de datos

Debido a que se utilizarán dos tipos de bases de datos se realizaron dos modelos de datos.

Modelo de datos relacional:



Diccionario de datos

TABLA ALUMNOS

NOMBRE CAMPO	Tipo Dato	Longitud Dato	Valor
ID_ALUMNO	INT		Unique, Not Null
NOMBRE_ALUMNO	VARCHAR	30	Not Null
APELLIDO_ALUMNO	VARCHAR	30	Not Null
MATRICULA_ALUMNO	VARCHAR	10	Llave Primaria, Not Null
TELEFONO_ALUMNO	VARCHAR	10	Null
SEXO_ALUMNO	CHAR		Not Null
FECHA_NACIMIENTO_ALUMNO	VARCHAR	10	Not Null
LICENCIATURA_ALUMNO	VARCHAR	20	Not Null

TABLA DOCENTES

NOMBRE CAMPO	Tipo Dato	Longitud Dato	Valor
ID_DOCENTE	INT		Unique, Not Null
NOMBRE_DOCENTE	VARCHAR	30	Not Null
NUMERO_PERSONAL	VARCHAR	10	Llave Primaria
TELEFONO_ALUMNO	VARCHAR	10	Null
SEXO_DOCENTE	CHAR		Not Null
FECHA_NACIMIENTO_DOCENTE	VARCHAR	10	Not Null

TABLA MATERIAS

NOMBRE CAMPO	Tipo Dato	Longitud Dato	Valor
ID_MATERIA	INT		Unique
LICENCIATURA_MATERIA	VARCHAR	20	Not Null
NOMBRE_MATERIA	VARCHAR	30	Not Null
TURNO_MATERIA	VARCHAR	10	Not Null
GRUPO_MATERIA	INT		Not Null
NRC_MATERIA	INT		Llave Primaria
CUPO_MATERIA	INT		Not Null

TABLA CUENTAS_ALUMNOS

NOMBRE CAMPO	Tipo Dato	Longitud Dato	Valor
ID_CUENTAS_ALUMNOS	INT		Unique
NOMBRE_USUARIO	VARCHAR	30	Not Null
APELLIDO_USUARIO	VARCHAR	30	Not Null
MATRICULA_ALUMNO	VARCHAR	10	Not Null, Llave Foránea Llave primaria
CORREO_USUARIO	VARCHAR	30	Not Null
CONTRASEÑA_USUARIO	VARCHAR	200	Not Null
FOTO_USUARIO	VARCHAR	250	Not Null

TABLA CUENTAS_DOCENTES

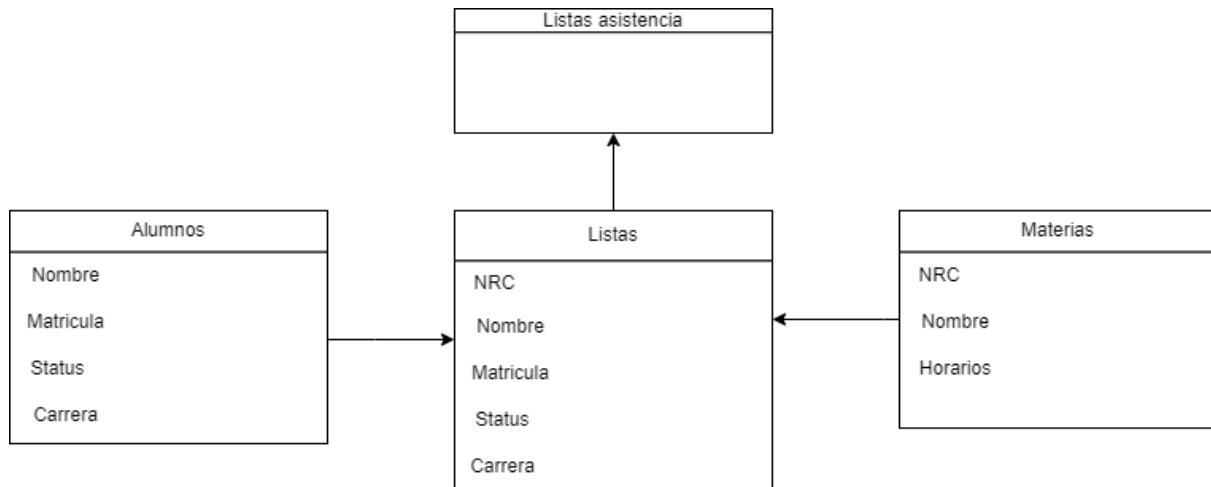
NOMBRE CAMPO	Tipo Dato	Longitud Dato	Valor
ID_CUENTAS_DOCENTES	INT		Unique
NOMBRE_USUARIO	VARCHAR	30	Not Null
NUMERO_PERSONAL	VARCHAR	10	Not Null, Llave Foránea, Llave primaria
CORREO_USUARIO	VARCHAR	30	Not Null
CONTRASEÑA_USUARIO	VARCHAR	200	Not Null
FOTO_USUARIO	VARCHAR	250	Not Null

TABLA MATERIAS_DOCENTES

NOMBRE CAMPO	Tipo Dato	Longitud Dato	Valor
NUMERO_PERSONAL	VARCHAR	10	Not Null, Llave Foránea
NRC_MATERIA	INT		Not Null, Llave Foránea

NOMBRE CAMPO	TIPO DATO	LONGITUD DATO	VALOR
MATRICULA_ALUMNO	VARCHAR	10	Not Null, Llave Foránea
NRC_MATERIA	INT		Not Null, Llave Foránea

Modelo de datos no relacional:



2.7 Implementación

2.7.1 Descripción de metodología/ ciclo de vida empleado para la implementación de la solución

Debido al tipo del proyecto SACAEMEA, y el tiempo limitante que se tenía para llevarlo a cabo, el equipo de desarrollo optó por utilizar una metodología ágil, la cual es denominada como metodología Scrum.

¿Qué es Scrum?

Es un marco de trabajo ágil que incluye entregas rápidas y un alto poder de decisión del equipo. Es ideal para proyectos de desarrollo de software, marketing y de otras áreas donde se requiere llegar a objetivos en el menor tiempo posible, con una gran flexibilidad para satisfacer las necesidades del cliente.

A sus ciclos cortos de desarrollo se les denomina Sprints, los cuales son pequeñas tareas que se cumplen en iteraciones de tiempo fijos, los cuales normalmente oscilan entre 1 semana y 4 semanas, dependiendo la complejidad de estos. Al iniciar un sprint, el equipo de trabajo debe establecer un objetivo para el sprint, así como también las tareas que se deberán cumplir o llevar a cabo para alcanzar dicho objetivo en el tiempo estipulado de duración. Los Sprint también deben llevar a cabo un control y monitoreo, para cumplir esto se deberán realizar reuniones de trabajo para poder visualizar el progreso de las tareas denominadas en este, así como también llevar a cabo la resolución de problemas presentados durante el tiempo de vida del sprint, y si es necesario, aplazar la fecha de entrega.

Un aspecto importante que caracteriza a Scrum es la comunicación que debe haber entre los involucrados del proyecto, esto a través de sus reuniones regulares donde deberán ser compartidas actualizaciones, obstáculos, incidencias, inconformidades, etc.

¿Cuándo utilizar Scrum?

No es necesario, ni siquiera conveniente, utilizar la metodología SCRUM en todo tipo de proyectos. Por eso debes saber qué requerimientos debe tener tu proyecto si quieras utilizarla con eficacia:

- Equipos pequeños: SCRUM funciona mejor con equipos de trabajo reducidos, generalmente no superando las 8 personas.
- Poca necesidad de documentación: Si el proyecto requiere una documentación detallada desde el inicio, con fases de consultoría y toma de requerimientos prolongadas, SCRUM puede no ser la metodología adecuada. Sin embargo, si el enfoque se centra en entregas rápidas y un mayor control del proyecto, SCRUM resultará muy útil.
- Proyectos con riesgos de cambios durante el proceso: SCRUM se adapta bien a proyectos donde se espera que los requisitos cambien durante el proceso. Debido a que SCRUM divide el proyecto en fases cortas de dos a cuatro semanas, proporciona flexibilidad para incorporar cambios a mitad del camino, ya que se revisan las tareas y objetivos después de cada fase.
- Confianza en la metodología: Es fundamental comprender los principios y la forma de operar de SCRUM antes de aplicarla en un proyecto. Como responsable, debes sentirte cómodo con esta metodología y transmitir confianza al resto de los participantes del proyecto.

Eventos típicos de la metodología Scrum

- Organiza tu trabajo pendiente: Antes de comenzar un sprint de Scrum, es importante que el líder del equipo, también conocido como Scrum Master, organice el trabajo pendiente que debe realizarse. Esto implica identificar las tareas y requerimientos del producto que serán abordados durante el sprint.
- Sprint Planning: Realiza una sesión de planificación del sprint. Antes de iniciar el sprint de Scrum, es necesario realizar una sesión de planificación para determinar en qué se enfocará el equipo durante ese sprint específico. Durante esta sesión, se evalúa el trabajo pendiente y se establecen las metas y objetivos a alcanzar.
- Inicia el sprint de Scrum: El sprint de Scrum comienza una vez que se ha realizado la planificación adecuada. Durante el sprint, el equipo se dedica a trabajar en las tareas establecidas en la planificación del sprint.
- Daily Stand Up: Organiza reuniones diarias de seguimiento. Las reuniones diarias de seguimiento, conocidas como Daily Stand Up, son una parte fundamental de Scrum. Estas reuniones cortas, de aproximadamente 15 minutos, se llevan a cabo todos los días y permiten al equipo informar sobre el progreso realizado, identificar posibles obstáculos y planificar las actividades del día.
- Sprint Review: Presenta el trabajo realizado en la revisión del sprint. Una vez finalizado el sprint, es importante realizar una revisión para presentar el trabajo realizado. Durante el sprint review, el equipo Scrum muestra y discute las funcionalidades o tareas completadas durante el sprint.
- Sprint Retrospective: Reflexiona y analiza el sprint. Al finalizar cada sprint, es recomendable realizar una retrospectiva para evaluar cómo se desarrolló el sprint y qué se puede mejorar en el futuro. La retrospectiva es una parte esencial de Scrum, ya que promueve la mejora continua del equipo y del proceso.

Actores o roles

Dentro de la metodología Scrum, se asignan cuatro roles fundamentales que desempeñan funciones clave en el proceso:

- Propietario del producto (Product Owner): Esta persona es responsable de representar los intereses de los usuarios o de la empresa y tiene la autoridad para tomar decisiones sobre las prioridades del proyecto. El Propietario del producto trabaja en estrecha colaboración con el equipo de desarrollo para definir y comunicar los requisitos y objetivos del producto.
- Equipo de desarrollo: El equipo de desarrollo está compuesto por profesionales con habilidades y conocimientos necesarios para llevar a cabo el trabajo y entregar el producto. Son responsables de la implementación y la entrega del producto en incrementos iterativos, siguiendo las directrices y prioridades establecidas por el Propietario del producto.
- Facilitador del proyecto (Scrum Master): El Facilitador del proyecto es el encargado de garantizar que el equipo de desarrollo pueda trabajar de manera eficiente y sin obstáculos. Su función principal es eliminar impedimentos y promover un entorno de trabajo colaborativo. El Scrum Master facilita las reuniones y ayuda al equipo a seguir las prácticas y principios de Scrum.
- Interesados (Stakeholders): Los interesados son aquellos individuos o grupos que tienen un interés directo en el producto y su éxito. Pueden ser clientes, patrocinadores, directivos de la empresa u otras partes interesadas relevantes. Los interesados desempeñan un papel importante al proporcionar retroalimentación y participar en la revisión y la validación del producto en cada iteración.

Estos roles trabajan en conjunto, colaborando y comunicándose de manera efectiva para garantizar el desarrollo y la entrega exitosa del producto en el marco de tiempo establecido. Cada uno de ellos tiene responsabilidades específicas que contribuyen al éxito general del proyecto Scrum.

Características destacadas de la metodología Scrum

- Flexibilidad y adaptabilidad: A diferencia de otros métodos que buscan eliminar la incertidumbre, Scrum reconoce la naturaleza cambiante de los proyectos y fomenta la flexibilidad y adaptabilidad en el proceso. Esto permite respuestas más novedosas y creativas a medida que se avanza en el proyecto.
- Organización autónoma de equipos: Scrum promueve la autonomía de los equipos de desarrollo, permitiéndoles encontrar su propia forma de trabajar y establecer metas.
- Desarrollo simultáneo en fases: Scrum reconoce que diferentes áreas pueden tener diferentes ritmos y tiempos establecidos. A pesar de ello, el proyecto en su conjunto debe mantener un pulso que se sincroniza a lo largo del tiempo. Esta característica permite la integración y el debate de diversos puntos de vista, lo que puede dar lugar a respuestas extraordinarias.
- Aprendizaje continuo: Scrum promueve un entorno de aprendizaje constante, donde los equipos dinámicos pueden adaptarse con flexibilidad a las dificultades y cambios en el mercado. Se fomenta el aprendizaje individual y grupal, permitiendo que los miembros del equipo aporten conocimientos y descubrimientos inesperados.
- Control sutil: En lugar de controles de calidad estructurados, Scrum utiliza puntos de inversión para reducir ambigüedades y tensiones, y proporcionar estabilidad. Esto implica contar con el talento adecuado, recordar al cliente, estimular al equipo, mantener un ambiente de trabajo abierto y gestionar los conflictos de manera efectiva. También se toleran los errores y se corrigen a tiempo.
- Transmisión organizacional del aprendizaje: Scrum se propaga en toda la organización a medida que se implementa en diferentes áreas. Los conocimientos adquiridos en proyectos anteriores, incluso aquellos que pudieron haber fracasado, se utilizan para superar los retos actuales.

Artefactos de la metodología

Los artefactos de scrum ofrecen información importante que el equipo de scrum utiliza para definir el producto y el trabajo que hay que hacer para crearlo. Existen tres artefactos en scrum: un backlog del producto, un backlog de sprint y un incremento en tu definición de "finalizado". Son las tres constantes sobre las que un equipo de scrum debe reflexionar durante los sprints y a lo largo del tiempo.

- Backlog del producto: El backlog del producto es una lista dinámica de funcionalidades, requisitos, mejoras y correcciones que deben ser realizadas por el propietario o responsable del producto. Actúa como la base para el backlog de sprint y representa todas las tareas que el equipo necesita realizar. El propietario del producto constantemente revisa, cambia las prioridades y realiza el mantenimiento del backlog del producto, eliminando elementos obsoletos o que ya no son relevantes, y asegurándose de que se aborden los problemas de manera adecuada.
- Backlog de sprint: El backlog de sprint es la lista de elementos, historias de usuario o correcciones de errores que son seleccionados por el equipo de desarrollo para ser implementados durante el ciclo de sprint actual. Durante la reunión de planificación de sprint, el equipo elige los elementos del backlog del producto en los que trabajará durante el sprint. El backlog de sprint puede ser flexible y evolucionar durante el sprint, pero el objetivo fundamental del sprint no debe verse comprometido. El equipo se esfuerza por lograr el objetivo establecido para el sprint actual.
- Incremento (u objetivo del sprint): El incremento es el resultado final utilizable al final de un sprint. Durante la demostración de fin de sprint, el equipo muestra lo que se ha completado en el sprint, lo cual puede corresponder a la definición de "finalizado" establecida por el equipo. Dependiendo de los objetivos y la naturaleza del proyecto, el incremento puede ser considerado como un hito, una versión completa, un epic lanzado u otro logro específico. Cada equipo define su propia definición de "finalizado" y establece sus objetivos de sprint de acuerdo con sus necesidades.

2.7.2 Evidencias de trabajo colaborativo en repositorio de código fuente

Repositorio de Front Docentes

Rama Master con los funcionamientos del sistema operando correctamente

The screenshot shows the GitHub repository page for `IrvingCM123/Sistema_Asistencia`. The master branch has 123 commits from `camgios`, with the latest commit being 3 days ago. The repository is public and has 1 star, 1 watching, and 0 forks. The sidebar shows activity, including 1 star, 1 watching, and 0 forks.

File	Author	Date
<code>.vscode</code>	Sistema	2 months ago
<code>src</code>	camgios	3 days ago
<code>.browserslistrc</code>	Sistema	2 months ago
<code>.editorconfig</code>	Sistema	2 months ago
<code>.gitignore</code>	Sistema	2 months ago
<code>angular.json</code>	cambios	3 weeks ago
<code>karma.conf.js</code>	Sistema	2 months ago
<code>package-lock.json</code>	cambios	2 months ago
<code>package.json</code>	cambios	2 months ago

Proyecto en github con los cambios realizados por el sistema

The screenshot shows the Asana project `@IrvingCM123's Sistema Docentes`. There are 7 tasks listed, all assigned to `IrvingCM123` and marked as `Done`.

Title	Assignees	Status
1 Configuración de las carpetas para la arquitectura hexagonal #2	IrvingCM123	Done
2 Puertos y adaptadores #3	IrvingCM123	Done
3 Archivos de Domain #4	IrvingCM123	Done
4 Archivos de Views #5	IrvingCM123	Done
5 Iniciar Proyecto #1	IrvingCM123	Done
6 Función Listado #9	IrvingCM123	Done
7 Función Docentes #7	IrvingCM123	Done

Periodo de 1 mes de trabajo



Repositorio de Front Alumnos

Rama Master con los funcionamientos del sistema operando correctamente

EmilianoTorres123 / alumnos-sistema Public

Watch 1 Fork 0 Star 0

Code Issues Pull requests Actions Projects Security Insights

master 1 branch 0 tags Go to file Add file ▾

About
No description, website, or topics provided.

Code
Readme Activity 0 stars 1 watching 0 forks Report repository

Releases
No releases published

Packages

EmilianoTorres123 actualizacion de codigo	5da3a93 4 minutes ago	3 commits
.vscode	initial commit	4 days ago
src	actualizacion de codigo	4 minutes ago
.editorconfig	initial commit	4 days ago
.gitignore	initial commit	4 days ago
README.md	initial commit	4 days ago
angular.json	initial commit	4 days ago
package-lock.json	qr-code	4 days ago
package.json	qr-code	4 days ago
tsconfig.app.json	initial commit	4 days ago

Periodo de 1 mes de trabajo



Repositorio de Back-End Alumnos

Rama Master con los funcionamientos del sistema operando correctamente

FerminDRA / backendAlumnos Public

Code Issues Pull requests Actions Projects Security Insights

master 4 branches 0 tags

Go to file Add file ▾

About

No description, website, or topics provided.

Activity 0 stars 2 watching 0 forks Report repository

Commits

FerminDRA create backendAlumnos projet 6008f01 3 weeks ago 1 commit

node_modules create backendAlumnos projet 3 weeks ago

index.js create backendAlumnos projet 3 weeks ago

package-lock.json create backendAlumnos projet 3 weeks ago

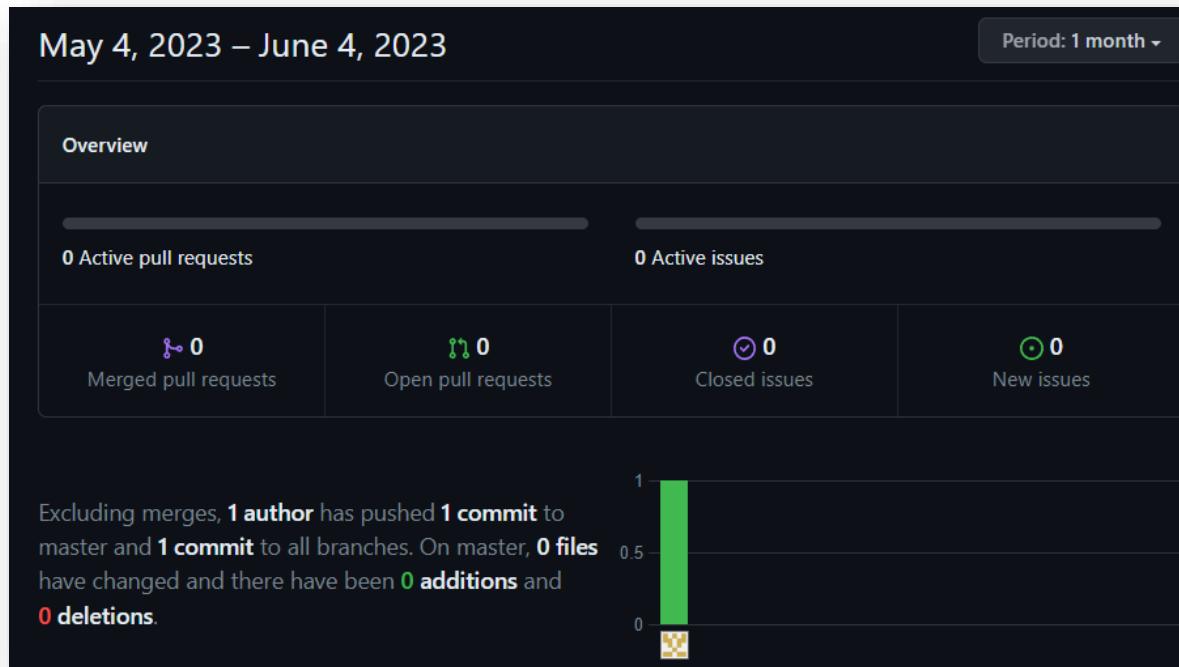
package.json create backendAlumnos projet 3 weeks ago

queries.js create backendAlumnos projet 3 weeks ago

Releases

No releases published

Periodo de 1 mes de trabajo

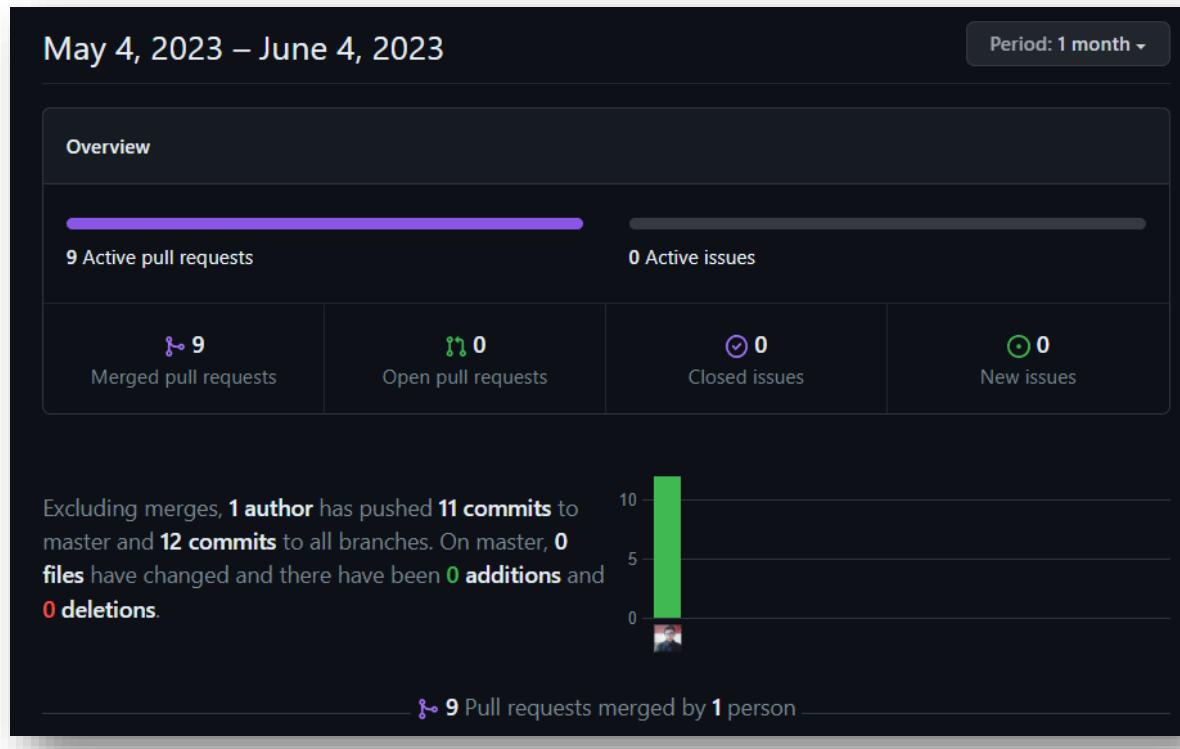


Repositorio de Back-End Docentes

Rama Master con los funcionamientos del sistema operando correctamente

The screenshot shows the GitHub repository page for IrvingCM123 / ApiNode. The repository is public and has 6 branches. The master branch is selected. A message indicates that the master branch isn't protected. The repository has 0 stars, 1 watching, and 0 forks. The About section notes: "No description, website, or topics provided." The Releases section shows no releases published and a link to "Create a new release". The Packages section shows no packages published. The Pull requests section lists a merge pull request from IrvingCM123 / release to master, which was merged 3 days ago with 21 commits. The commit history shows changes in request, src, .babelrc, .env, .gitignore, and package-lock.json files over the past 4 days.

Periodo de 1 mes de trabajo



Colaboradores

Manage access

Select all

Find a collaborator...

<input type="checkbox"/>		Adriel Peregrina Awaiting Adpsec's response
--------------------------	---	---

Repositorio de Front-End Simulador Escaner

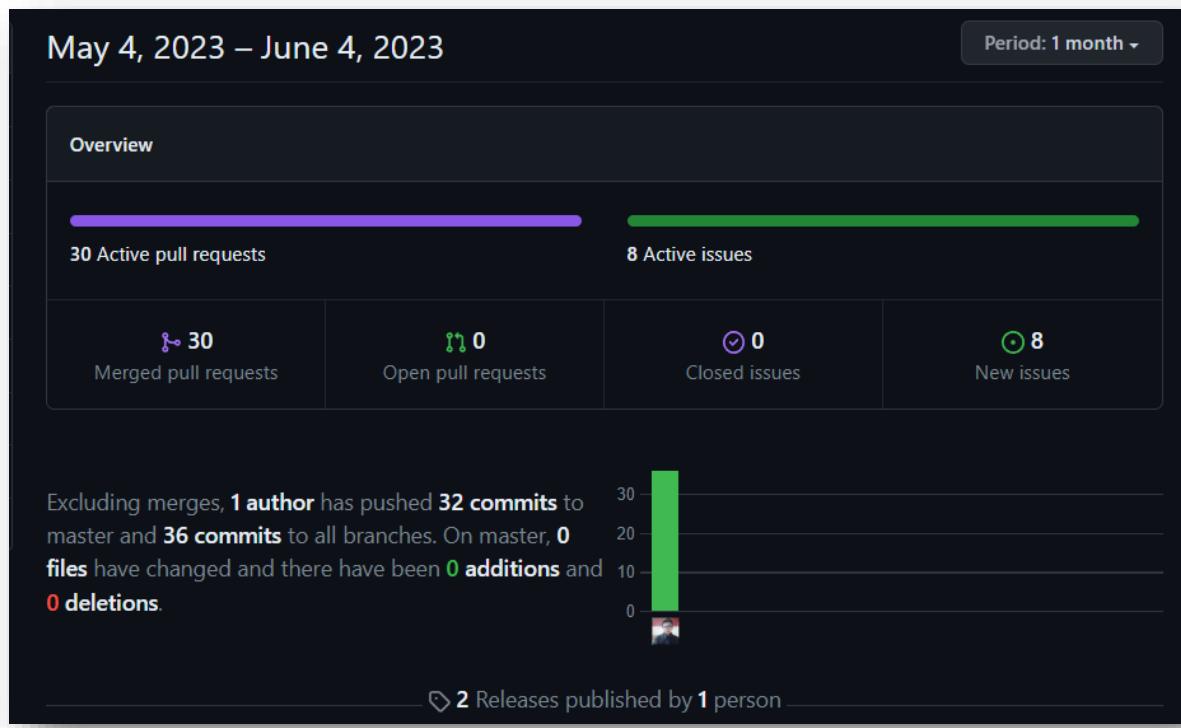
Rama Master con los funcionamientos del sistema operando correctamente

The screenshot shows the GitHub repository page for 'IrvingCM123 / Simulador_Escanner'. The 'Code' tab is selected. The commit history for the 'master' branch is displayed, showing 64 commits over the last month. The commits are mostly related to changing file paths ('cambiar rutas') across various files like .VSCodeCounter, .vscode, src, .browserslistrc, .editorconfig, .gitignore, README.md, angular.json, and karma.conf.js. The repository has 8 issues, 1 pull request, 1 project, and 0 forks. It is public and has 0 stars. The repository URL is simulador-escaner.vercel.app, and it uses technologies like JavaScript, HTML, behavior, angular, typescript, css3, firebase-database, zxing-qrcode, bootstrap5, zxing-qrscanreader, and firebase-auth.

Proyecto en github con los cambios realizados por el sistema

The screenshot shows the GitHub Projects board for the repository '@IrvingCM123's Sistema Escaner'. The board displays seven tasks: 'Iniciar Proyecto #8', 'Branch Develop #9', 'Branch Release #10', 'Funcionalidad Configuración #12', 'Funcionalidad QR #11', 'Camara #7', and 'Funcionalidad Mandar_Datos #13'. Each task has an assignee (IrvingCM123) and a status: 'Done' for tasks 1, 5, 6, and 7; 'In Progress' for tasks 2, 3, and 4; and 'Not Started' for task 8. The tasks are listed in a grid format with columns for Title, Assignees, Status, and Fecha Inicio. A search bar at the top right allows users to search or jump to specific items. A note at the bottom indicates that users can use Control + Space to add an item.

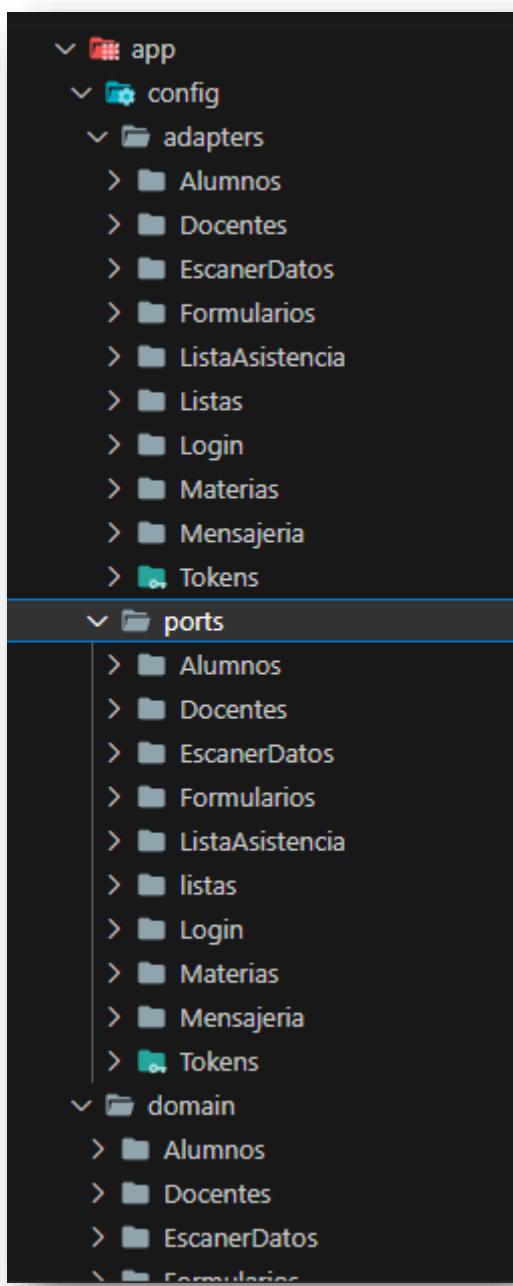
Periodo de 1 mes de trabajo



2.7.3 Capturas de pantalla con la descripción de las partes del código más relevante

Front End de Docentes

Organización de carpetas para manejar Arquitectura Hexagonal en el sistema



Adaptador para poder comunicarse con el escaner a través de la nube Firestore, aquí se consultará el registro de toma de asistencia en tiempo real, utilizando las funcionalidades de tiempo real que nos ofrecen las bibliotecas de firebase

```

11
12     constructor(
13       private firestore: AngularFirestore
14     ) { }
15
16   async getEscanerDatos(nrc: string, diaLista: string): Promise<Observable<EscanerDatos_Entity>> {
17     let url = '/Registro/Asistencia/' + nrc + '/' + diaLista + '/Alumnos';
18
19     return new Observable<EscanerDatos_Entity>((observer) => {
20       const collectionRef = this.firebaseio.collection(url);
21
22       collectionRef.snapshotChanges().subscribe((snapshot) => {
23         const datos_leidos: EscanerDatos_Entity[] | any = [];
24         snapshot.forEach((doc) => {
25           const data = doc.payload.doc.data() as EscanerDatos_Entity;
26           const id = doc.payload.doc.id;
27           datos_leidos.push({ id, ...data });
28         });
29
30         observer.next(datos_leidos);
31       }, (error) => {
32         console.error('Error al obtener los datos de Firestore:', error);
33         observer.error(error);
34       });
35     });
36   }
37 }
38

```

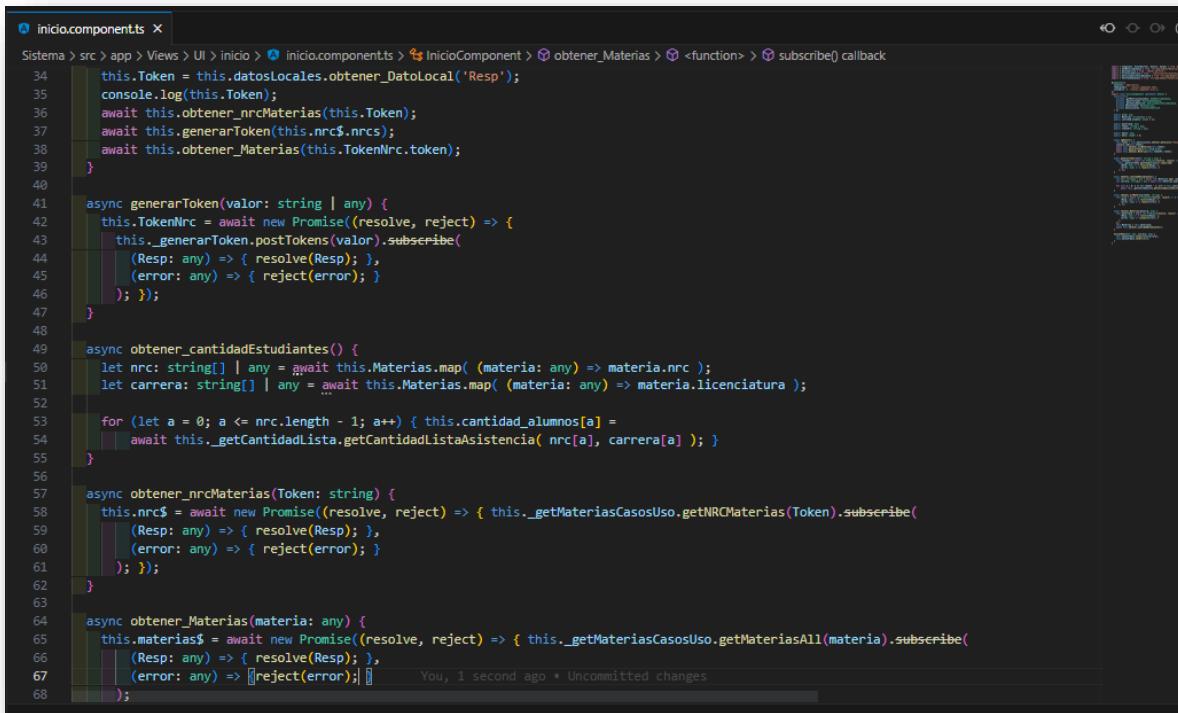
Adaptador para poder consultar las listas de asistencia dependiendo la carrera y el nrc solicitado, además, contará los registros de los datos almacenados en el sistema para llevar un control sobre qué tantos alumnos hay registrados en la materia

```

16   async getCantidadListaAsistencia(nrc: string, carrera: string): Promise<Observable<Number>> {
17     try {
18       let apiFirestore = '/' + carrera + '/Materias/' + nrc;
19       const lista_encontrada = await this.firebaseio.collection(apiFirestore).get().toPromise();
20       if (lista_encontrada) {
21         const datos_lista: any = await lista_encontrada.docs.map((alumnos) => alumnos.data());
22         let contador = datos_lista.length;
23         return contador;
24       } else {
25         console.log('No se pudo obtener la información de Firestore.');
26         let error: any = [];
27         return error;
28       }
29     } catch (error) {
30       console.error('Error al obtener la información de Firestore:', error);
31       let retornar: any = [];
32       return retornar;
33     }
34   }
35
36   async getListadoAsistenciaByNrcCarrera(nrc: string, carrera: string): Promise<Observable<ListaAsistencia_Entity>> {
37     try {
38       let apiFirestore = '/' + carrera + '/Materias/' + nrc;
39       const lista_encontrada = await this.firebaseio.collection(apiFirestore).get().toPromise();
40       if (lista_encontrada) {
41         const datos_lista: any = await lista_encontrada.docs.map((alumnos) => alumnos.data());
42         return datos_lista;
43       } else {
44         console.log('No se pudo obtener la información de Firestore.');
45         let error: any = [];
46         return error;
47       }
48     } catch (error) {
49       console.error('Error al obtener la información de Firestore:', error);
50       let retornar: any = [];

```

Consultar la información necesaria para llenar los apartados con la información de las materias a través de sus respectivos nrc, así como la implementación del método para el conteo de estudiantes. Todo esto en base al uso de tokens para transportar de manera segura el número de personal del docente almacenado en el caché del navegador

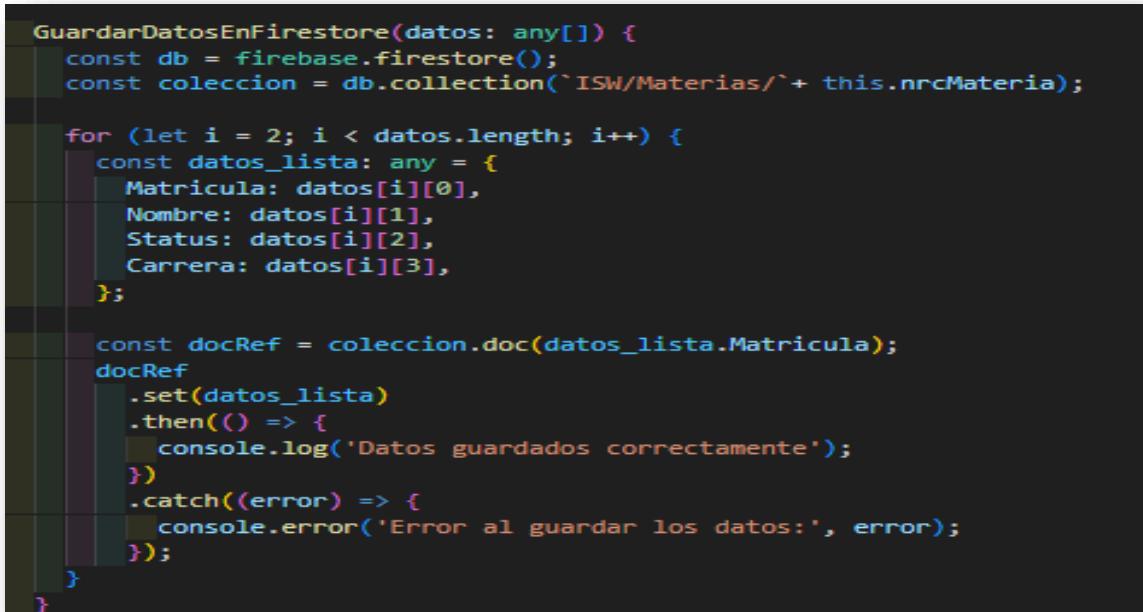


```

1 inicio.component.ts
Sistema > src > app > Views > UI > inicio > inicio.component.ts > InicioComponent > obtener_Materias > <function> > subscribe() callback
24     this.Token = this.datosLocales.obtener_DatoLocal('Resp');
25     console.log(this.Token);
26     await this.obtener_nrcMaterias(this.Token);
27     await this.generarToken(this.nrc$.nrcs);
28     await this.obtener_Materias(this.TokenNrc.token);
29   }
30
31   async generarToken(valor: string | any) {
32     this.TokenNrc = await new Promise((resolve, reject) => {
33       this._generarToken.postTokens(valor).subscribe(
34         (Resp: any) => { resolve(Resp); },
35         (error: any) => { reject(error); }
36       );
37     });
38
39   async obtener_cantidadEstudiantes() {
40     let nrc: string[] | any = await this.Materias.map( (materia: any) => materia.nrc );
41     let carrera: string[] | any = await this.Materias.map( (materia: any) => materia.licenciatura );
42
43     for (let a = 0; a < nrc.length - 1; a++) { this.cantidad_alumnos[a] =
44       await this._getCantidadLista.getCantidadListaAsistencia( nrc[a], carrera[a] ); }
45
46   }
47
48   async obtener_nrcMaterias(Token: string) {
49     this.nrc$ = await new Promise((resolve, reject) => { this._getMateriasCasosUso.getNRCMaterias(Token).subscribe(
50       (Resp: any) => { resolve(Resp); },
51       (error: any) => { reject(error); }
52     ); });
53
54   }
55
56   async obtener_Materias(materia: any) {
57     this.materias$ = await new Promise((resolve, reject) => { this._getMateriasCasosUso.getMateriasAll(materia).subscribe(
58       (Resp: any) => { resolve(Resp); },
59       (error: any) => { reject(error); }
60     ); });
61
62   }
63
64   async obtener_Materias(materia: any) {
65     this.materias$ = await new Promise((resolve, reject) => { this._getMateriasCasosUso.getMateriasAll(materia).subscribe(
66       (Resp: any) => { resolve(Resp); },
67       (error: any) => { reject(error); }
68     ); });
69
70   }

```

Almacenar los datos de los archivos Excel recibidos en la nube firestore, asignando el nrc de la materia deseada, y los objetos siendo almacenados con el número de matrícula como su index para un mejor control de datos



```

1 GuardarDatosEnFirestore(datos: any[]) {
2   const db = firebase.firestore();
3   constleccion = db.collection(`ISW/Materias/${this.nrcMateria}`);
4
5   for (let i = 2; i < datos.length; i++) {
6     const datos_lista: any = {
7       Matricula: datos[i][0],
8       Nombre: datos[i][1],
9       Status: datos[i][2],
10      Carrera: datos[i][3],
11    };
12
13    const docRef = coleccion.doc(datos_lista.Matricula);
14    docRef
15      .set(datos_lista)
16      .then(() => {
17        console.log('Datos guardados correctamente');
18      })
19      .catch((error) => {
20        console.error('Error al guardar los datos:', error);
21      });
22  }
23

```

Obtener datos de la lista de asistencia del Excel subido por el docente, mediante al ejemplo recibido por un docente, se pudo realizar un control sobre el patrón del archivo, pudiendo extraer los datos mediante ubicaciones específicas de las columnas.

```
RecibirArchivo(file: File) {
    const archivo: any = new FileReader();
    archivo.onload = (e: any) => {
        const leer_Archivo: any = new Uint8Array(e.target.result);
        const acceder_Datos: any = XLSX.read(leer_Archivo, { type: 'array' });
        const acceder_HojaArchivo: any =
            acceder_Datos.Sheets[acceder_Datos.SheetNames[0]];

        console.log(acceder_HojaArchivo);

        const Matricula: any = XLSX.utils.sheet_to_json(acceder_HojaArchivo, {
            range: 'D11:D36',
            header: 1,
        });
        const Nombre: any = XLSX.utils.sheet_to_json(acceder_HojaArchivo, {
            range: 'H11:H36',
            header: 1,
        });
        const Status: any = XLSX.utils.sheet_to_json(acceder_HojaArchivo, {
            range: 'O11:O36',
            header: 1,
        });
        const Carrera: any = XLSX.utils.sheet_to_json(acceder_HojaArchivo, {
            range: 'P11:P36',
            header: 1,
        });

        const Guardar_Datos: any[] = [];
        for (let i = 2; i < Matricula.length; i++) {
            const datos_lista: any[] = [
                Matricula[i][0],
                Nombre[i][0],
                Status[i][0],
                Carrera[i][0],
            ];
            .
        }
    }
}
```

Si el sistema detecta que no hay una lista de asistencia almacenada en la nube, mostrará un apartado para subir una lista de asistencia, en cambio sí se detecta una lista, se mostrará y desplegará en forma de tabla. Cuando el docente cargue un archivo Excel en la página, se mostrarán los datos de los alumnos para su verificación

```
1 <form *ngIf="!listaAsistencia || listaAsistencia.length == 0">
2   <input type="file" (change)="Archivo($event)" accept=".xlsx, .xls">
3   <button >Cargar archivo</button>
4 </form>
5
6 <div *ngIf="vistaPreviaArchivo">
7   <h2>Previsualización del archivo:</h2>
8   <p>{{ vistaPreviaArchivo.name }}</p>
9
10 <div *ngIf="jsonData && jsonData.length > 0">
11   <table class="excel-table">
12     <thead>
13       <tr>
14         <th *ngFor="let header of jsonData[0]" [ngStyle]="{ width: 100 / jsonData[0].length + '%' }">{{ header }}</th>
15       </tr>
16     </thead>
17     <tbody>
18       <tr *ngFor="let row of jsonData.slice(1)">
19         <td *ngFor="let cell of row" [ngStyle]="{ width: 100 / jsonData[0].length + '%' }">
20           <div class="cell-content">{{ cell }}</div>
21           You, yesterday • cambios ...
22         </td>
23       </tr>
24     </tbody>
25   </table>
26 </div>
27
28 <table class="table table-striped table-hover" *ngIf="listaAsistencia && listaAsistencia.length > 0">
29   <thead>
30     <h2>Lista de Asistencia</h2>
31     <tr>
32       <th>Índice</th>
33       <th>Alumno</th>
34       <th>Matrícula</th>
```

El sistema primero verificará que los datos ingresados en el Login sean válidos, si es correcto se generará un Token y se habilitará la visibilidad de los demás componentes a través de un Observable

```
async login(usuario: string, contraseña: string) {
    let response$;
    this.responseSuccessful = false;

    response$ = await this._IniciarSesion.postLogin(usuario, contraseña).toPromise();

    try {
        const Resp:any = await response$;
        this.datosLocales.guardar_DatoLocal('Resp', Resp.token);
        this.responseSuccessful = true;
    } catch (error) {
        this.responseSuccessful = false;
    }

    return this.responseSuccessful;
}

You, last week • cambios

async IniciarSesion() {
    const loginSuccessful = await this.login(this.username, this.password);
    if (loginSuccessful) {
        this.datosLocales.Actualizar_Login(true);
        this.datosLocales.guardar_DatoLocal('login', true);
        this.router.navigate(['/Sistema/Inicio/']);
    } else {
        this.datosLocales.Actualizar_Login(false);
        this.loginFailed = true;
        this.loggedIn = false;
    }
}
```

Al realizar el registro de cuentas, primero se subirá la imagen seleccionada a la nube, obteniendo el url para poder visualizarla después

```
async SubirImagenFirestore() {
  if (this.file) {
    const filePath = `images/${this.file.name}`;
    const fileRef = this.storage.ref(filePath);
    try {
      await this.storage.upload(filePath, this.file);
      const downloadUrl = await fileRef.getDownloadURL().toPromise();
      this.imageURL = downloadUrl;
    } catch (error) {
      console.error('Error uploading image:', error);
    }
  }
}
```

Se utilizó el caché del navegador para almacenar los datos necesarios para el funcionamiento del sistema

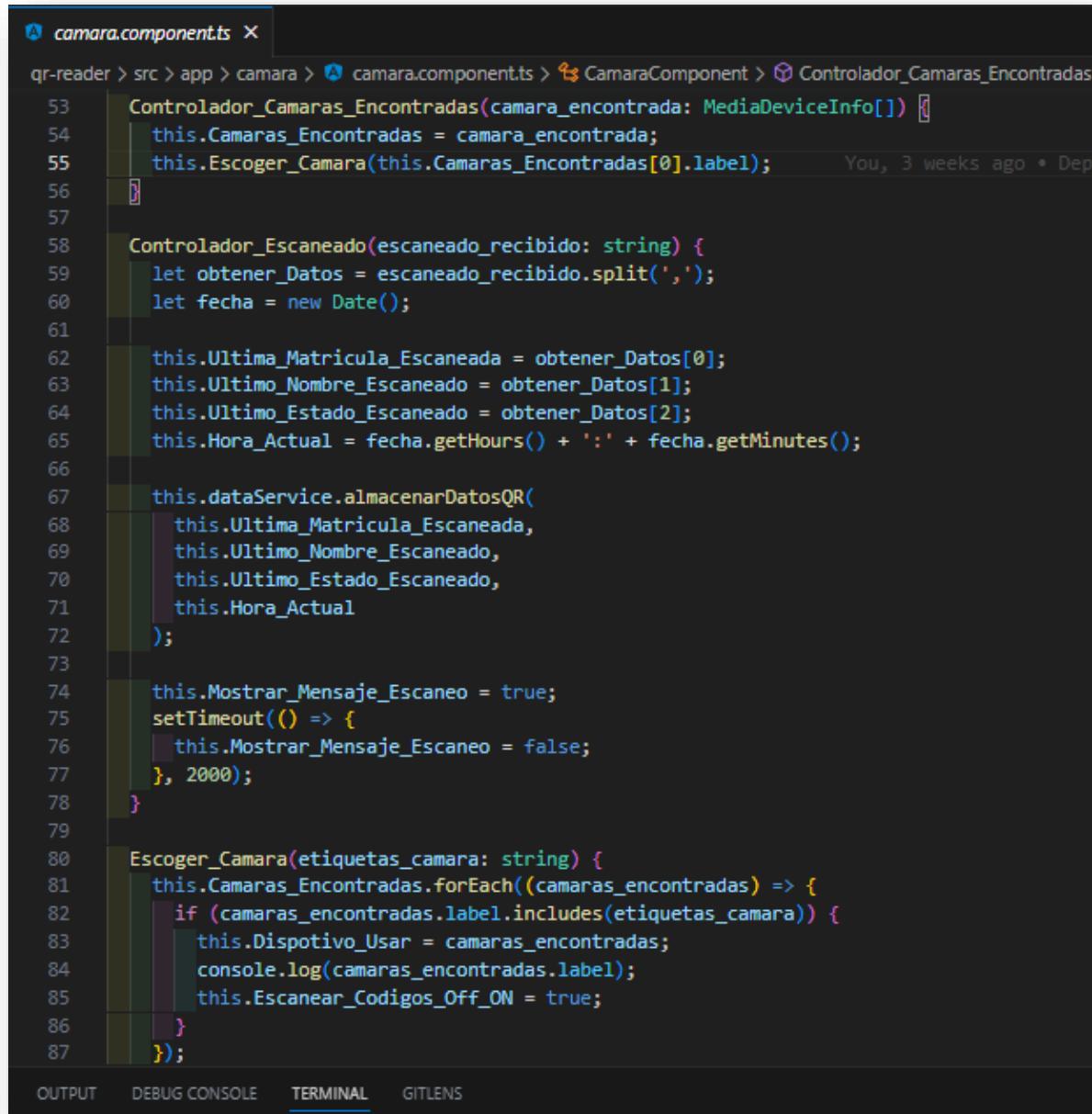
```
20 ) { }
21
22 obtener_DatoLocal(indice: string): any {
23   return localStorage.getItem(indice);
24 }
25 guardar_DatoLocal(indice: string, valor: any): void {
26   localStorage.setItem(indice, valor);
27   console.log(valor);
28 }
29
30 eliminar_DatoLocal(indice: string): void {
31   localStorage.removeItem(indice);
32 }
33
34 Actualizar_Login(loggedIn: boolean) {
35   this.loggedInSubject.next(loggedIn);
36 }
37
38 Actualizar_Docente(docente: string | number) {
39   this.docenteSubject.next(docente);
40 }
41
42 Actualizar_Formulario(formulario: string | any) {
43   this.formularioSubject.next(formulario);
44 }
45
46 eliminarCacheNavegador() {
47   if (caches && caches.keys) [
48     caches.keys().then(function (keys) {
49       keys.forEach(function (key) {
50         caches.delete(key);
51       });
52     });
53   ]
54 }
```

You, yesterday • cambios ...

OUTPUT DEBUG CONSOLE TERMINAL GITLENS

Front End Simulador Escaner

Código para escoger la cámara por defualt del dispositivo, además la habilitará para su utilización en la lectura de códigos QR



The screenshot shows a code editor window with the file `camara.component.ts` open. The code is written in TypeScript and handles QR code scanning logic. It includes methods for selecting a camera, handling scanned data, and saving data to a service. The code uses variables like `CamaraComponent`, `Controlador_Camaras_Encontradas`, `Ultima_Matricula_Escaneada`, `Ultimo_Nombre_Escaneado`, `Ultimo_Estado_Escaneado`, and `Hora_Actual`. It also includes a timeout mechanism for message display. The code is annotated with comments and includes imports from `qr-reader` and `DataService`.

```
camara.component.ts
qr-reader > src > app > camara > camara.component.ts > CamaraComponent > Controlador_Camaras_Encontradas
53     Controlador_Camaras_Encontradas(camara_encontrada: MediaDeviceInfo[]) {
54         this.Camaras_Encontradas = camara_encontrada;
55         this.Escoger_Camara(this.Camaras_Encontradas[0].label);
56     }
57
58     Controlador_Escaneado(escaneado_recibido: string) {
59         let obtener_Datos = escaneado_recibido.split(',');
60         let fecha = new Date();
61
62         this.Ultima_Matricula_Escaneada = obtener_Datos[0];
63         this.Ultimo_Nombre_Escaneado = obtener_Datos[1];
64         this.Ultimo_Estado_Escaneado = obtener_Datos[2];
65         this.Hora_Actual = fecha.getHours() + ':' + fecha.getMinutes();
66
67         this.dataService.almacenarDatosQR(
68             this.Ultima_Matricula_Escaneada,
69             this.Ultimo_Nombre_Escaneado,
70             this.Ultimo_Estado_Escaneado,
71             this.Hora_Actual
72         );
73
74         this.Mostrar_Mensaje_Escaneo = true;
75         setTimeout(() => {
76             this.Mostrar_Mensaje_Escaneo = false;
77         }, 2000);
78     }
79
80     Escoger_Camara(etiquetas_camara: string) {
81         this.Camaras_Encontradas.forEach((camaras_encontradas) => {
82             if (camaras_encontradas.label.includes(etiquetas_camara)) {
83                 this.Dispositivo_Urar = camaras_encontradas;
84                 console.log(camaras_encontradas.label);
85                 this.Escaneear_Codigos_Off_ON = true;
86             }
87         });
88     }
89
90     Escaneear_Codigos() {
91         if (this.Escaneear_Codigos_Off_ON) {
92             this.camara.open();
93             this.camara.start();
94         } else {
95             this.camara.stop();
96             this.camara.close();
97         }
98     }
99
100    Mostrar_Mensaje_Escaneo() {
101        if (this.Mostrar_Mensaje_Escaneo) {
102            alert('QR code escaneado');
103        }
104    }
105
106    Dispositivo_Urar(label: string) {
107        this.camara.setDevice(label);
108    }
109
110    Escaneear_Codigos_Off_ON: boolean = false;
111
112    Camaras_Encontradas: MediaDeviceInfo[] = [];
113
114    Ultima_Matricula_Escaneada: string = '';
115
116    Ultimo_Nombre_Escaneado: string = '';
117
118    Ultimo_Estado_Escaneado: string = '';
119
120    Hora_Actual: string = '';
121
122    Mostrar_Mensaje_Escaneo: boolean = false;
123
124    Dispositivo_Urar: MediaDeviceInfo | null = null;
```

OUTPUT DEBUG CONSOLE TERMINAL GITLENS

Verificar que los datos leídos a través del QR pertenezcan a la lista de asistencia en turno

```
listas.component.ts X
qr-reader > src > app > listas > listas.component.ts > ListasComponent
34     async ngOnInit() {
35         this.datos_locales
36             .Lista_Datos_QR_Observable()
37             .subscribe((nuevoValor: any) => {
38                 this.datosLeidos = JSON.parse(nuevoValor) || [];
39                 if (this.conexionService.getOnlineStatus().getValue()) {
40                     this.conexionService.enviarDatos();
41                     this.datos_locales.eliminarDatosAlFinalizarDia();
42                 }
43             });
44
45         const Obtener = this.datos_locales.obtener_DatoLocal('almacenarDatosQR');
46         this.datosLeidos = Obtener ? JSON.parse(Obtener) : [];
47
48         this.cargarDatos().then(() => {
49             this.datosCargados = true;
50         });
51     }
52
53     async cargarDatos() {
54         this.carrera = await this.firebaseioService.getCarrera();
55         this.nrcMateria = await this.firebaseioService.getNrcByHorario();
56         this.listaAsistencia = await this.firebaseioService.getListaAsistencia(
57             this.nrcMateria,
58             this.carrera
59         );
60     }
61
62     aparece_en_Lista(alumno_recibido: Estructura): boolean {
63         const buscar = this.listaAsistencia.find(
64             (buscar_coincidencia) =>
65                 buscar_coincidencia.Matricula === alumno_recibido.Matricula
66         );
67         return !!buscar;
68     }

```

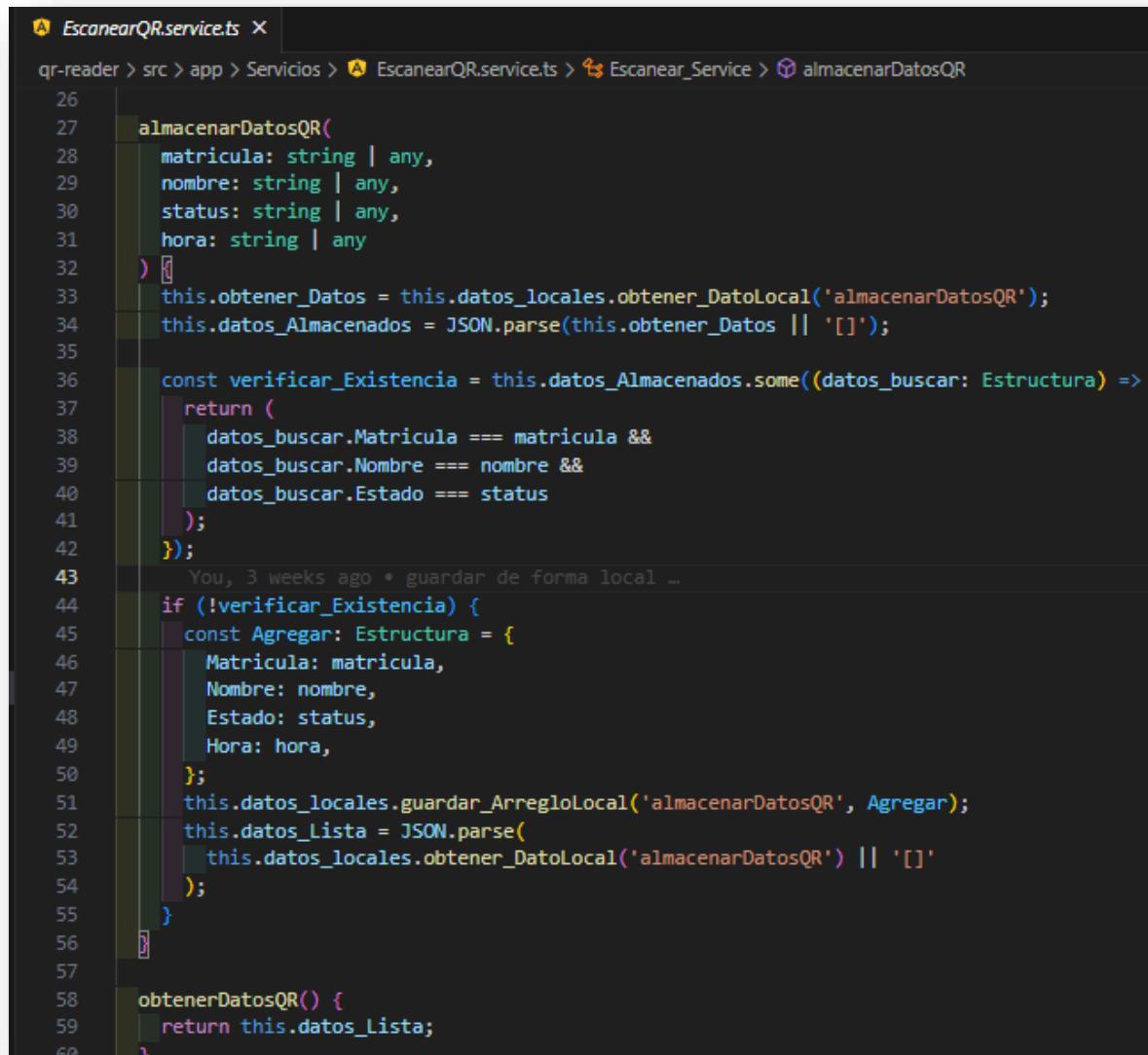
OUTPUT DEBUG CONSOLE TERMINAL GITLENS

Almacenar los datos de la toma de asistencia en tiempo real, para que pueda ser utilizado por el front-end de docentes

```
qr-reader > src > app > Servicios > Conexion.service.ts > ConexionService > verificarDatos > datosLeidos.filter() callback > buscar

41  async enviarDatos() {
42    let fecha = new Date();
43    let dia = fecha.getDate();
44    let mes = fecha.getMonth() + 1;
45    let año = fecha.getFullYear();
46    let fechaCompleta = año + ':' + mes + ':' + dia;
47
48    const listaAsistencia: any = await this.firestoreService.getListaAsistencia(
49      await this.nrcMateria,
50      await this.carrera
51    );
52    const datosAEviar = this.verificarDatos(listaAsistencia);
53
54    datosAEviar.forEach(async (dato: any) => {
55      const colección = this.firestore
56        .collection('/Registro/Asistencia/' + await this.nrcMateria + '/' + fechaCompleta + '/Alumnos')
57        .doc(dato.Matricula)
58        .set(dato)
59        .then(() => {
60          console.log('Datos guardados correctamente en Firestore');
61        })
62        .catch((error) => {
63          console.error('Error al guardar los datos en Firestore: ', error);
64        });
65    });
66  }
67
68  verificarDatos(lista: Estructura[]): Estructura[] {
69    const datosLeidos = JSON.parse(
70      localStorage.getItem('almacenarDatosQR') || '[]'
71    );
72
73    return datosLeidos.filter((dato: { Matricula: string }) => {
74      const buscar = lista.find(
75        (alumno) => alumno.Matricula === dato.Matricula
76      );
77
78      if (buscar) {
79        const index = lista.indexOf(buscar);
80        lista[index].Matricula = dato.Matricula;
81      }
82
83      return true;
84    });
85  }
86}
```

Evitar la duplicidad en los datos leídos por el escaner de códigos QR



```
 26
 27     almacenarDatosQR(
 28         matricula: string | any,
 29         nombre: string | any,
 30         status: string | any,
 31         hora: string | any
 32     ) {
 33         this.obtener_Datos = this.datos_locales.obtener_DatoLocal('almacenarDatosQR');
 34         this.datos_Almacenados = JSON.parse(this.obtener_Datos || '[]');
 35
 36         const verificar_Existencia = this.datos_Almacenados.some((datos_buscar: Estructura) => {
 37             return (
 38                 datos_buscar.Matricula === matricula &&
 39                 datos_buscar.Nombre === nombre &&
 40                 datos_buscar.Estado === status
 41             );
 42         });
 43         You, 3 weeks ago * guardar de forma local ...
 44         if (!verificar_Existencia) {
 45             const Agregar: Estructura = {
 46                 Matricula: matricula,
 47                 Nombre: nombre,
 48                 Estado: status,
 49                 Hora: hora,
 50             };
 51             this.datos_locales.guardar_ArregloLocal('almacenarDatosQR', Agregar);
 52             this.datos_Lista = JSON.parse(
 53                 this.datos_locales.obtener_DatoLocal('almacenarDatosQR') || '[]'
 54             );
 55         }
 56     }
 57
 58     obtenerDatosQR() {
 59         return this.datos_Lista;
 60     }
}
```

Front End Alumnos

Recibir datos para la generación de códigos QR

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** On the left, it shows the project structure under the 'ALUMNOS' folder. The 'codigo' component is selected, displaying its files: component.html, component.scss, and component.spec.ts.
- Code Editor:** The main area displays the 'codigo.component.ts' file. The code defines a component named 'CodigoComponent' that implements the 'OnInit' interface. It imports 'Component' from '@angular/core' and 'MandarDatosQR' from a service. The component has a selector of 'app-codigo' and a template URL of './codigo.component.html'. The style URL is defined as ['./codigo.component.scss']. The constructor injects 'mandar' of type 'MandarDatosQR'. The component's logic includes setting 'DatosQR' to an empty string, and then assigning values from 'mandar' to 'Alumno', 'Materia', and 'Matricula'.
- Terminal:** At the bottom, the terminal window shows the command 'emil@MSI MINGW64 ~/alumnos (master)' followed by a '\$' prompt.

Mostrar código QR para que el alumno pueda realizar el proceso de toma de asistencia automatizada

The screenshot shows the Visual Studio Code interface with the following details:

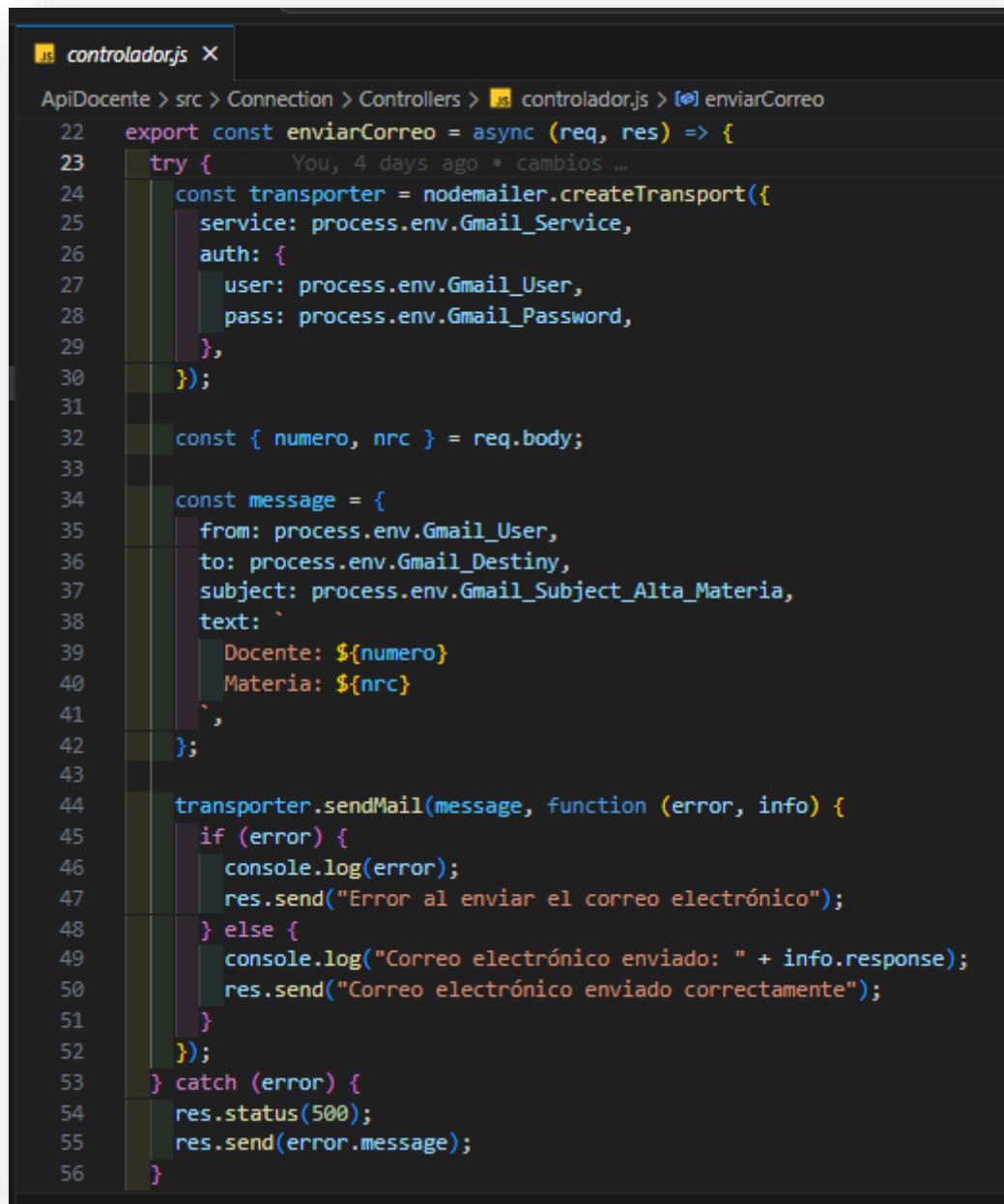
- File Explorer:** Shows the project structure with files like `codigo.component.html`, `codigo.component.spec.ts`, `codigo.component.ts`, `Cuenta.png`, `Escudo_Página.png`, `Formulario.jpg`, `header.component.html`, `header.component.ts`, `menu.component.html`, `menu.component.ts`, and `app-routing.module.ts`.
- Code Editor:** Displays the `codigo.component.html` file content:

```
<body>
  <div class="container">
    <qrcode [qrdata]="DatosQR" [width]="300" [errorCorrectionLevel]="'M'"></qrcode>
  </div>
  <p> Datos Recibidos: </p>
  <input type="text" [(ngModel)]="DatosQR">
  <br/>-- otra forma de mostrar los datos es: -->
  {{ Alumno }}
  {{ Materia }}
  {{ Matricula }}
</div>
</body>
```
- Terminal:** Shows the command line output:

```
esteban@PSI-MINION64:~/alumnos (master)
$
```
- Bottom Bar:** Includes icons for search, file operations, and extensions.

Api Docentes

Utilización del servicio de mensajería de GMAIL para poder enviar los correos electrónicos para el proceso de alta y baja de materias, utilizando un correo usuario que será perteneciente al front de docente, y un correo usuario destinatario que será el que reciba los datos para realizar dicho funcionamiento



```
controlador.js X
ApiDocente > src > Connection > Controllers > controlador.js > [e] enviarCorreo
22  export const enviarCorreo = async (req, res) => {
23    try {
24      You, 4 days ago • cambios ...
25      const transporter = nodemailer.createTransport({
26        service: process.env.Gmail_Service,
27        auth: {
28          user: process.env.Gmail_User,
29          pass: process.env.Gmail_Password,
30        },
31      );
32      const { numero, nrc } = req.body;
33
34      const message = {
35        from: process.env.Gmail_User,
36        to: process.env.Gmail_Destiny,
37        subject: process.env.Gmail_Subject_Alta_Materia,
38        text: `
39          Docente: ${numero}
40          Materia: ${nrc}
41        `,
42      };
43
44      transporter.sendMail(message, function (error, info) {
45        if (error) {
46          console.log(error);
47          res.send("Error al enviar el correo electrónico");
48        } else {
49          console.log("Correo electrónico enviado: " + info.response);
50          res.send("Correo electrónico enviado correctamente");
51        }
52      });
53    } catch (error) {
54      res.status(500);
55      res.send(error.message);
56    }
  
```

Verificar que la cuenta que se desea crear no exista ya en el sistema, además de que validará de que el número de personal coincida con algún docente registrado en la base de datos

```
65
70  export const AltaCuentaDocente = async (req, res) => {
71    try {
72      const errors = validationResult(req);
73      if (!errors.isEmpty()) {
74        return res.status(400).json({ errors: errors.array() });
75      }
76
77      const [Numero_Personal, Contraseña, Correo, URL_Imagen] = req.body;
78
79      const client = await pool.connect({
80        charset: "utf8mb4",
81      );
82
83      const existencia = await client.query(querys.DocenteExistencia, [
84        Numero_Personal,
85        ]);
86      const contar = await client.query(querys.CuentaExistente, [Correo]);
87      if ((existencia.rows[0].count == 1)) {
88        if (contar.rows[0].count == 0) {
89          await client.query(querys.InsertarCuentaDocente, [
90            Numero_Personal,
91            Contraseña,
92            Correo,
93            URL_Imagen,
94            ]);
95          res.json({ Numero_Personal, Contraseña, Correo, URL_Imagen });
96        } else {
97          return res.status(400).json("La cuenta ya existe!!!");
98        }
99      } else {
100        return res
101          .status(400)
102          .json("El numero de personal no esta registrado en el sistema");
103    }
104  }
```

2.8 Repositorio de la solución

2.8.1 URL

Repositorio	URL
Front-End Docentes	https://github.com/IrvingCM123/Sistema_Asistencia.git
Front-End Alumnos	https://github.com/EmilianoTorres123/alumnos-sistema.git
Front-End Simulador Escaner	https://github.com/IrvingCM123/Simulador_Escaner.git
Back-End Docentes	https://github.com/IrvingCM123/ApiNode.git
Back-End Alumnos	https://github.com/FerminDRA/backendAlumnos.git

2.9 Administración del proyecto

2.9.1 Roles que desempeño cada miembro del equipo

Alumno	Rol que desempeñó	Actividades
Irving Rafael Conde Marín	Líder de proyecto	Front-End Docentes Front-End Simulador Api Docentes Documentación Reporte Técnico
Aaron Emiliano Torres Nuñez	Sublíder de proyecto	Front-End Alumnos Documentación MockUps
Adriel Eduardo Peregrina Soto	Equipo de desarrollo	Front-End Alumnos Documentación
Aldo Torres Ramírez	Equipo de desarrollo	Gestión de nubes
Fermín Del Rosario Antonio	Equipo de desarrollo	Api Alumnos
Magdyel Aguilar Cid	Equipo de desarrollo	Gestión de nubes Documentación MockUps

2.9.2 Metodología empleada para el desarrollo de la solución

Como se utilizó la metodología Scrum, se tuvieron que hacer ciertos pasos para aplicarla en nuestro proyecto.

Front-End Docentes

Backlog del producto:

ID	EPICAS	FUNCIONALIDAD	RAZÓN	DIFICULTAD
001	Registro Usuario	Permitir crear cuentas nuevas	Para que el usuario pueda acceder al sistema	50,0
		Validar que los campos ingresados cumplan los requerimientos	Para evitar errores previstos en el registro de cuentas	20,0
		Almacenar imagen de usuario	Es requisito del cliente poder mostrar la imagen del docente	30,0
002	Inicio de Sesión	Validar que las credenciales sean correctas	Verificar que solo el personal docente pueda acceder al sistema	40,0
		Generar Token para las consultas a la Api	Cuestiones de seguridad	60,0
003	Interfaces funcionales y amigables	Mostrar información docente	Verificar que los datos almacenados en el sistema sean correctos	40,0
		Mostrar información materias	Mostrar las materias que imparte el docente, pudiendo acceder a su lista de asistencia	30,0
		Mostrar lista de asistencia	Consultar la lista de asistencia de la materia	30,0
004	Guardar Lista Asistencia	Subir lista de asistencia	Guardar en el sistema una lista de asistencia en caso de que no exista	70,0
05	Formulario	Alta Materias	Solicitar que el sistema asigne una nueva materia al docente	50,0
		Baja Materias	Solicitar que el sistema elimine una materia al docente	50,0
06	Listado	Observar toma de asistencia en tiempo real	Visualizar la toma de asistencia de los alumnos en tiempo real	80,0
07	Cerrar Sesión	Salir de la cuenta en el sistema	Evitar que otros docentes accedan a su cuenta	50,0

Backlog de sprints

ID	Épica	Descripción	Criterios de Aceptación	Dificultad	Estado	Responsable	Duración
001	Registro Usuarios	Realizar componente para crear cuenta	El componente funciona perfectamente	50,0	Terminado	Irving Rafael Conde Marín	2 días
	Registro Usuarios	Validar los campos ingresados	Los campos fueron validados correctamente	20,0	Terminado	Irving Rafael Conde Marín	1 día
	Registro Usuarios	Almacenar imagen de usuario	La imagen del usuario se almacena correctamente	30,0	Terminado	Irving Rafael Conde Marín	1 día
002	Inicio de Sesión	Validar que las credenciales sean correctas	Las credenciales son validadas correctamente	40,0	Terminado	Irving Rafael Conde Marín	3 días
	Inicio de Sesión	Generar Token para las consultas a la API	Se genera el token correctamente	60,0	Terminado	Irving Rafael Conde Marín	1 día
003	Interfaces funcionales y amigables	Mostrar información docente	La información se muestra correctamente	40,0	Terminado	Irving Rafael Conde Marín	2 días
	Interfaces funcionales y amigables	Mostrar información materias	Las materias se muestran correctamente	30,0	Terminado	Irving Rafael Conde Marín	2 días
	Interfaces funcionales y amigables	Mostrar lista de asistencia	La lista de asistencia se muestra correctamente	30,0	Terminado	Irving Rafael Conde Marín	3 días
004	Guardar Lista Asistencia	Subir lista de asistencia	La lista de asistencia se guarda correctamente	70,0	Terminado	Irving Rafael Conde Marín	4 días
005	Formulario Alta Materias	Alta Materias	Se asigna una nueva materia	50,0	Terminado	Irving Rafael Conde Marín	1 día
	Formulario Baja Materias	Baja Materias	Se elimina una materia	50,0	Terminado	Irving Rafael Conde Marín	1 día
006	Listado	Observar toma de asistencia en tiempo real	La toma de asistencia se visualiza en tiempo real	80,0	Terminado	Irving Rafael Conde Marín	4 días
007	Cerrar Sesión	Salir de la cuenta en el sistema	Se cierra la sesión correctamente	50,0	Terminado	Irving Rafael Conde Marín	1 día

Front-End Alumnos

Backlog del producto:

ID	EPICAS	FUNCIONALIDAD	RAZÓN	DIFICULTAD
001	Registro Usuario	Permitir crear cuentas nuevas	Para que el usuario pueda acceder al sistema	50,0
		Validar que los campos ingresados cumplan los requerimientos	Para evitar errores previstos en el registro de cuentas	20,0
		Almacenar imagen de usuario	Es requisito del cliente poder mostrar la imagen del docente	30,0
002	Inicio de Sesión	Validar que las credenciales sean correctas	Verificar que solo el personal docente pueda acceder al sistema	40,0
		Generar Token para las consultas a la Api	Cuestiones de seguridad	60,0
003	Interfaces funcionales y amigables	Mostrar información alumno	Verificar que los datos almacenados en el sistema sean correctos	40,0
		Mostrar información materias	Mostrar las materias que imparte el docente, pudiendo acceder a su lista de asistencia	30,0
004	Generar código	Genera código QR con información correspondiente	Función necesaria para realizar el proceso de toma de asistencia	70,0
005	Cerrar Sesión	Salir de la cuenta en el sistema	Evitar que otros docentes accedan a su cuenta	50,0

Backlog de sprints

ID	ÉPICA	DESCRIPCIÓN	CRITERIOS DE ACEPTACIÓN	DIFICULTAD	ESTADO	RESPONSABLE	DURACIÓN
001	Registro Usuario	Permitir crear cuentas nuevas	Los usuarios pueden crear cuentas y acceder al sistema	50,0	En proceso	Aaron Torres Nuñez Adriel Eduardo Peregrina Soto	2 días
		Validar que los campos ingresados cumplen los requerimientos	Los campos ingresados en el registro cumplen los requerimientos	20,0	En proceso	Aaron Torres Nuñez Adriel Eduardo Peregrina Soto	1 día
		Almacenar imagen de usuario	La imagen de usuario se almacena correctamente	30,0	En proceso	Aaron Torres Nuñez Adriel Eduardo Peregrina Soto	1 día
002	Inicio de Sesión	Validar que las credenciales sean correctas	Solo el personal docente puede acceder al sistema con las credenciales correctas	40,0	En proceso	Aaron Torres Nuñez Adriel Eduardo Peregrina Soto	3 días
		Generar Token para las consultas a la API	Se genera correctamente un token para las consultas a la API	60,0	En proceso	Aaron Torres Nuñez Adriel Eduardo Peregrina Soto	1 día
003	Interfaces funcionales y amigables	Mostrar información alumno	La información del alumno se muestra correctamente	40,0	Terminado	Aaron Torres Nuñez Adriel Eduardo Peregrina Soto	2 días
		Mostrar información materias	Las materias que imparte el docente se muestran correctamente, permitiendo acceder a la lista de asistencia	30,0	Terminado	Aaron Torres Nuñez Adriel Eduardo Peregrina Soto	2 días
004	Generar código QR	Generar código QR con información correspondiente	Se genera correctamente un código QR con la información correspondiente para la toma de asistencia	70,0	Terminado	Aaron Torres Nuñez Adriel Eduardo Peregrina Soto	4 días
005	Cerrar Sesión	Salir de la cuenta en el sistema	Se cierra la sesión correctamente y se evita el acceso no autorizado a la cuenta del alumno	50,0	Terminado	Aaron Torres Nuñez Adriel Eduardo Peregrina Soto	1 día

Front-End Simulador QR

Backlog del producto:

ID	EPICAS	FUNCIONALIDAD	RAZÓN	DIFICULTAD
001	Interfaces funcionales y amigables	Mostrar previalización de cámara	Mostrarle al usuario si debe alinear el QR para que se pueda escanear correctamente	90,0
		Mostrar historial de códigos QR	Mostrar la información de los códigos QR escaneados previamente	50,0
		Mostrar apartado de configuración	Permitir al usuario configurar el equipo con los datos requeridos	40,0
		Mostrar información de listas de asistencias	Mostrar la lista de alumnos de la materia en turno	60,0
		Mostrar información de materias en turno	Mostrar la información de las materias en turno con la configuración establecida	60,0
002	Cámara	Prender Cámara	Encender la cámara del equipo para el proceso de la toma de asistencia	70,0
		Apagar Cámara	Apagar la cámara para evitar que se sigan registrando códigos QR	70,0

Backlog de sprints

ID	ÉPICA	DESCRIPCIÓN	CRITERIOS DE ACEPTACIÓN	DIFICULTAD	ESTADO	RESPONSABLE	DURACIÓN
001	Interfaces funcionales y amigables	Mostrar previsualización de cámara	Se muestra una previsualización de la cámara para alinear correctamente el código QR	90,0	Terminado	Irving Rafael Conde Marín	3 días
		Mostrar historial de códigos QR	Se muestra correctamente la información de los códigos QR escaneados previamente	50,0	Terminado	Irving Rafael Conde Marín	2 días
		Mostrar apartado de configuración	El usuario puede configurar el equipo con los datos requeridos	40,0	Terminado	Irving Rafael Conde Marín	2 días
		Mostrar información de listas de asistencias	Se muestra correctamente la lista de alumnos de la materia en turno	60,0	Terminado	Irving Rafael Conde Marín	2 días
		Mostrar información de materias en turno	Se muestra correctamente la información de las materias en turno con la configuración establecida	60,0	Terminado	Irving Rafael Conde Marín	2 días
002	Cámara	Prender Cámara	La cámara del equipo se enciende para el proceso de toma de asistencia	70,0	Terminado	Irving Rafael Conde Marín	3 días
		Apagar Cámara	La cámara del equipo se apaga para evitar que se sigan registrando códigos QR	70,0	Terminado	Irving Rafael Conde Marín	3 días
003	Lectura	Leer códigos QR	El sistema será capaz de detectar QR y leerlos para el proceso de asistencia	80,0	Terminado	Irving Rafael Conde Marín	5 días

Back-End Docentes

Backlog del producto:

ID	EPICAS	FUNCIONALIDAD	RAZÓN	DIFICULTAD
001	Registro Usuario	Permitir crear cuentas nuevas Validar que los campos ingresados cumplan los requerimientos	Para que el usuario pueda acceder al sistema Para evitar errores previstos en el registro de cuentas	50,0
002	Inicio de Sesión	Validar que las credenciales sean correctas	Verificar que solo el personal docente pueda acceder al sistema	40,0
003	Formulario	Generar Token Enviar correos Gmail para alta materia Enviar correos Gmail para baja materia	Cuestiones de seguridad Solicitar que el sistema asigne una nueva materia al docente Solicitar que el sistema elimine una materia al docente	60,0 60,0 60,0
004	Consultas	Acceder a la base de datos para solicitar la información deseada por el front	Visualizar la información que desee el usuario	80,0

Backlog de sprints

ID	ÉPICA	DESCRIPCIÓN	CRITERIOS DE ACEPTACIÓN	DIFICULTAD	ESTADO	RESPONSABLE	DURACIÓN
001	Registro Usuario	Permitir crear cuentas nuevas	Los usuarios pueden crear cuentas y acceder al sistema	50,0	Terminado	Irving Rafael Conde Marín	2 días
		Validar que los campos ingresados cumplan los requerimientos	Los campos ingresados en el registro cumplen los requerimientos	50,0	Terminado	Irving Rafael Conde Marín	2 días
002	Inicio de Sesión	Validar que las credenciales sean correctas	Solo el personal docente puede acceder al sistema con las credenciales correctas	40,0	Terminado	Irving Rafael Conde Marín	2 días
		Generar Token	Se genera correctamente un token para la autenticación	60,0	Terminado	Irving Rafael Conde Marín	1 día
003	Formulario	Enviar correos Gmail para alta materia	Se envían correos de confirmación al docente para el alta de una nueva materia	60,0	Terminado	Irving Rafael Conde Marín	3 días
		Enviar correos Gmail para baja materia	Se envían correos de confirmación al docente para la baja de una materia	60,0	Terminado	Irving Rafael Conde Marín	3 días
004	Consultas	Acceder a la base de datos para solicitar la información deseada por el front	Se puede visualizar correctamente la información solicitada por el usuario	80,0	Terminado	Irving Rafael Conde Marín	6 días

Back-End Alumnos

Backlog del producto:

ID	EPICAS	FUNCIONALIDAD	RAZÓN	DIFICULTAD
001	Registro Usuario	Permitir crear cuentas nuevas	Para que el usuario pueda acceder al sistema	50,0
		Validar que los campos ingresados cumplan los requerimientos	Para evitar errores previstos en el registro de cuentas	50,0
002	Inicio de Sesión	Validar que las credenciales sean correctas	Verificar que solo el personal docente pueda acceder al sistema	40,0
		Generar Token	Cuestiones de seguridad	60,0
003	Consultas	Acceder a la base de datos para solicitar la información deseada por el front	Visualizar la información que desee el usuario	80,0

Backlog de sprints

ID	ÉPICA	DESCRIPCIÓN	CRITERIOS DE ACEPTACIÓN	DIFÍCULTAD	ESTADO	RESPONSABLE	DURACIÓN
001	Registro Usuario	Permitir crear cuentas nuevas	Los usuarios pueden crear cuentas y acceder al sistema	50,0	Terminado	Fermín Del Rosario Antonio	2 días
		Validar que los campos ingresados cumplen los requerimientos	Los campos ingresados en el registro cumplen los requerimientos	50,0	Terminado	Fermín Del Rosario Antonio	2 días
002	Inicio de Sesión	Validar que las credenciales sean correctas	Solo el personal docente puede acceder al sistema con las credenciales correctas	40,0	Terminado	Fermín Del Rosario Antonio	2 días
		Generar Token	Se genera correctamente un token para la autenticación	60,0	Terminado	Fermín Del Rosario Antonio	1 día
003	Consultas	Acceder a la base de datos para solicitar la información deseada por el front	Se puede visualizar correctamente la información solicitada por el usuario	80,0	Terminado	Fermín Del Rosario Antonio	6 días

Instancia Google Cloud

Backlog de producto

ID	EPICAS	FUNCIONALIDAD	RAZÓN	DIFICULTAD
001	Configuración de cuenta	Meter las credenciales pertenecientes a la cuenta de Google Cloud	Es necesario loguearse para ocupar Google Cloud	20,0
		Solicitar créditos gratuitos para ocupar Google Cloud	Para ocupar instancias Google Cloud es necesario pagar.	50,0
002	Instancia	Configurar instancia Google Cloud	Es necesaria la configuración para crear una instancia	70,0
		Desplegar instancia	Para utilizar la instancia es necesario desplegarla	60,0
003	Base de datos	Configurar Postgres	Es necesario configurar postgres para el funcionamiento del sistema	60,0

Backlog de sprint

ID	ÉPICA	DESCRIPCIÓN	CRITERIOS DE ACEPTACIÓN	DIFÍCULTAD	ESTADO	RESPONSABLE	DURACIÓN
001	Configuración de cuenta	Meter las credenciales pertenecientes a la cuenta de Google Cloud	Las credenciales de la cuenta de Google Cloud se integran correctamente en el sistema	20,0	Terminado	Aldo Ramirez Magdely Cid	1 día
		Solicitar créditos gratuitos para ocupar Google Cloud	Se realiza correctamente la solicitud de créditos gratuitos para poder utilizar instancias en Google Cloud	50,0	Terminado	Aldo Ramirez Magdely Cid	2 días
002	Instancia	Configurar instancia Google Cloud	La instancia de Google Cloud se configura correctamente para su funcionamiento	70,0	Pendiente	Aldo Ramirez Magdely Cid	5 días
		Desplegar instancia	La instancia de Google Cloud se despliega correctamente y está lista para ser utilizada	60,0	Pendiente	Aldo Ramirez Magdely Cid	4 días
003	Base de datos	Configurar Postgres	Postgres se configura correctamente y está listo para su uso	60,0	Pendiente	Aldo Ramirez Magdely Cid	4 días

Nube Firebase

Backlog del producto

ID	ÉPICAS	FUNCIONALIDAD	RAZÓN	DIFICULTAD
001	Configuración de cuenta	Meter las credenciales pertenecientes a la cuenta de Google	Es necesario iniciar sesión con Google para ocupar Firebase	20,0
		Crear proyecto nuevo	Para ocupar los funcionamientos de Firebase es necesario crear un nuevo proyecto	20,0
002	Firebase	Configurar instancia Firestore	Es necesaria la configuración para crear una instancia firestore y poder almacenar datos	40,0
		Configurar instancia Storage	Es necesaria la configuración para crear una instancia storage y poder almacenar imágenes	40,0
003	Reglas	Configurar reglas	Es necesario configurar las reglas para saber de qué URL aceptar las peticiones	40,0

Backlog de sprints

ID	ÉPICA	DESCRIPCIÓN	CRITERIOS DE ACEPTACIÓN	DIFICULTAD	ESTADO	RESPONSABLE	DURACIÓN
001	Ingresar datos de cuenta	Meter las credenciales pertenecientes a la cuenta de Google	Las credenciales de la cuenta de Google se integran correctamente con el sistema y permiten iniciar sesión	20,0	Terminado	Irving Rafael Conde Marín	1 día
		Crear proyecto nuevo	Se crea correctamente un nuevo proyecto en Firebase que se utilizará para el funcionamiento del sistema	20,0	Terminado	Irving Rafael Conde Marín	1 día
002	Firebase	Configurar instancia Firestore	Se configura correctamente una instancia de Firestore para poder almacenar datos en el sistema	40,0	Terminado	Irving Rafael Conde Marín	1 día
		Configurar instancia Storage	Se configura correctamente una instancia de Storage en Firebase para poder almacenar imágenes en el sistema	40,0	Terminado	Irving Rafael Conde Marín	1 día
003	Reglas	Configurar reglas	Se configuran correctamente las reglas en Firebase para aceptar las peticiones desde las URL correspondientes	40,0	Terminado	Irving Rafael Conde Marín	1 día

2.9.3 Evidencias de la gestión del proyecto

Gestión de ambientes de producción

Los códigos fueron desplegados para ser vistos en un ambiente real y notar detalles o errores presentes en ellos

Nota: La Api no ha podido ser desplegada, por lo que si intenta realizar el login no podrá debido a esto

The screenshot shows the Vercel dashboard with the following information:

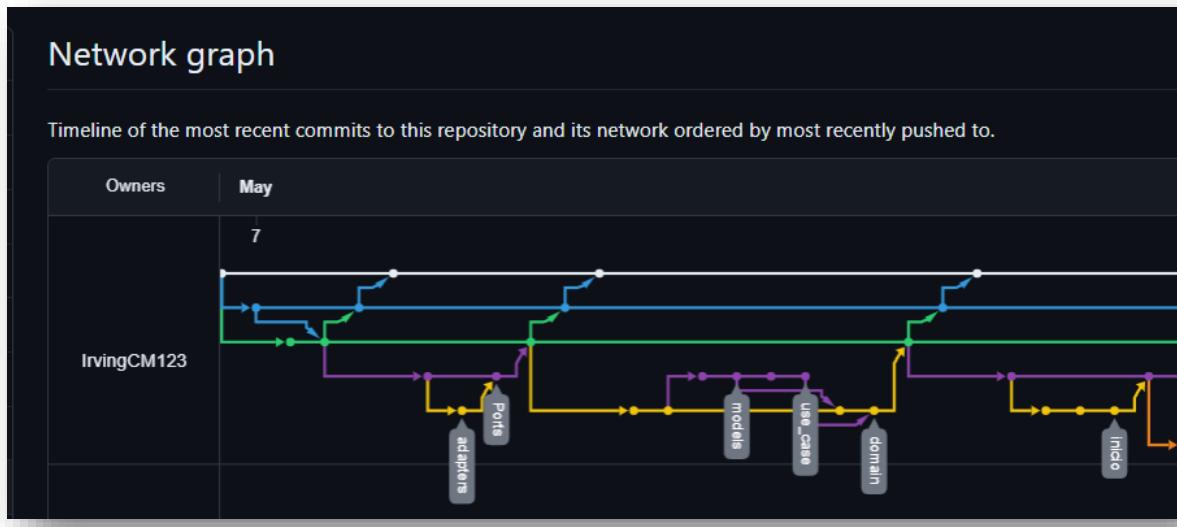
- sistema-asistencia**: sistema-asistencia.vercel.app
Merge pull request #71 from IrvingCM123/views
Views
From develop
2d ago via
- simulador-escaner**: simulador-escaner.vercel.app
Merge pull request #38 from IrvingCM123/release
Release
From master
2d ago via
- apidocente**: No Production Deployment
No Git Repository connected.
3d ago
- animes-graphql-react**: animes-graphql-react.vercel.app
Merge pull request #13 from IrvingCM123/release
Release
From master
35d ago via
- página-vercel**: página-vercel.vercel.app
Cambios
From main
268d ago via
- prueba**: prueba-omega-two.vercel.app
prueba
From main
285d ago via

The dashboard includes a search bar, a "Add New..." button, and navigation tabs for Overview, Integrations, Activity, Domains, Usage, Monitoring, Storage, and Settings.

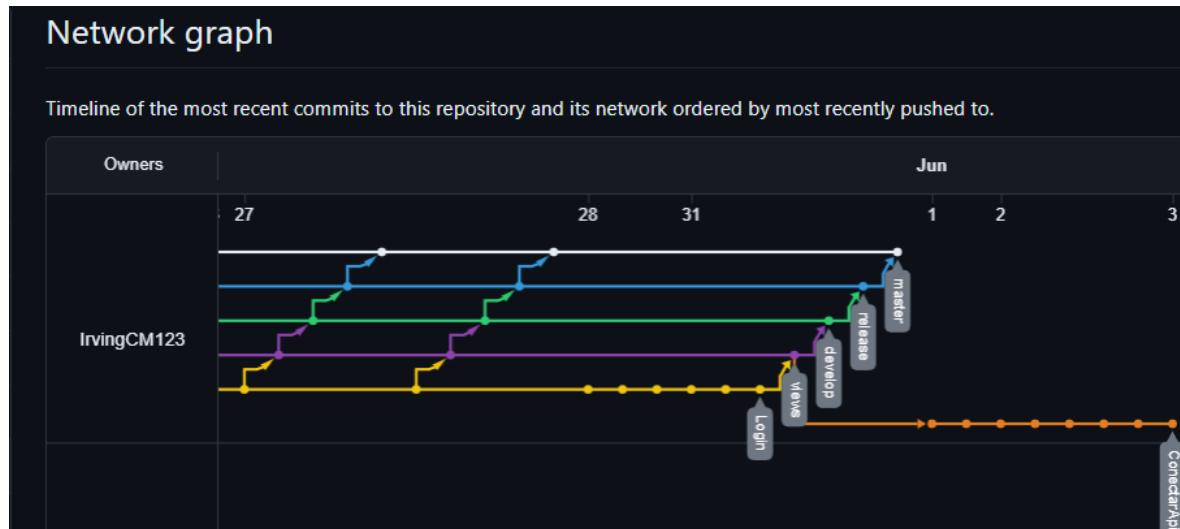
Gestión del front de Docentes

El código fue dividido en distintos Branch para llevar un mejor control de cambios y de errores.

El código fue subido por primera vez el 23 de Abril

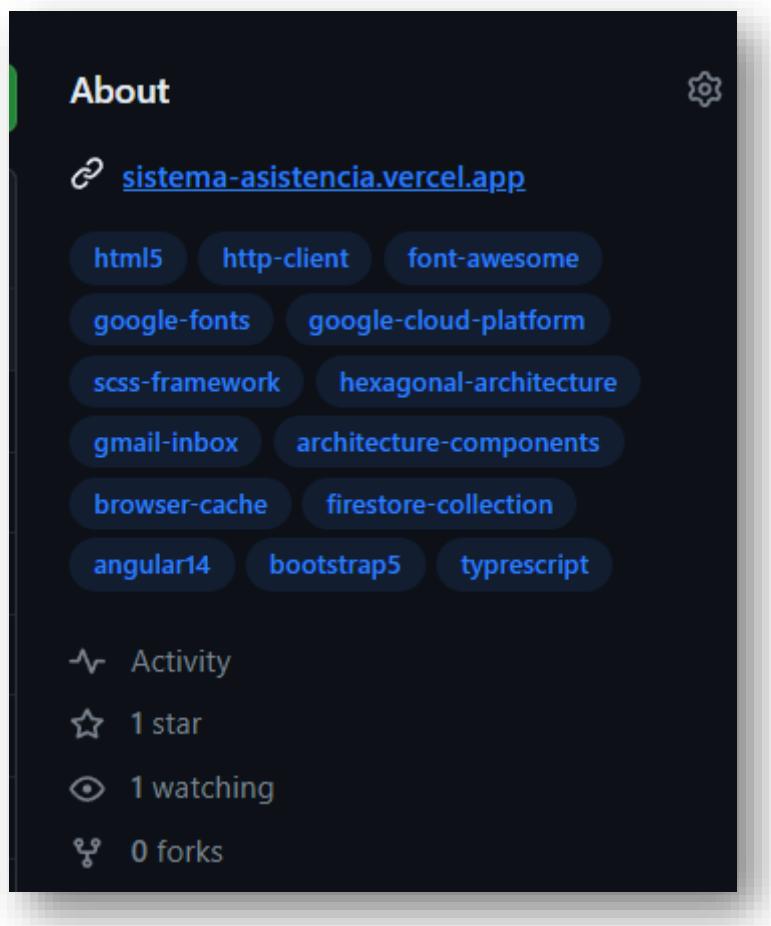


Y hasta el día de hoy se siguen subiendo cambios

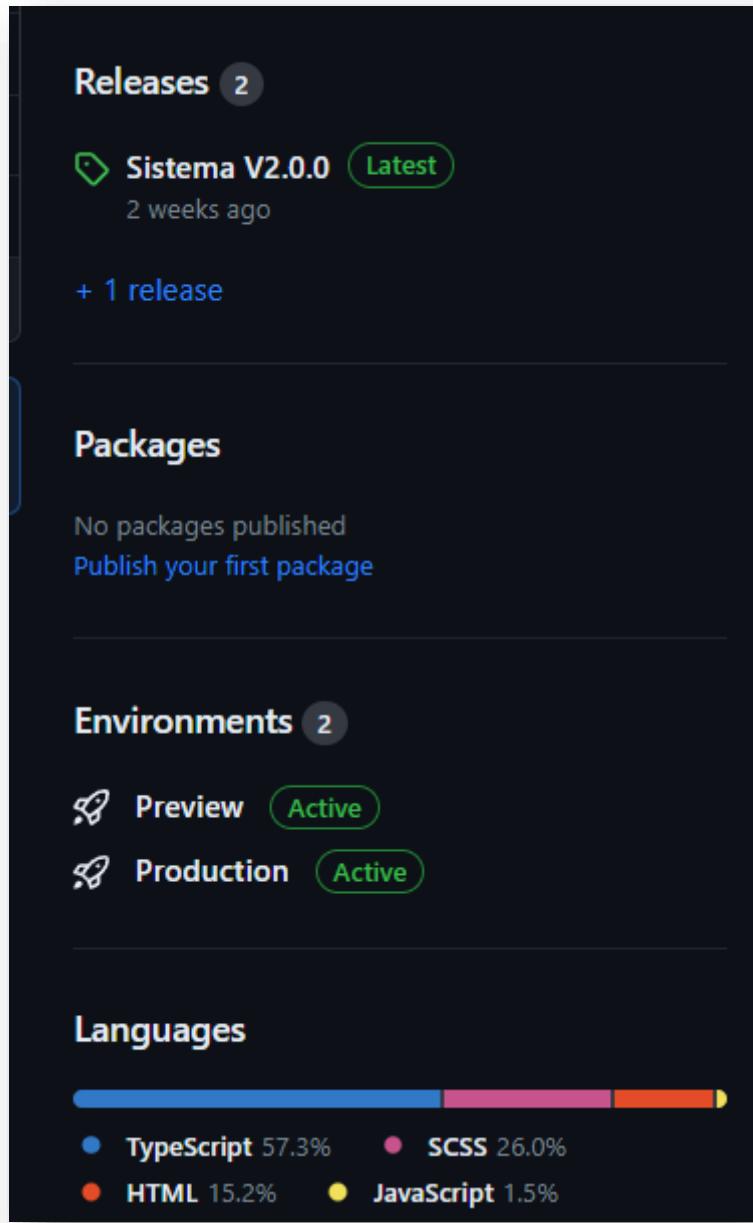


El Front fue desplegado en un servidor llamado Vercel, en el que está operativo con el Branch Develop para observar que los cambios y las funciones estén correctas en un ambiente de operación.

Nota: El despliegue no está completo, por lo que puede haber errores o faltar funciones si accede al url



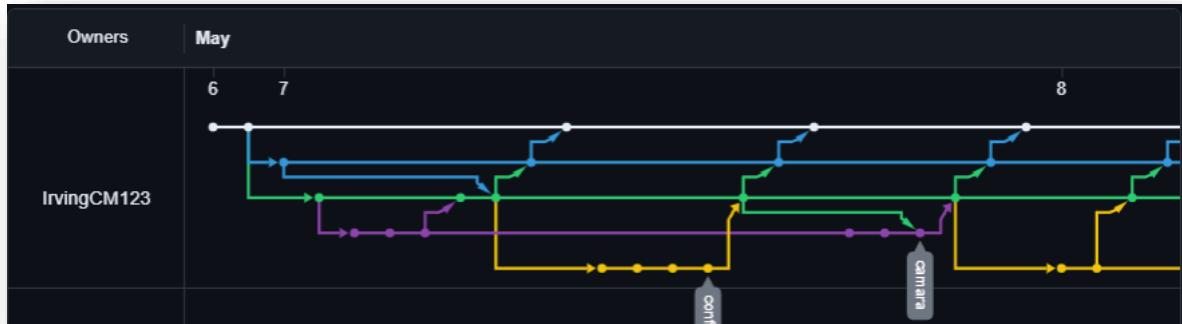
El repositorio cuenta con un manejo de versiones, actualmente solo tiene 2 versiones, y cuenta con un ambiente de prevializuación y un ambiente de producción



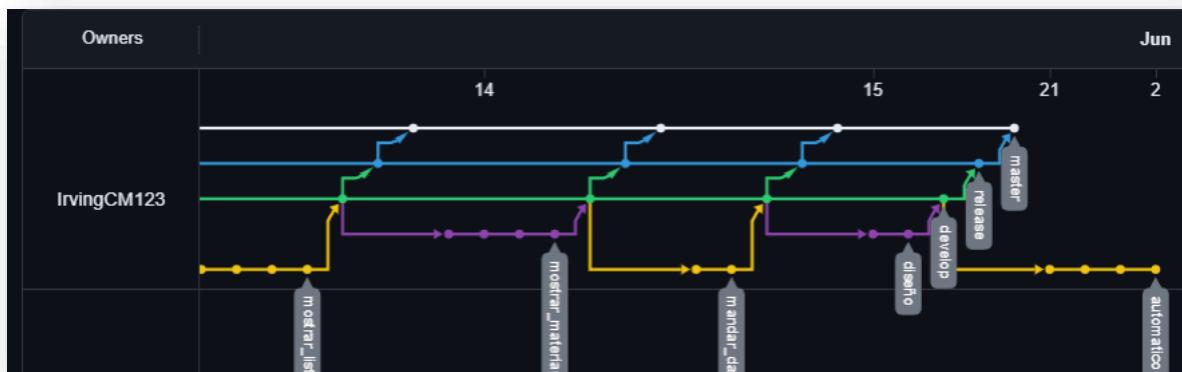
Gestión del front de Simulador

El código fuente fue dividido en distintos Branch para llevar un mejor control de cambios y de errores.

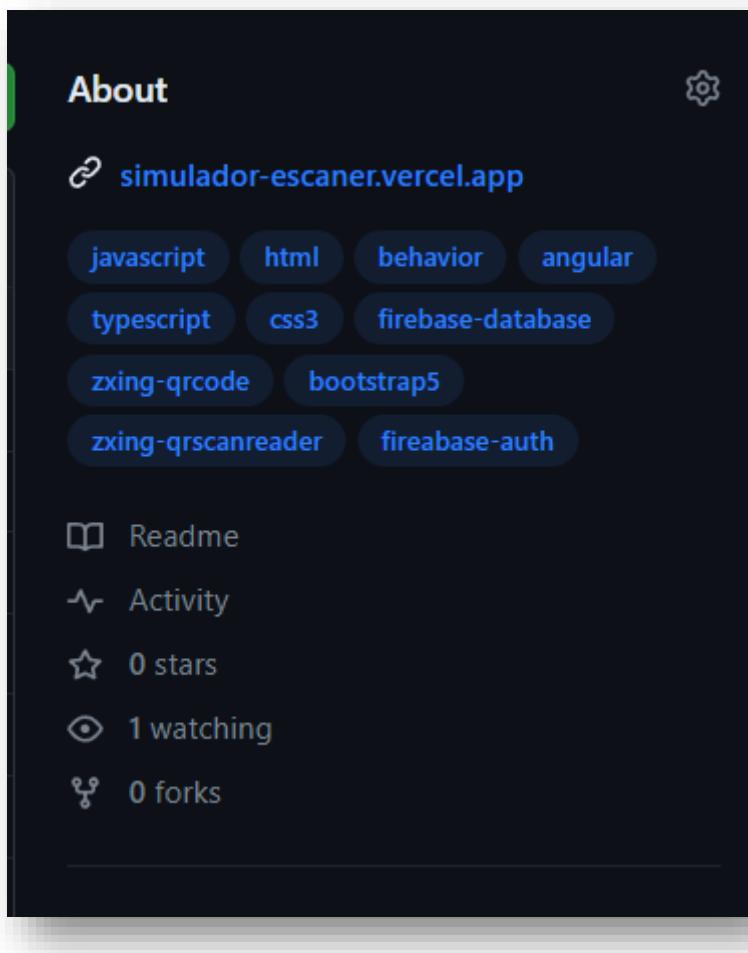
El código fue subido por primera vez el 6 de Mayo



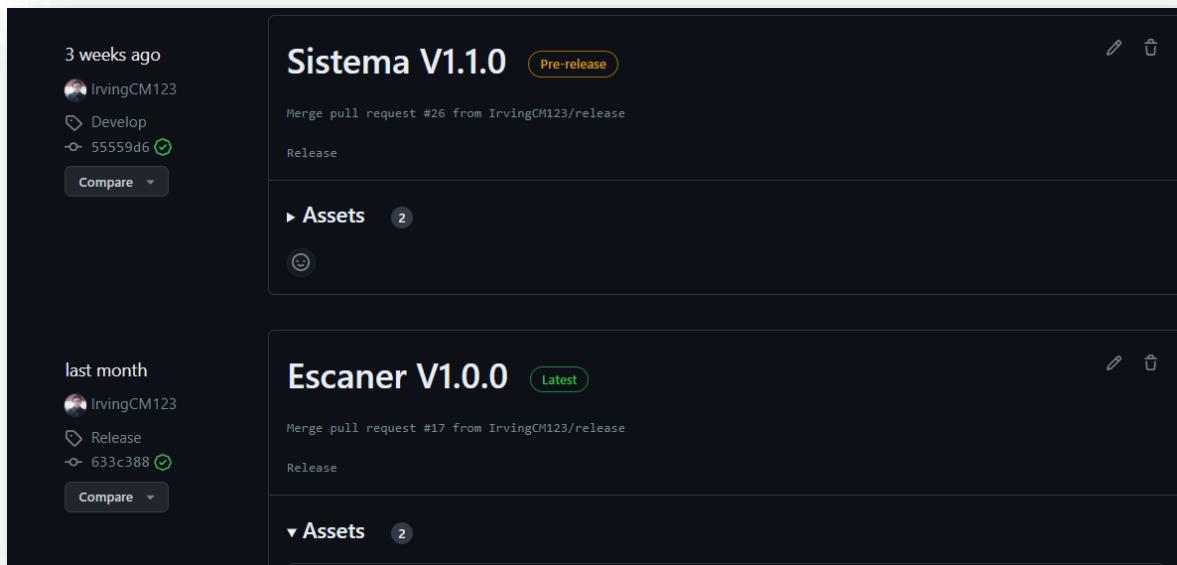
El código ya está terminado, lo único que falta es conectarlo con la base de datos final



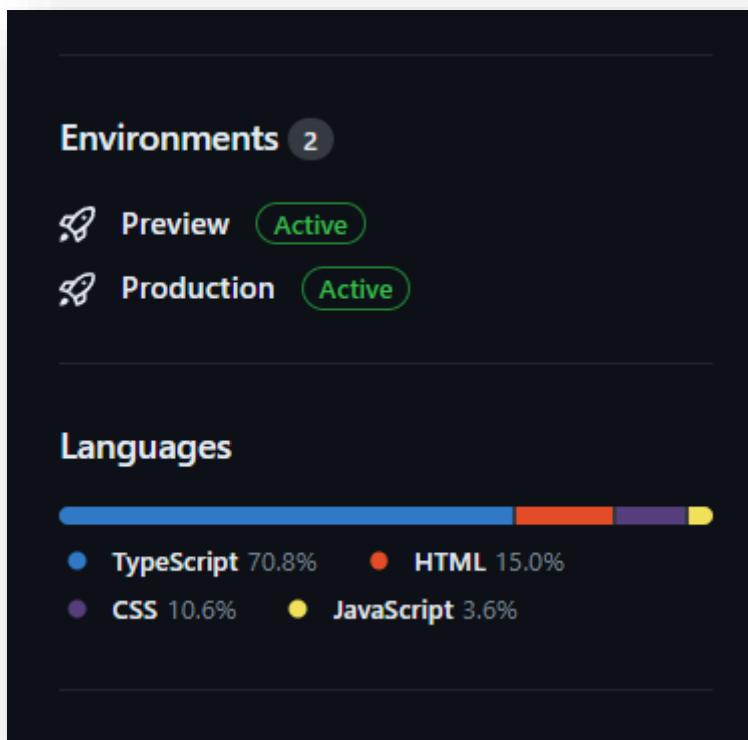
El código igualmente está desplegado en la nube de Vercel y cuenta con un URL



Este repositorio cuenta con dos versiones para controlar sus cambios



De igual forma, este repositorio cuenta con un ambiente de prevializuacion y producción



Gestión del Back-End de Docentes

El repositorio cuenta con un mapa de Branch no tan largo, puesto que el código no es muy grande



Puesto que el código no ha podido ser desplegado en un ambiente de producción, el repositorio no cuenta con un ambiente

About

No description, website, or topics provided.

Activity 0 stars 1 watching 0 forks

Releases

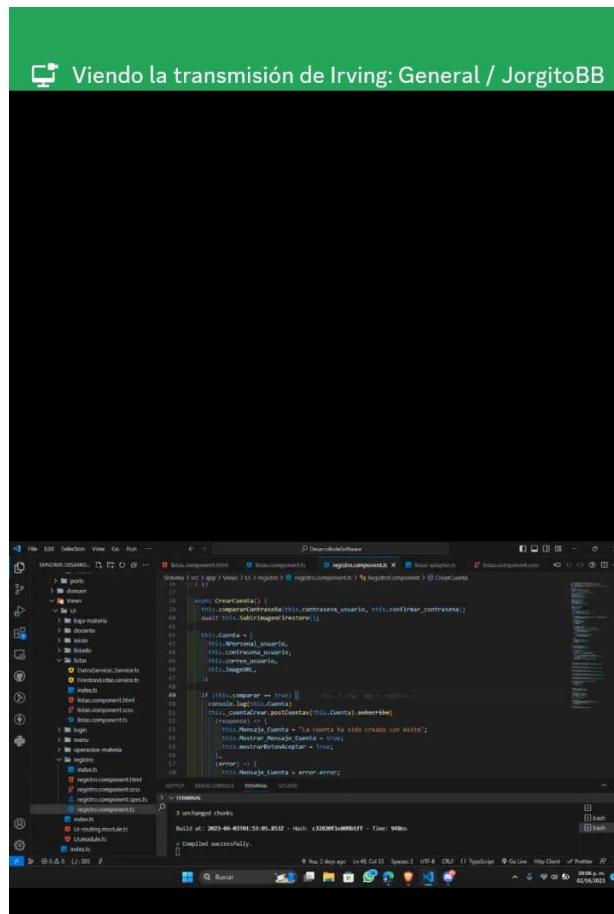
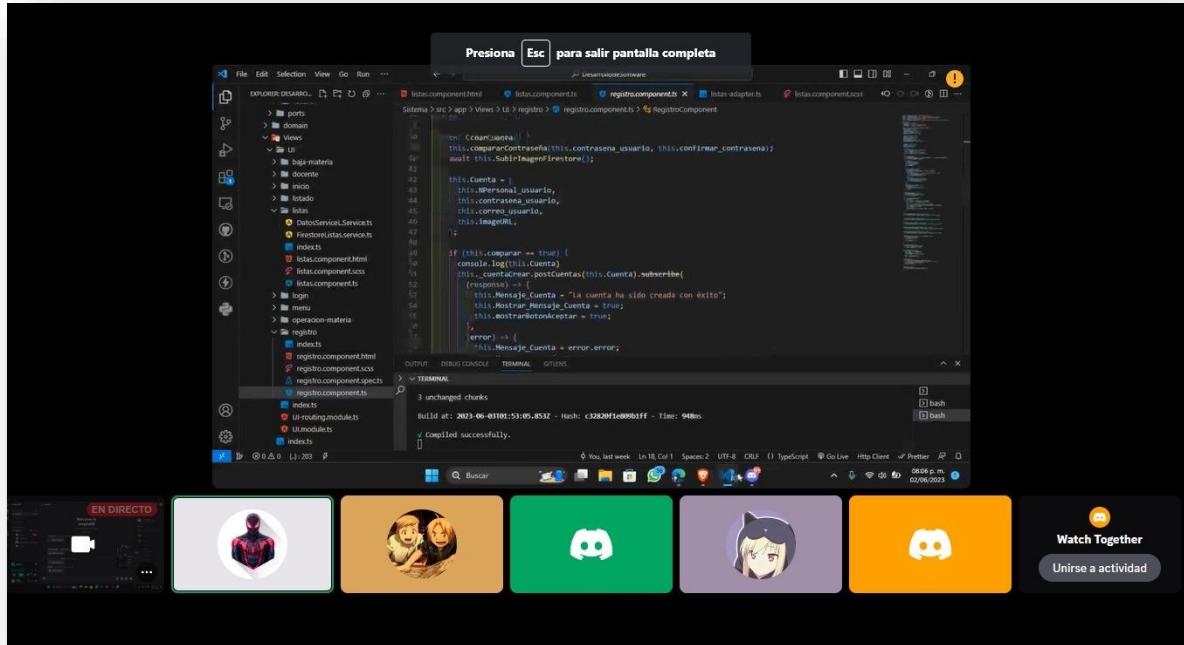
No releases published [Create a new release](#)

Packages

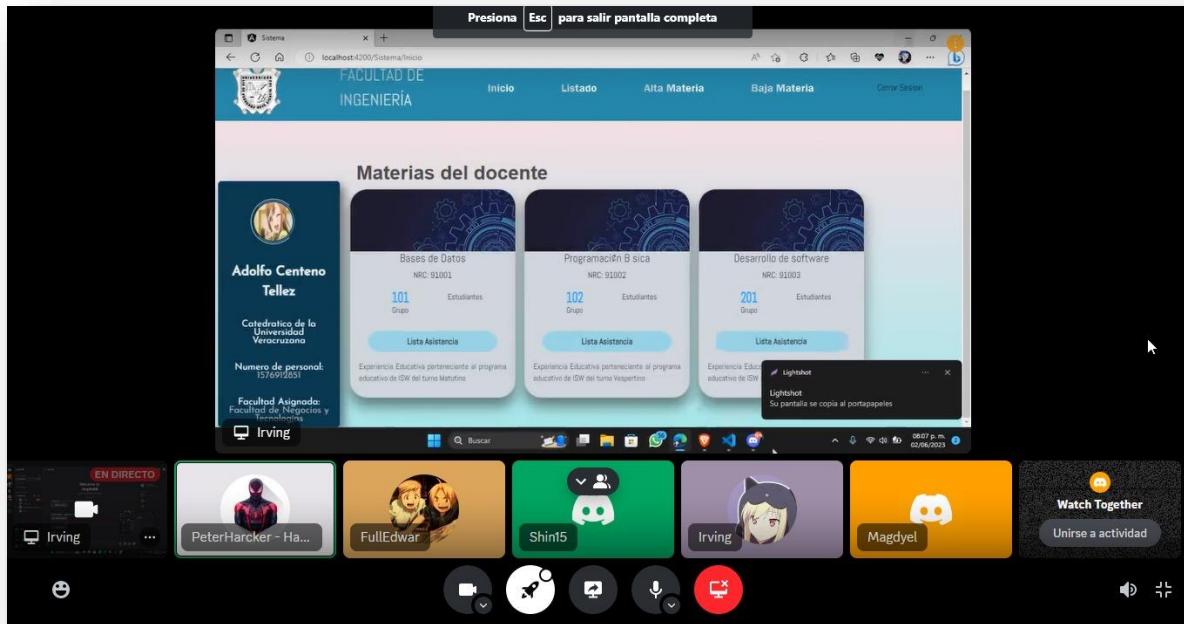
No packages published [Publish your first package](#)

Reuniones de trabajo

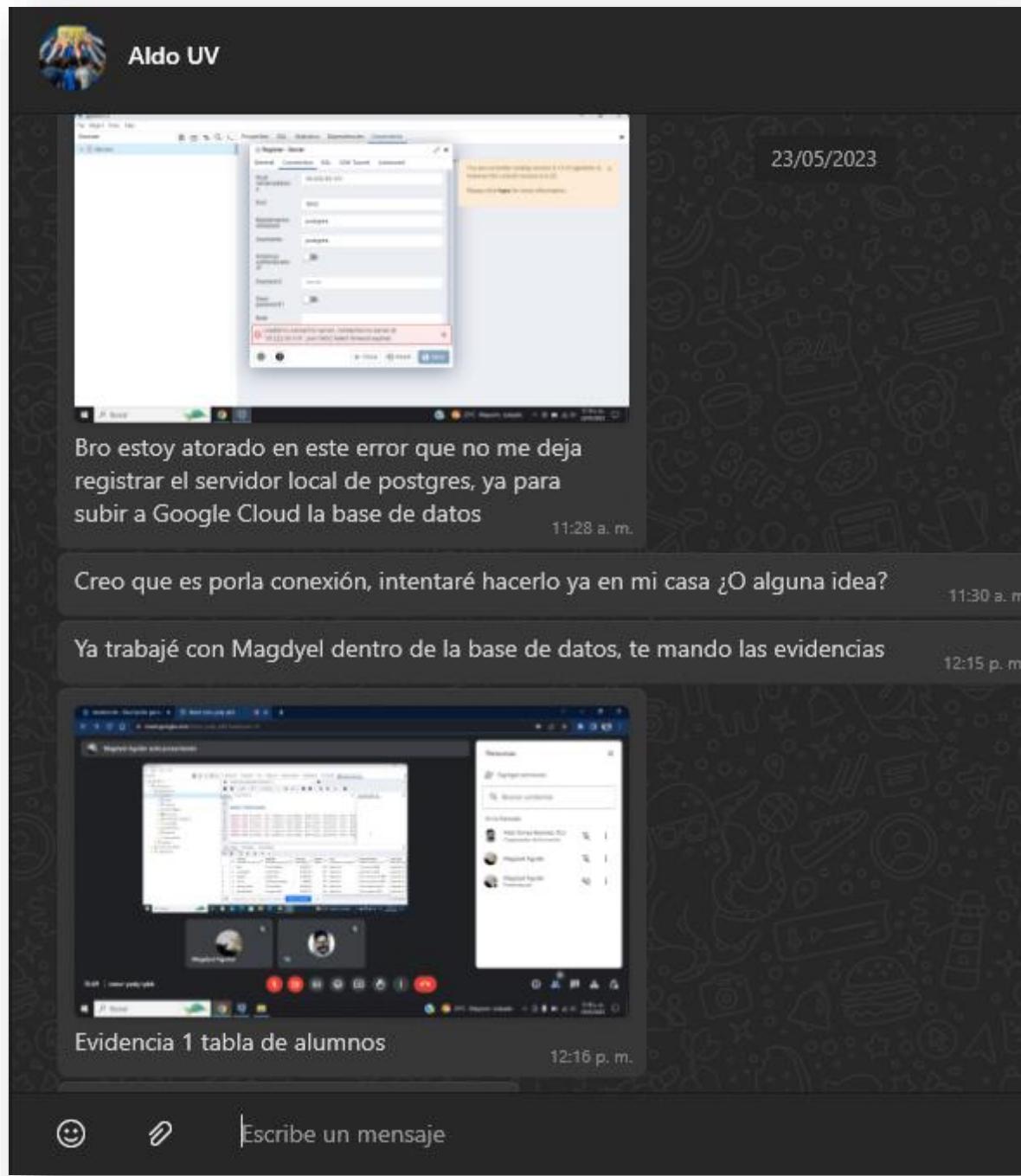
Nuestras reuniones de trabajo solían ser en la escuela o por medio de llamadas de discord



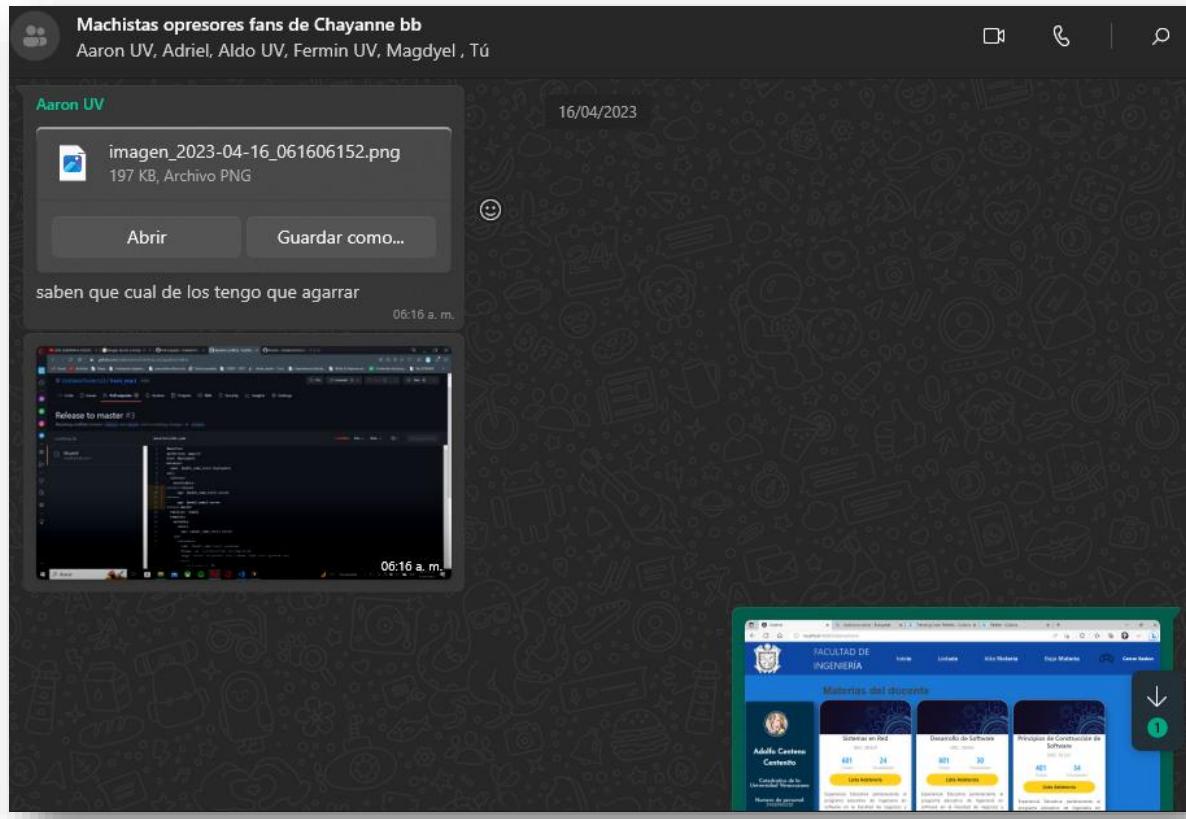
La revisión de los sprints usualmente eran la noche de los viernes, no lo hacíamos diario debido a que nos veíamos en la facultad



Nuestra principal herramienta de comunicación era WhatsApp para informar de cualquier suceso que ocurriese



Se tenía un grupo de WhatsApp para informar las dudas o hablar sobre el proyecto y mantenernos informados de cualquier cosa



Además, se solía trabajar en biblioteca para revisar dudas y evitar el ruido del salón de clases, aunque en ocasiones era necesario trabajar allí para resolver dudas con el profesor



2.10 Conclusiones

En resumen, se identificó una problemática relacionada con el proceso manual de toma de asistencia en la facultad de negocios y tecnologías, que afectaba tanto a los docentes como a los alumnos. Para solucionar este problema, se propuso el desarrollo de un sistema automatizado para el control de asistencia en escuelas, denominado SACAEMEA (Sistema Automatizado para el Control de Asistencia en Escuelas Mediante el uso de Escáneres Auxiliares).

El proyecto SACAEMEA tiene como objetivo principal agilizar el proceso de toma de asistencia utilizando un software especializado y dispositivos periféricos, que en este caso serán simulados mediante una interfaz web. El sistema permitirá a los docentes llevar un control más preciso de las asistencias de los alumnos, eliminando errores y demoras asociadas al registro manual. Además, los alumnos podrán consultar su información registrada y generar códigos QR para la toma de asistencia.

El alcance de la solución incluye el desarrollo de software especializado con interfaces de usuario web para docentes y alumnos, APIs para el funcionamiento de las interfaces, simulador de lectura de códigos QR, y el uso de nubes y bases de datos para el almacenamiento de la información.

En conclusión, el proyecto SACAEMEA ofrece un conjunto de funcionalidades que buscan mejorar y agilizar el proceso de toma de asistencia en entornos educativos. La implementación de estas características permitirá a los docentes gestionar de manera eficiente y precisa la asistencia de los alumnos, brindando una solución automatizada y confiable.