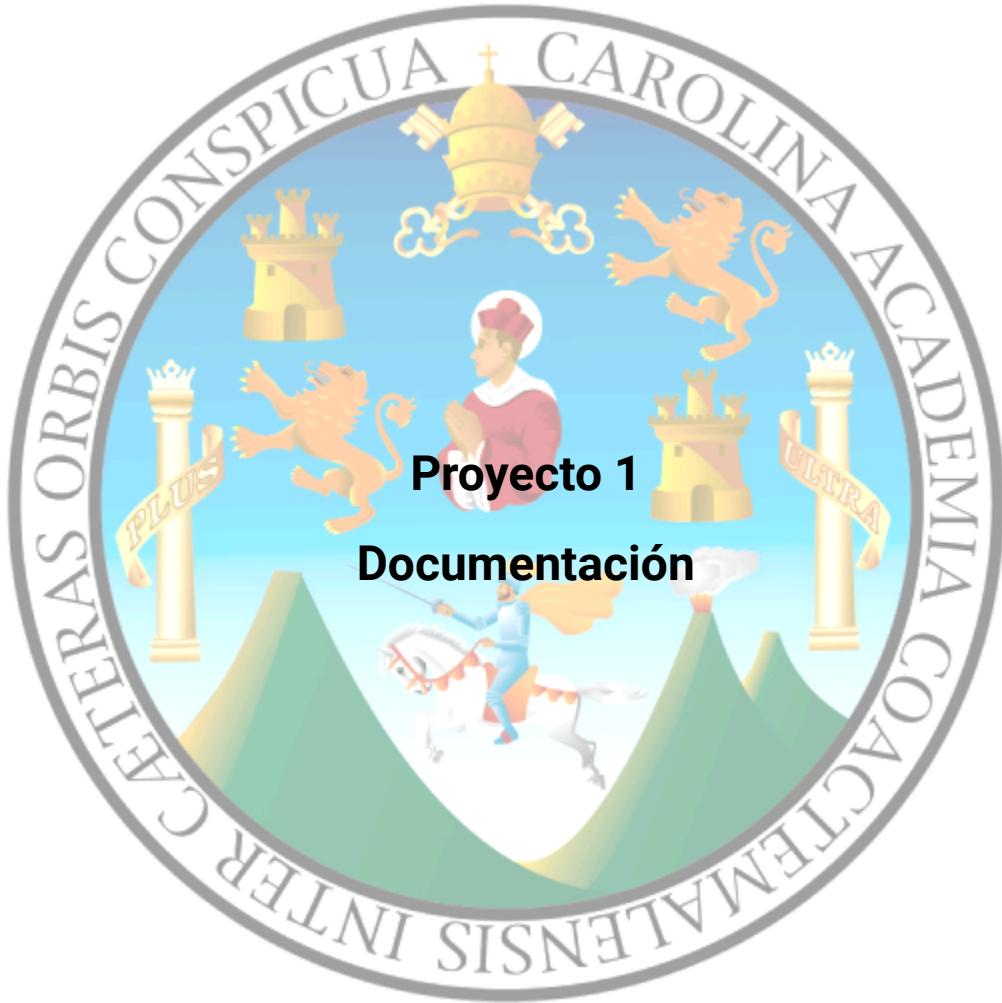


**UNIVERSIDAD DE SAN CARLOS DE GUATEMALA  
LABORATORIO ANÁLISIS Y DISEÑO DE SISTEMAS 1**



**Proyecto 1**

**Documentación**

**NOMBRES**

Eddy Fernando Díaz Galindo

Neidy Aracely Flores Molina

Angel Miguel García Urizar

Saul Jafet Menchu Recinos

Raúl Josue Castillo Barco

**CARNET:**

201906558

201801671

201901421

201906444

202001932

## ÍNDICE

<b>METODOLOGÍA UTILIZADA.....</b>	<b>3</b>
<b>MODELO DE BRANCHING.....</b>	<b>4</b>
<b>REQUERIMIENTOS.....</b>	<b>5</b>
<b>HISTORIAS DE USUARIO.....</b>	<b>6</b>
<b>ARQUITECTURA DE SISTEMA.....</b>	<b>17</b>
<b>DIAGRAMA DE DATOS O ALMACENAMIENTO.....</b>	<b>18</b>
<b>DIAGRAMA DE ACTIVIDADES.....</b>	<b>20</b>
<b>SEGURIDAD DE LA APLICACIÓN.....</b>	<b>21</b>
Home.....	23
Login.....	24
Dashboard Administrador.....	25
1. Home.....	25
2. Registrar Estudiante.....	26
3. Registrar Docente.....	27
4. Registrar Curso.....	28
5. Registrar Carrera.....	29
6. Habilitar Curso.....	30
7. Agregar Horario.....	31
Dashboard Estudiante.....	32
1. Home.....	32
2. Asignar Curso.....	33
3. Desasignar Curso.....	35
4. Consultar Pensum.....	36
5. Perfil.....	37
6. Solicitudes - Cambio de Carrera - Carrera Simultánea - Equivalencias.....	38
Dashboard Docente.....	40
1. Home.....	40
2. Ingresar Notas.....	41
3. Generar Acta.....	43
4. Perfil.....	44
5. Consultar Estudiantes Asignados.....	45
<b>REUNIONES DE DAILY SCRUM.....</b>	<b>46</b>
<b>EVENTOS SCRUM.....</b>	<b>61</b>

## METODOLOGÍA UTILIZADA

Scrum es un marco de trabajo ágil para la gestión y desarrollo de proyectos, especialmente aquellos donde los requisitos son difíciles de especificar o pueden cambiar con el tiempo. Se basa en principios iterativos e incrementales, lo que significa que el trabajo se divide en iteraciones cortas y se entrega de manera incremental en lugar de intentar completar todo el proyecto de una vez.

Roles:

- Scrum Master: Es el facilitador del equipo, se encarga de asegurar que el equipo comprenda y siga los principios y prácticas de Scrum, además de eliminar los obstáculos que puedan surgir.
- Product Owner: Representa a los stakeholders y es responsable de definir y priorizar los requisitos del producto.
- Equipo de Desarrollo: Profesionales encargados de convertir los requisitos en incrementos del producto.

Artefactos:

- Product Backlog: Lista de todos los requisitos del producto, priorizados por el Product Owner.
- Sprint Backlog: Lista de tareas seleccionadas del Product Backlog que el equipo se compromete a completar durante un sprint.

Eventos:

- Sprint Planning: Reunión al inicio de cada sprint donde el equipo selecciona las tareas a realizar y establece el objetivo del sprint.
- Daily Scrum: Reunión diaria de 15 minutos donde el equipo sincroniza su trabajo y planifica las próximas actividades.
- Sprint Review: Reunión al final de cada sprint donde el equipo presenta el incremento completado y recibe retroalimentación.
- Sprint Retrospective: Reunión al final de cada sprint donde el equipo analiza su desempeño y busca mejoras.

El uso de Scrum permite adaptarse a los cambios en los requisitos del proyecto, comunes en proyectos de desarrollo de software. Además Scrum fomenta la colaboración entre los miembros del equipo y la transparencia en todas las actividades del proyecto.

## MODELO DE BRANCHING

Para un proyecto utilizando tecnologías como Laravel, Golang y PostgreSQL desplegado en AWS, es fundamental establecer un modelo de branching que facilite el desarrollo colaborativo, la integración continua y el despliegue continuo. El modelo que se utilizó se basa en la idea de mantener dos ramas principales: master y develop, y utiliza ramas adicionales para desarrollar nuevas características, solucionar problemas y preparar lanzamientos.

- Master Branch (master): Representa la última versión estable del software que está en producción. Las confirmaciones en esta rama deberían reflejar versiones de producción listas para ser desplegadas en el entorno de producción.
- Develop Branch (develop): Rama principal de desarrollo donde se integran todas las características completadas. Las confirmaciones en esta rama son estables y pasan todas las pruebas automatizadas.
- Feature Branches (feature/): Ramas individuales para cada integrante del equipo creadas para desarrollar nuevas características o funcionalidades. Estas ramas se ramifican desde develop y se fusionan nuevamente en develop una vez completadas.
- Release Branches (release/): Ramas creadas para preparar nuevas versiones del software para el lanzamiento. Se utilizan para realizar pruebas finales, correcciones de errores y ajustes menores antes del lanzamiento.
- Hotfix Branches (hotfix/): Ramas creadas para corregir errores críticos en producción que requieren una solución inmediata. Se ramifican directamente desde master, se corrige el error y luego se fusionan tanto en master como en develop.

## **REQUERIMIENTOS**

### **Requerimientos funcionales**

- El sistema permite a los estudiantes iniciar sesión con sus credenciales proporcionadas por la administración.
- Los profesores acceden al sistema con sus credenciales específicas.
- El administrador tiene acceso privilegiado para gestionar estudiantes, docentes, cursos, horarios, carreras y desasignaciones.
- Los estudiantes pueden ver y solicitar cambios de carrera, simultaneidad y equivalencias a través del portal.
- Los estudiantes pueden asignar y desasignar cursos
- Los estudiantes pueden consultar su pensum y generar certificados de aprobación de sus cursos
- Los docentes pueden ingresar notas, generar y consultar actas de sus cursos.
- Los docentes pueden consultar a los estudiantes asignados a sus cursos.
- Los docentes pueden consultar aprobaciones y horarios de sus cursos.
- El sistema permite la habilitación de cursos.
- El sistema permite la carga masiva de horarios.

### **Requerimientos no funcionales**

- El sistema es seguro y protege los datos de los usuarios.
- El rendimiento del sistema es óptimo.
- El sistema es fácil de usar en cualquier dispositivo.
- El sistema es accesible desde diferentes navegadores.
- La comunicación entre el frontend y el backend es eficiente y rápida.
- El sistema es escalable para futuras expansiones de la facultad o cambios en los requisitos.
- El sistema está disponible para su acceso y uso durante un porcentaje alto del tiempo.
- El código del sistema sigue estándares de código documentación.

## HISTORIAS DE USUARIO

### HISTORIA #1

#### Interfaz

**Como:** cliente del producto

**Quiero:** Una página web que se pueda visualizar a través del cualquier dispositivo.

La aplicación web debe contar con los siguientes módulos

1. Administrador
2. Estudiantes
3. Docentes

Que sea un sistema cerrado, es decir, que las credenciales de ingreso al sistema para estudiantes y docentes deben ser generadas por el administrador.

Además quiero una página de bienvenida que ofrezca la posibilidad de iniciar sesión.

**Para:** Tener control total de los usuarios y administrar diferentes módulos que conforman el sistema

## **HISTORIA #2**

### **Administrador**

**Como:** administrador del sistema

**Quiero:** Gestionar los estudiantes, docentes, carreras y cursos del portal de ingeniería

**Para:** Tener control total de los usuarios y administrar los diferentes módulos que conforma el sistema

## **HISTORIA #3**

### **Estudiantes**

**Como:** administrador del sistema

**Quiero:** Gestionar la creación de usuarios con rol de estudiante

**Para:** Registrar nuevos estudiantes para que el portal sea un sistema cerrado

## **HISTORIA #4**

### **Docentes**

**Como:** administrador del sistema

**Quiero:** Gestionar la creación de usuarios con rol de docente

**Para:** Registrar nuevos docentes para que el portal sea un sistema cerrado

## **HISTORIA #5**

### **Carreras**

**Como:** administrador del sistema

**Quiero:** Gestionar la creación de carreras disponibles para impartir en la facultad

**Para:** Registrar nuevas carreras para que los estudiantes puedan asignarse a ellas

## **HISTORIA #6**

### **Cursos**

**Como:** administrador del sistema

**Quiero:** Gestionar la creación de cursos para las distintas carreras registradas

**Para:** Definir el pensum de una carrera en específico

## **HISTORIA #7**

### **Habilitación de cursos**

**Como:** administrador del sistema

**Quiero:** Establecer que cursos serán impartidos en el ciclo correspondiente para cada una de las carreras

**Para:** Construir el horario de cursos

## **HISTORIA #8**

### **Horarios**

**Como:** administrador del sistema

**Quiero:** Definir y establecer cada uno de los horarios en que se imparten los cursos habilitados para cada una de las carreras

**Para:** Aperturar el periodo de asignación para que los estudiantes y docentes conozcan los cursos y sus horarios a impartir en el ciclo correspondiente

## **HISTORIA #9**

### **Consultas**

**Como:** administrador del sistema

**Quiero:** Tener la posiblidad de buscar un estudiante o docente en específico por su identificador único

**Para:** Visualizar la información del usuario

## **HISTORIA #10**

### **Asignacion de cursos**

**Como:** estudiante

**Quiero:** Tener la opción de asignarme a un curso que fue previamente habilitado

**Para:** Cursar cada una de las asignaturas en los diferentes periodos del año, tales como primer semestre, segundo semestre, y vacaciones de ambos semestres.

## **HISTORIA #11**

### **Desasignacion de cursos**

**Como:** estudiante

**Quiero:** Tener la opción de desasignarme a un curso que fue previamente fue asignado

**Para:** Quitar y dejar de llevar un curso de un ciclo por razones específicas

## **HISTORIA #12**

**Ver pensums**

**Como:** estudiante

**Quiero:** Que exista la opcion de ver el listado cursos

**Para:** Cada uno de los pensums de cada una de las carreras impartidas en la facultad

## **HISTORIA #13**

**Ver mi perfil**

**Como:** estudiante

**Quiero:** Listar la información de mi perfil creado como estudiante de esta facultad

**Para:** Verificar mi información correcta y datos personales

## **HISTORIA #14**

**Cambio de carrera**

**Como:** estudiante

**Quiero:** Tener la opcion de solicitar el cambio de carrera por algun motivo

**Para:** realizar las equivalencias para los cursos respectivos a la nueva carrera

## **HISTORIA #15**

**Vista Estudiante**

**Como:** estudiante

**Quiero:** Tener la vista de estudiante

**Para:** Realizar cada una las funcionalidades o acciones que estan disponibles para un estudiante

## **HISTORIA #16**

### **Página Inicial Docente**

**Como:** docente de la facultad

**Quiero:** Quiero poder visualizar todas las opciones que ofrece la página web para un docente

**Para:** Poder acceder a las distintas funcionalidades disponibles desde un solo lugar

## **HISTORIA #17**

### **Notas**

**Como:** docente de la facultad

**Quiero:** Quiero poder visualizar todos los cursos de los cuales soy docente y los estudiantes que están asignados en cada uno

**Para:** Poder ingresar la nota obtenida de los estudiantes de un curso específico

## **HISTORIA #18**

### **Actas**

**Como:** docente

**Quiero:** Tener la opción de generar las actas de los cursos para los que soy docente

**Para:** Una vez ingresadas todas las notas de los estudiantes poder generar el acta correspondiente de manera digital y facilitar el proceso de actas

## **HISTORIA #19**

### **Perfil Docente**

**Como:** docente

**Quiero:** Poder ver mi información personal

**Para:** Asegurarme que la información registra en la facultad siempre sea la correcta y se encuentre actualizada

## **HISTORIA #20**

### **Estudiantes del curso**

**Como:** docente

**Quiero:** Saber que estudiantes están asignados a los diferentes cursos en los que yo soy docente

**Para:** Facilitar el llevar un control sobre los estudiantes que pertenecen al curso, para tener una gestión de zonas y notas más adecuada

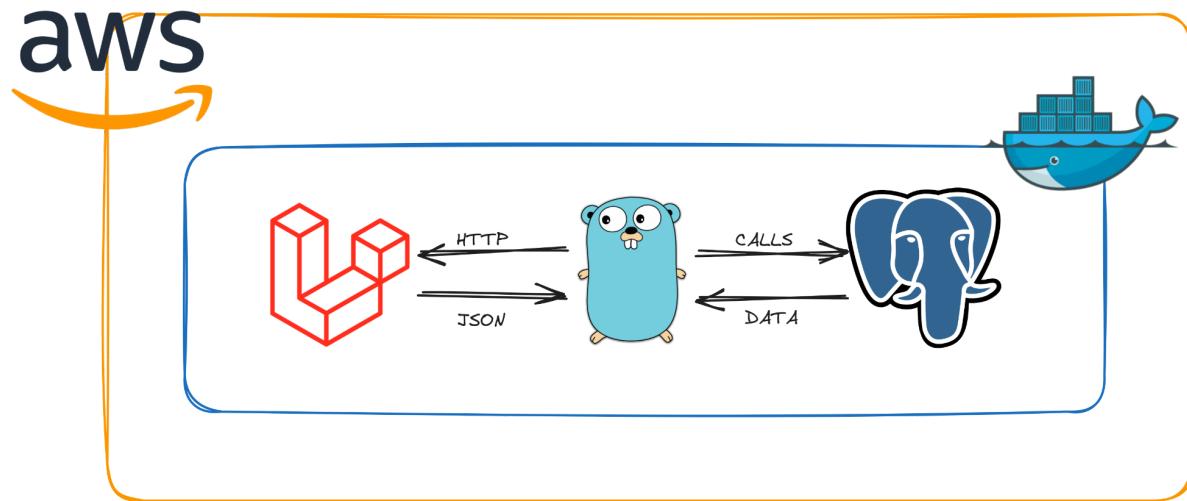
## TECNOLOGÍAS UTILIZADAS

Laravel: Es un popular framework de desarrollo de aplicaciones web PHP, conocido por su elegancia, simplicidad y potencia. Proporciona herramientas y características que permiten a los desarrolladores construir aplicaciones web de manera eficiente y segura.

- Golang (Go): Es un lenguaje de programación de código abierto desarrollado por Google, diseñado para ser eficiente, conciso y productivo. Es conocido por su compilación rápida, su gestión eficiente de la concurrencia y su facilidad de uso. Su tipado estático y sintaxis sencilla que fomenta la legibilidad y la mantenibilidad del código. Además que tiene eficiencia en el consumo de recursos y rendimiento, especialmente en aplicaciones concurrentes.
- PostgreSQL: Es un sistema de gestión de bases de datos relacional de código abierto y altamente potente. Es conocido por su robustez, extensibilidad y conformidad con los estándares SQL. Contempla su modelo de transacciones ACID (Atomicidad, Consistencia, Aislamiento, Durabilidad) que garantizan la integridad de los datos. Además de ser optimizador de consultas que permite la ejecución eficiente de consultas complejas.
- AWS (Amazon Web Services): Es una plataforma de servicios en la nube ofrecida por Amazon, que proporciona una amplia gama de servicios de infraestructura y plataforma como servicio (IaaS y PaaS). Es ampliamente utilizado por empresas y desarrolladores para implementar aplicaciones escalables, seguras y de alta disponibilidad. Para el proyecto se utilizó EC2 (Elastic Compute Cloud) para el almacenamiento y gestión de servidores virtuales.

## ARQUITECTURA DE SISTEMA

Cuando un usuario realiza una solicitud a la aplicación web, esta solicitud se dirige a la máquina virtual que está desplegada en AWS. La máquina virtual tiene ejecutando todo el sistema con ayuda de docker el cual tiene tres imágenes en una misma red para lograr la conexión eficiente entre laravel, golang y postgres.



## DIAGRAMA DE DATOS O ALMACENAMIENTO

El diagrama lógico se enfocó en la estructura conceptual de la base de datos, mientras que el diagrama físico se concentró en la implementación concreta y física de esa estructura.

Diagrama Lógico: Ayuda a los diseñadores y a los usuarios a comprender la organización y la relación de los datos dentro de la base de datos.

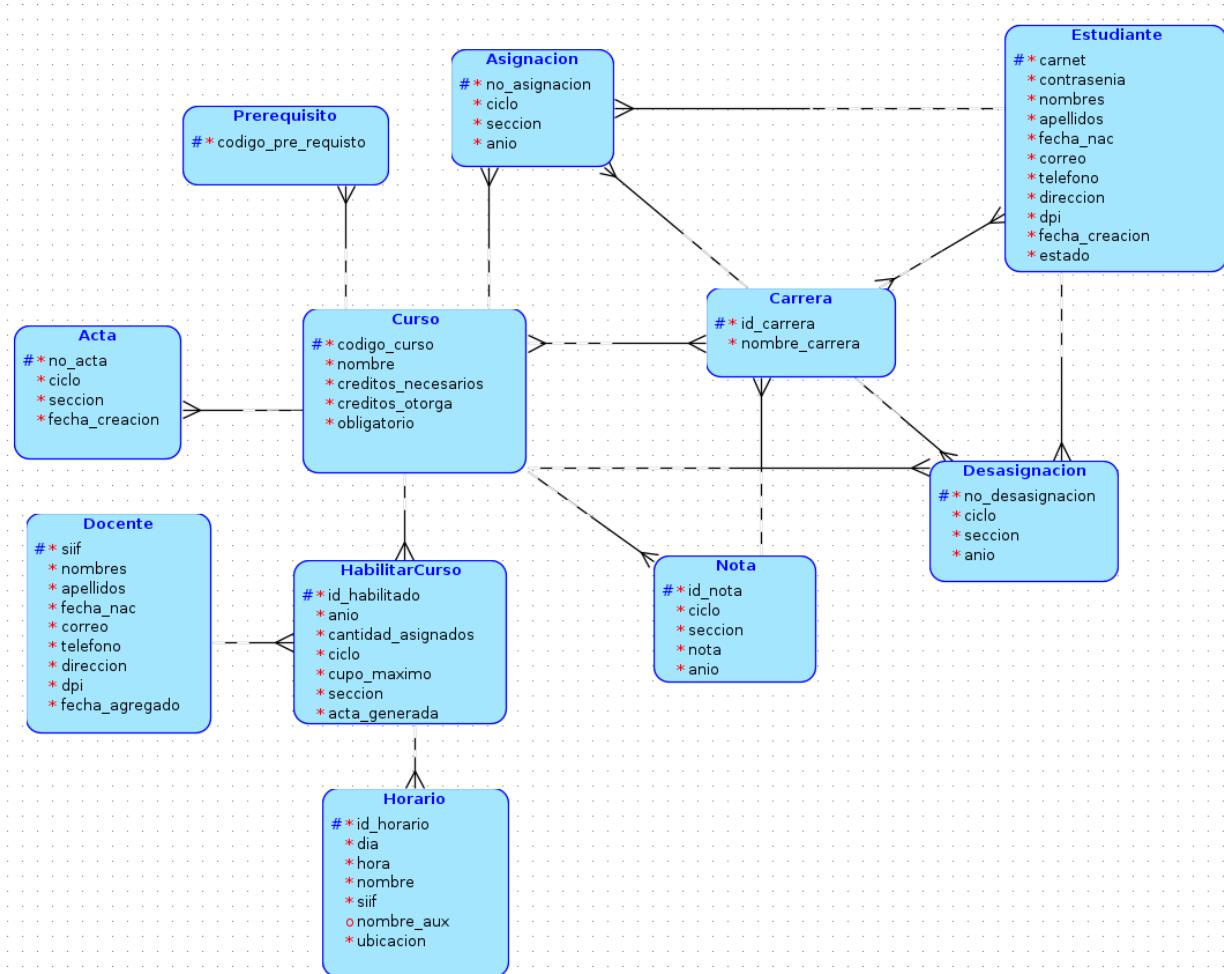
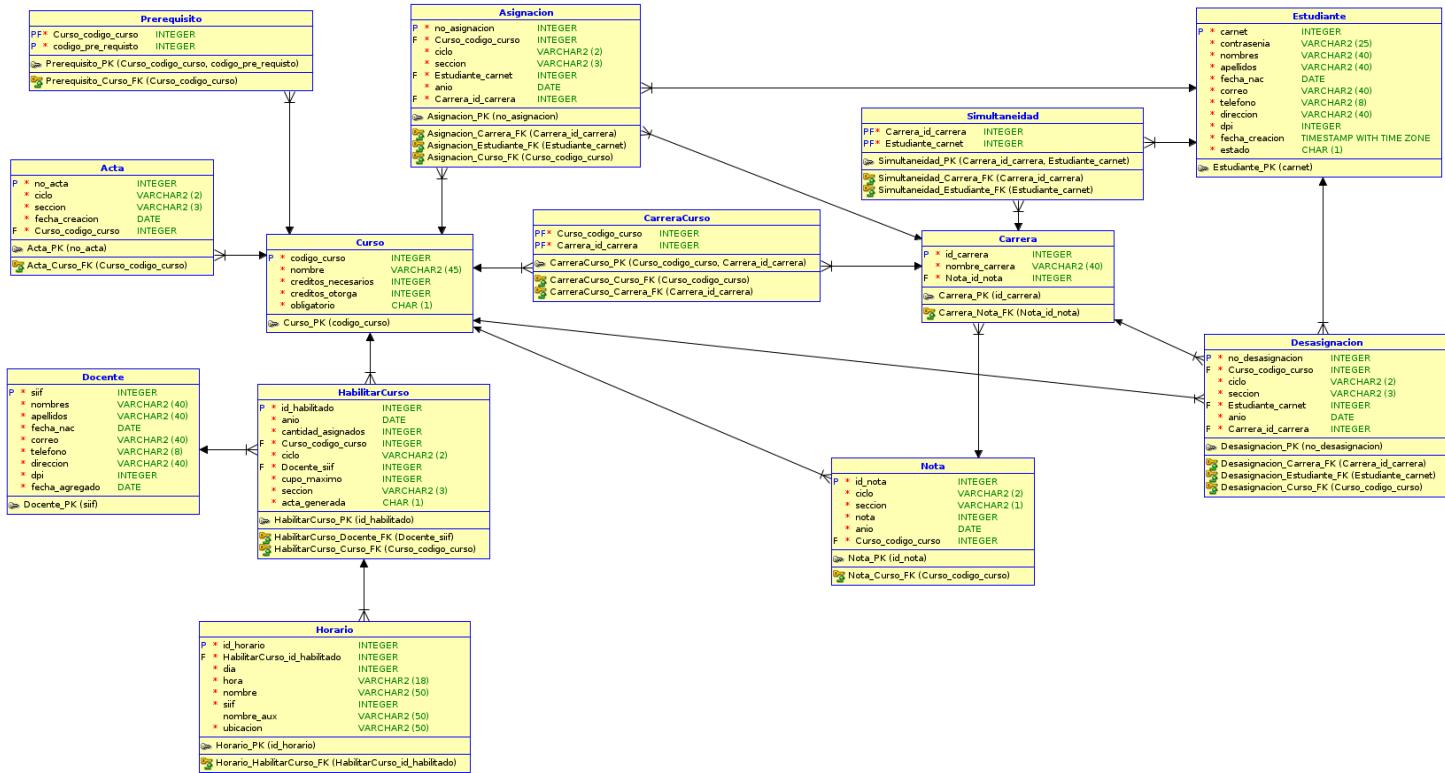


Diagrama Relacional: Se utilizó para facilitar la comunicación entre los desarrolladores y los administradores de bases de datos al presentar una representación visual de la estructura de datos.



## DIAGRAMA DE ACTIVIDADES

Día	Área de trabajo				Actividad
	Frontend	Backend	BBDD	Cloud	
20/03/2024			X		Creación del modelo lógico y físico
20/03/2024			X		Creación de tablas del modelo
20/03/2024			X		Conexión a PostgreSQL
21/03/2024	X				Conexión a Golang
21/03/2024	X				Vista home principal
21/03/2024	X				Vista home del docente
21/03/2024	X				Login
21/03/2024	X				Vista home del administrador
21/03/2024	X				Registro de docentes
21/03/2024	X				Registro de estudiantes
21/03/2024	X				Vista home de estudiantes
22/03/2024		X			Registro de estudiantes
22/03/2024		X			Registro de docentes
22/03/2024					Registro de estudiantes
22/03/2024			X		Registro de docentes
22/03/2024			X		Creación de curso
22/03/2024			X		Creación de horario
22/03/2024			X		Creación de prerequisito
23/03/2024		X			Creación de cursos
23/03/2024		X			Habilitación de horario
23/03/2024		X			Creación de prerequisitos

23/03/2024		X			Carga de horarios
23/03/2024	X				Creación de usuarios
23/03/2024	X				Carga de horarios
23/03/2024	X				Habilitación de curso para asignación
23/03/2024	X				Vista para cargar CSV
24/03/2024			X		Creación de carrera
24/03/2024			X		Creación de simultaneidad
24/03/2024		X			Creación de carrera
24/03/2024		X			Creación de carrera para simultaneidad
25/03/2024	X				Creación de carrera
25/03/2024	X				Creación de solicitud de simultaneidad
25/03/2024			X		Asignación de curso
25/03/2024			X		Desasignación de curso
25/03/2024		X			Asignación de curso
25/03/2024		X			Desasignación de curso
26/03/2024	X				Asignación de curso
26/03/2024	X				Desasignación de curso
26/03/2024		X			Carga de notas
26/03/2024		X			Generación de actas
26/03/2024			X		Creación de actas
26/03/2024			X		Creación de notas
27/03/2024	X				Carga de notas
27/03/2024	X				Generación de actas
27/03/2024			X		Consultar pensum

28/03/2024		X		Consultar cursos aprobados
28/03/2024	X			Consultar pensum
28/03/2024	X			Consultar cursos aprobados
28/03/2024		X		Consultar info estudiante
28/03/2024		X		Consultar info docente
28/03/2024	X			Consultar info estudiante
28/03/2024	X			Consultar info docente
28/03/2024		X		Consulta de asignados
28/03/2024		X		Consulta de actas
28/03/2024		X		Consulta de aprobaciones
29/03/2024	X			Consulta de asignados
29/03/2024	X			Consulta de actas
29/03/2024	X			Consulta de aprobaciones
30/03/2024				Creación de cuenta en AWS
30/03/2024				Creación de dockerfiles
30/03/2024				Creación de docker compose

## **SEGURIDAD DE LA APLICACIÓN**

Para la seguridad de ciertas funcionalidades se implementó la autenticación por cookies. Al iniciar sesión, luego que un usuario envía sus credenciales y estas se validan, el servidor registra datos con el fin de recordar el usuario que fue autenticado. Estos datos que se registran en el backend, en correspondencia con el identificador de sesión, es lo que se conoce como estado.

En el lado del cliente una cookie es creada para almacenar el identificador de sesión, mientras que los datos se almacenan en el servidor y son llamadas variables de sesión.

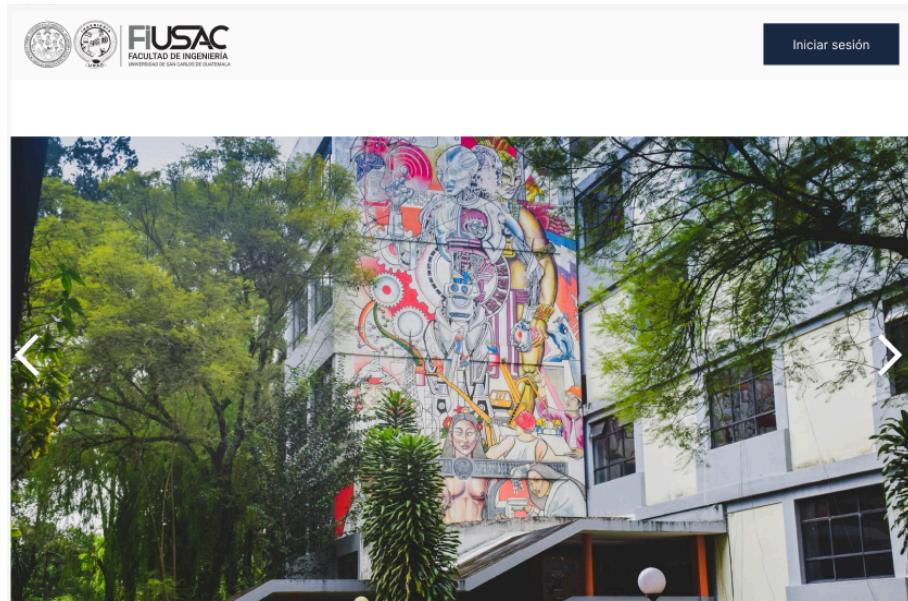
El flujo que sigue este sistema de autenticación tradicional es el siguiente:

- Un usuario ingresa sus credenciales.
- El servidor verifica que las credenciales sean correctas, y crea una sesión (esto puede corresponderse con la creación de un archivo, un registro nuevo en una base de datos, o alguna otra solución server-side)
- Una cookie con el session ID es puesta en el navegador web del usuario
- En las peticiones siguientes, el session ID es comparado con las sesiones creadas por el servidor
- Una vez que el usuario se desconecta, la sesión es destruida en ambos lados (tanto en el cliente como en el servidor)

## MOCKUPS DE LAS VISTAS DE LA PÁGINA WEB

Para darle solución al proyecto planteado en el enunciado, se definieron tres roles distintos que podrán interactuar con la web del proyecto, cada uno tendrá acceso a diferentes funcionalidades después de ser autenticados en el Login se establece la autorización del usuario y su nivel de permisos, de modo que cada rol se mantenga aislado y seguro.

### Home



Ruta	http://localhost:8080/		
Componente	Objetivo	Característica Funcional	Característica No Funcional
Botón de Inicio	Redirigir a la vista de login	X	
Flecha Siguiente	Cambiar a la imagen del siguiente display	X	
Flecha Anterior	Cambiar a la imagen anterior del display	X	

## Login

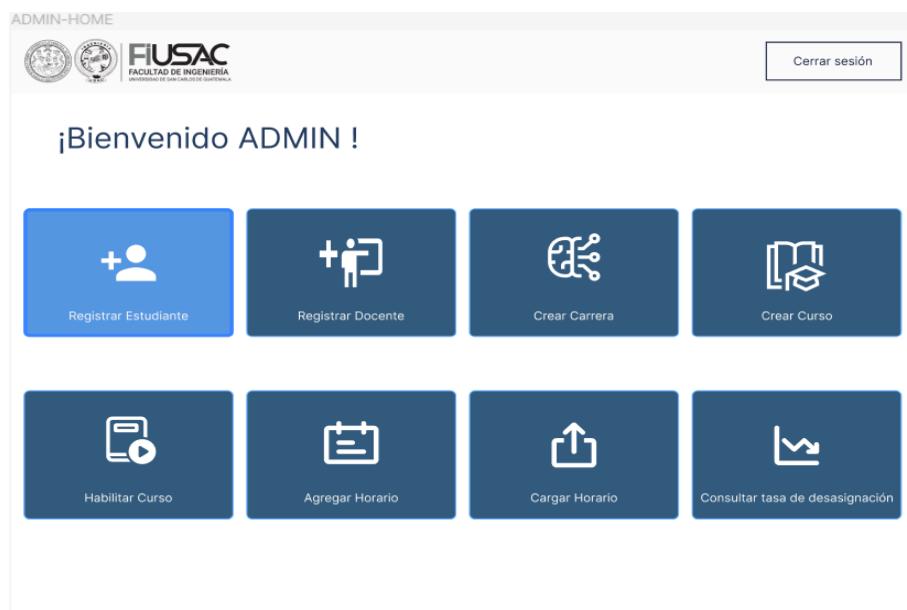
The screenshot shows a login interface for the Faculty of Engineering. At the top, there is a logo of the University of San Carlos of Guatemala and the text "FACULTAD DE INGENIERÍA" and "UNIVERSIDAD DE SAN CARLOS DE GUATEMALA". Below this, the word "LOGIN" is visible. The main area is titled "Ingresa a tu perfil". It contains two input fields: one for "Usuario" and one for "Contraseña", both represented by blue rectangular boxes. Below these fields is a blue rectangular button labeled "Iniciar sesión".

Ruta	http://localhost:8080/login		
Componente	Objetivo	Característica Funcional	Característica No Funcional
Botón de Inicio de Sesión	Obtener los datos del usuario para enviarlos por medio del formulario con una petición POST	X	
Autenticación de Usuarios	Comunicarse con el backend para verificar si el usuario tiene acceso y su nivel de		X

<b>Ruta</b>	<a href="http://localhost:8080/login">http://localhost:8080/login</a>		
<b>Componente</b>	<b>Objetivo</b>	<b>Característica Funcional</b>	<b>Característica No Funcional</b>
	autorización con una petición GET		

## Dashboard Administrador

### 1. Home



<b>Ruta</b>	<a href="http://localhost:8080/adminHome">http://localhost:8080/adminHome</a>		
<b>Componente</b>	<b>Objetivo</b>	<b>Característica Funcional</b>	<b>Característica No Funcional</b>
Botón Cerrar Sesión	Cerrar sesión y redirigir a la pantalla de login	X	
Botón Registrar Estudiante	Redirigir al formulario para registrar un nuevo estudiante	X	
Botón Registrar Docente	Redirigir al formulario para registrar un nuevo docente	X	
Botón Crear Carrera	Redirigir al formulario para registrar una nueva carrera	X	

Ruta	http://localhost:8080/adminHome		
Componente	Objetivo	Característica Funcional	Característica No Funcional
Botón Crear Curso	Redirigir al formulario para crear un nuevo curso	X	
Botón Habilitar curso	Redirigir a formulario donde se podrá solicitar la habilitación de un curso previamente creado	X	
Botón Agregar Horario	Redirigir a un formulario para ingresar un horario para un curso habilitado	X	
Botón Cargar Horario	Desplegar el selector de archivos para cargar un archivo de tipo csv	X	
Botón Consultar Tasa de Desasignación	Redirigir a la pantalla en donde se desplegarán datos de alumnos desasignados	X	

## 2. Registrar Estudiante

ADMIN-REGISTER-STUDENT

DATOS DEL NUEVO ESTUDIANTE

Carnet	Carrera
<input type="text"/>	09 - Ingeniería en Ciencias y Sistemas
Número de DPI	Correo
<input type="text"/>	<input type="text"/>
Nombre Completo	Teléfono
<input type="text"/>	<input type="text"/>
Fecha de Nacimiento	Dirección
<input type="text"/>	<input type="text"/>
<b>Registrar Estudiante</b>	

FIUSAC

Ruta	http://localhost:8080/adminStudent		
Componente	Objetivo	Característica Funcional	Característica No Funcional
Botón de Registrar Docente	Obtener los datos del formulario y solicitar la creación de un nuevo estudiante al backend con una petición POST	X	
Botón Cerrar Sesión	Cerrar sesión y redirigir a la pantalla de login	X	

### 3. Registrar Docente

The screenshot shows a web page titled "DATOS DEL NUEVO DOCENTE" (New Teacher Data). The page has a blue header with the text "ADMIN-REGISTER-TEACHER" and the logo of FIUSAC (Facultad de Ingeniería de la Universidad de San Carlos de Guatemala). On the right side of the header is a "Cerrar sesión" (Logout) button. The main form area has several input fields: "Registro Personal" and "Correo" in the top row; "Número de DPI" and "Teléfono" in the middle row; "Nombre Completo" and "Dirección" in the bottom row. Below these fields is a "Fecha de Nacimiento" field. At the bottom of the form is a blue button labeled "Registrar Docente". The background of the form features a faint image of a person's face.

Ruta	http://localhost:8080/adminDocente		
Componente	Objetivo	Característica Funcional	Característica No Funcional
Botón de Registrar Docente	Obtener los datos del formulario y solicitar la creación de un nuevo docente al backend con una petición POST	X	
Botón Cerrar Sesión	Cerrar sesión y redirigir a la pantalla de login	X	

#### 4. Registrar Curso

ADMIN-REGISTER-COURSE

DATOS DEL NUEVO CURSO

Código	Créditos Necesarios
Nombre	Créditos que Otorga
Carrera a la que pertenece 09 - Ingeniería en Ciencias y Sistemas	Es obligatorio

Registrar Curso

Ruta	http://localhost:8080/adminCurso		
Componente	Objetivo	Característica Funcional	Característica No Funcional
Botón de Registrar Curso	Obtener los datos del formulario y solicitar la creación de un nuevo curso al backend con una petición de POST	X	
Petición de carreras	Comunicarse con el backend con una petición GET para obtener el listado de carreras disponibles y desplegarlas en el componente select		X
Botón Cerrar Sesión	Cerrar sesión y redirigir a la pantalla de login	X	

## 5. Registrar Carrera

The screenshot shows a web-based administrative interface for registering new programs. At the top left is the header "ADMIN-REGISTER-CAREER". To the right of the header are the logos of the University of San Carlos de Guatemala (USAC) and the Faculty of Engineering (FIUSAC), along with the text "FIUSAC" and "FACULTAD DE INGENIERÍA". In the top right corner is a "Cerrar sesión" (Logout) button. The main content area is titled "DATOS DE LA NUEVA CARRERA" (Data of the New Program). It contains three input fields: "Código" (Code), "Nombre" (Name), and "Créditos Necesarios" (Required Credits). Below these fields is a blue rectangular button labeled "Registrar Carrera" (Register Program). The background of the form features a large, faint watermark of the USAC logo.

Ruta	http://localhost:8080/adminCarrera		
Componente	Objetivo	Característica Funcional	Característica No Funcional
Botón de Registrar Carrera	Obtener los datos del formulario y solicitar la creación de una nueva carrera al backend con una petición de POST	X	
Botón Cerrar Sesión	Cerrar sesión y redirigir a la pantalla de login	X	

## 6. Habilitar Curso

ADMIN-ENABLE-COURSE

**DATOS DEL CURSO**

Código del Curso  
283 - Análisis y Diseño de Sistemas 1

Ciclo

Docente

Cupo Máximo

Sección

Habilitar Curso

Ruta	http://localhost:8080/adminHabilitar		
Componente	Objetivo	Característica Funcional	Característica No Funcional
Botón de Habilitar Curso	Obtener los datos del formulario y solicitar la habilitación de un curso (previamente creado) al backend con una petición de POST	X	
Petición de cursos	Comunicarse con el backend con una petición GET para obtener el listado de cursos disponibles y desplegarlos en el componente select		X
Botón Cerrar Sesión	Cerrar sesión y redirigir a la pantalla de login	X	

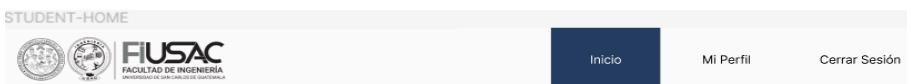
## 7. Agregar Horario

The screenshot shows a web-based administrative interface for scheduling. At the top, it displays the logo of the Faculty of Engineering at the University of San Carlos of Guatemala (FIUSAC). The main title of the page is "DATOS DEL HORARIO". Below the title, there is a dropdown menu labeled "Código del Curso Habilitado" which is currently set to "283 - Análisis y Diseño de Sistemas 1". There are two input fields: "Día" and "Horario", both of which are currently empty. At the bottom of the form is a large blue button labeled "Agregar Horario". In the top right corner of the main window, there is a small rectangular button labeled "Cerrar sesión" (Close session).

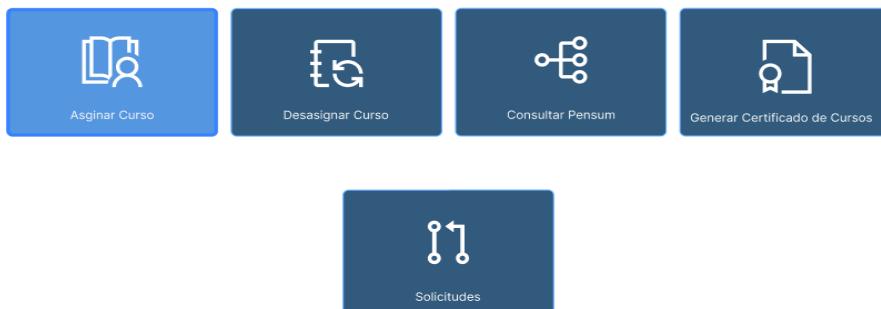
Ruta	http://localhost:8080/adminAgregar		
Componente	Objetivo	Característica Funcional	Característica No Funcional
Botón de Agregar Horario	Obtener los datos del formulario y solicitar la asignación de horarios a un curso (previamente habilitado) al backend con una petición de POST	X	
Petición de cursos	Comunicarse con el backend con una petición GET para obtener el listado de cursos disponibles y desplegarlos en el componente select		X
Botón Cerrar Sesión	Cerrar sesión y redirigir a la pantalla de login	X	

## Dashboard Estudiante

### 1. Home



¡Bienvenido NOMBRE !



Ruta	http://localhost:8080/studentHome		
Componente	Objetivo	Característica Funcional	Característica No Funcional
Botón Cerrar Sesión	Cerrar sesión y redirigir a la pantalla de login	X	
Botón Mi Perfil	Redirigir a la pantalla en donde se muestra la información del estudiante	X	
Botón Inicio	Redirigir al menú home de estudiante	X	
Botón Asignar Curso	Redirigir al formulario para asignarse a un nuevo curso	X	
Botón Desasignar Curso	Redirigir a la vista de cursos asignados para desasignarse un curso	X	
Botón Consultar Pensum	Redirigir a formulario donde se podrá solicitar el despliegue de cursos correspondiente a una carrera	X	

Ruta	http://localhost:8080/studentHome		
Componente	Objetivo	Característica Funcional	Característica No Funcional
Botón Generar Certificado de Cursos	Generar una solicitud de un nuevo certificado de cursos aprobados	X	
Botón Solicitudes	Redirigir a la vista de formularios de solicitudes cambio de carrera, simultánea y equivalencia de cursos	X	

## 2. Asignar Curso

The screenshot displays a web interface titled "DATOS DEL CURSO" (Course Data). At the top, there is a header with the text "STUDENT-ASSING-COURSE", the logo of the Facultad de Ingeniería (FIUSAC), and navigation links for "Inicio", "Mi Perfil", and "Cerrar Sesión". The main content area features a large background image of a graduation cap (mortarboard) with the letters "USAC" on it. The form itself has three input fields: "Código del Curso" containing "283 - Análisis y Diseño de Sistemas 1", "Ciclo" (Cycle), and "Sección" (Section). Below these fields is a blue button labeled "Asignar Curso" (Assign Course).

Ruta	http://localhost:8080/studentAssign		
Componente	Objetivo	Característica Funcional	Característica No Funcional
Botón Cerrar Sesión	Cerrar sesión y redirigir a la pantalla de login	X	
Botón Mi Perfil	Redirigir a la pantalla en donde se muestra la información del estudiante	X	
Botón Inicio	Redirigir al menú home de estudiante	X	
Botón de Asignar Curso	Obtener los datos del formulario y solicitar la asignación de horarios a un curso (previamente habilitado) al backend con una petición de POST	X	
Petición de cursos	Comunicarse con el backend con una petición GET para obtener el listado de cursos disponibles y desplegarlos en el componente select		X

### 3. Desasignar Curso



Ruta	http://localhost:8080/studentDeassign		
Componente	Objetivo	Característica Funcional	Característica No Funcional
Botón Cerrar Sesión	Cerrar sesión y redirigir a la pantalla de login	X	
Botón Mi Perfil	Redirigir a la pantalla en donde se muestra la información del estudiante	X	
Botón Inicio	Redirigir al menú home de estudiante	X	
Botones Desasignar	Obtener los datos de la fila seleccionada y solicitar la desasignación del curso asignado, al backend con una petición de POST	X	
Petición de cursos asignados	Comunicarse con el backend con una petición GET para obtener el listado de cursos en los que el estudiante se encuentra asignado y desplegarlos en la tabla		X

#### 4. Consultar Pensum



Ruta	http://localhost:8080/studentGetCurriculum		
Componente	Objetivo	Característica Funcional	Característica No Funcional
Botón Cerrar Sesión	Cerrar sesión y redirigir a la pantalla de login	X	
Botón Mi Perfil	Redirigir a la pantalla en donde se muestra la información del estudiante	X	
Botón Inicio	Redirigir al menú home de estudiante	X	
Botón Consultar	Obtener los datos del formulario y solicitar el listado de cursos que pertenecen a la carrera seleccionada con una petición GET y desplegando los datos en la tabla	X	
Petición de Carreras	Comunicarse con el backend con una petición GET para obtener el listado de carreras disponibles y desplegarlos en el		X

Ruta	<a href="http://localhost:8080/studentGetCurriculum">http://localhost:8080/studentGetCurriculum</a>		
Componente	Objetivo	Característica Funcional	Característica No Funcional
	componente select		

## 5. Perfil

STUDENT-PROFILE

NOMBRE COMPLETO

Carnet  Fecha de Nacimiento

Número de DPI  Correo

Carrera  Teléfono   
09 - Ingeniería en Ciencias y Sistemas

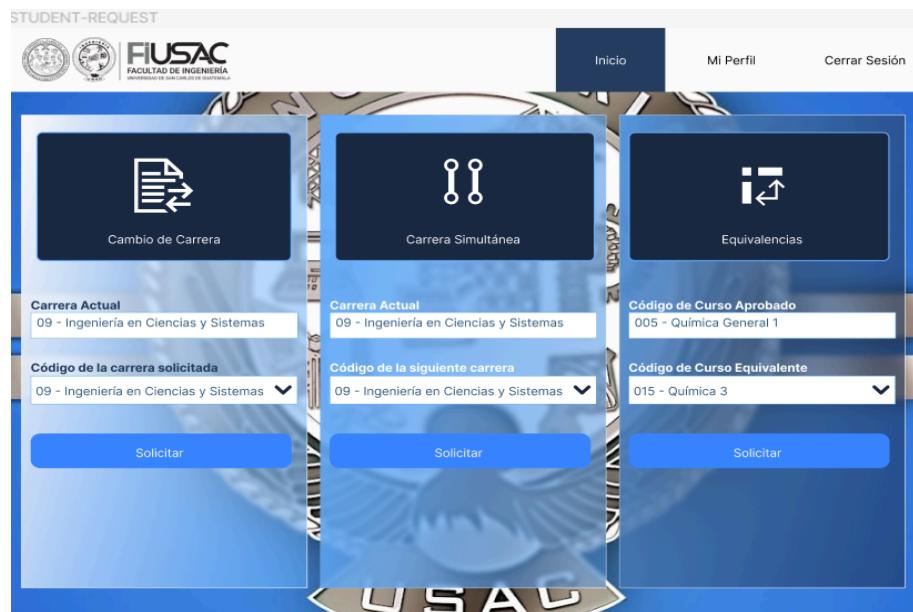
Créditos aprobados  Dirección

Darme de Baja

	<a href="http://localhost:8080/studentProfile">http://localhost:8080/studentProfile</a>		
Componente	Objetivo	Característica Funcional	Característica No Funcional
Botón Cerrar Sesión	Cerrar sesión y redirigir a la pantalla de login	X	
Botón Mi Perfil	Redirigir a la pantalla en donde se muestra la información del estudiante	X	
Botón Inicio	Redirigir al menú home de estudiante	X	
Botón de Darme de Baja	Obtener los datos del formulario y solicitar la baja del sistema	X	

	<a href="http://localhost:8080/studentProfile">http://localhost:8080/studentProfile</a>		
Componente	Objetivo	Característica Funcional	Característica No Funcional
Petición de estudiante	Comunicarse con el backend con una petición GET para obtener toda la información del estudiante logueado y mostrarla en el formulario		X

## 6. Solicitudes - Cambio de Carrera - Carrera Simultánea - Equivalencias



Ruta	<a href="http://localhost:8080/studentRequest">http://localhost:8080/studentRequest</a>		
Componente	Objetivo	Característica Funcional	Característica No Funcional
Botón Cerrar Sesión	Cerrar sesión y redirigir a la pantalla de login	X	
Botón Mi Perfil	Redirigir a la pantalla en donde se muestra la información del estudiante	X	
Botón Inicio	Redirigir al menú home de	X	

Ruta	http://localhost:8080/studentRequest		
Componente	Objetivo	Característica Funcional	Característica No Funcional
	estudiante		
Botón Solicitar Cambio	Obtener los datos del formulario y solicitar el cambio de carrera al backend	X	
Botón Solicitar Simultánea	Obtener los datos del formulario y solicitar la inscripción a una carrera simultánea al backend mediante una petición POST	X	
Botón Solicitar Equivalencia	Obtener los datos del formulario y solicitar la equivalencia	X	
Petición de estudiante	Comunicarse con el backend con una petición GET para obtener la(s) carrera(s) que el estudiante cursa y desplegarlas en el select de carrera actual		X
Petición de Carreras	Comunicarse con el backend con una petición GET para obtener las carreras disponibles y desplegarlas en los select		X

## Dashboard Docente

### 1. Home

The screenshot shows the 'PROFESSOR-HOME' interface. At the top, there is a header with the university's logo and name 'FIUSAC' (Facultad de Ingeniería Universidad de San José de Matagalpa). On the right side of the header are three buttons: 'Inicio' (Home), 'Mi Perfil' (My Profile), and 'Cerrar Sesión' (Logout). Below the header, a welcome message '¡Bienvenido NOMBRE!' is displayed. The main area contains several functional buttons arranged in a grid:

- Top row: 'Ingresar Notas' (Ingress Notes) with a blue background, 'Generar Acta' (Generate Acta) with a dark blue background, 'Consultar Estudiantes Asignados' (Consult Assigned Students) with a dark blue background, and 'Consultar Aprobaciones' (Consult Approvals) with a dark blue background.
- Second row: 'Consultar Actas' (Consult Acts) with a dark blue background, and 'Consultar Horarios' (Consult Schedules) with a dark blue background.

Ruta	http://localhost:8080/professorHome		
Componente	Objetivo	Característica Funcional	Característica No Funcional
Botón Cerrar Sesión	Cerrar sesión y redirigir a la pantalla de login	X	
Botón Mi Perfil	Redirigir a la pantalla en donde se muestra la información del docente	X	
Botón Inicio	Redirigir al menú home de docente	X	
Botón Ingresar Notas	Redirigir a la pantalla en la que podrá ingresar notas de los estudiantes asignados a un curso	X	
Botón Generar Actas	Redirigir a la vista generación de actas de cursos impartidos	X	
Botón Consultar Estudiantes Asignados	Redirigir a la pantalla en la que visualizará el listado de estudiantes asignados a un	X	

Ruta	http://localhost:8080/professorHome		
Componente	Objetivo	Característica Funcional	Característica No Funcional
	curso		
Botón Consultar Aprobaciones	Redirigir a la pantalla en la que se desplegarán los datos de estudiantes que aprobaron un curso	X	
Botón Consultar Actas	Redirigir a la pantalla en la que se desplegarán los datos de las actas generadas para un curso	X	
Botón Consultar Horarios	Redirigir a la pantalla en la que podrá consultar los horarios de cursos disponibles en un período específico	X	

## 2. Ingresar Notas

The screenshot shows the 'INGRESAR NOTAS' (Enter Grades) page. At the top, there's a header with the FUSAC logo and navigation links for 'Inicio', 'Mi Perfil', and 'Cerrar Sesión'. Below the header, the main form has the following fields:

- Código del curso: A dropdown menu showing '283 - Análisis y diseño de sistemas'.
- Ciclo: A dropdown menu showing '1S'.
- Año: A dropdown menu showing '2023'.
- Sección: A dropdown menu showing 'A'.
- Carnet: A text input field containing '20000000'.
- Nombre: A text input field containing 'Nombre completo'.
- Carrera: A text input field containing 'Sistemas'.
- Nota: A text input field containing '61'.
- Ingresar: A blue button with a checkmark icon.

Ruta	http://localhost:8080/professorEnterNotes		
Componente	Objetivo	Característica Funcional	Característica No Funcional
Botón Cerrar Sesión	Cerrar sesión y redirigir a la pantalla de login	X	
Botón Mi Perfil	Redirigir a la pantalla en donde se muestra la información del docente	X	
Botón Inicio	Redirigir al menú home de docente	X	
Botón Consultar	Obtener los datos del formulario y enviarlos al backend con una petición GET para obtener el listado de estudiantes asignados a un curso en un período y sección específicos	X	
Botones Ingresar	Obtener los datos del estudiante de la fila seleccionada y enviarlos junto con el valor del campo NOTA con una petición POST para asignarle una nota al estudiante	X	
Petición de Cursos	Obtener el listado de cursos que el profesor imparte mediante una petición GET y desplegarlos en el componente select		X

### 3. Generar Acta

PROFESSOR-GENERATE-REPORT

DATOS DEL CURSO

Código del Curso  
283 - Análisis y Diseño de Sistemas 1

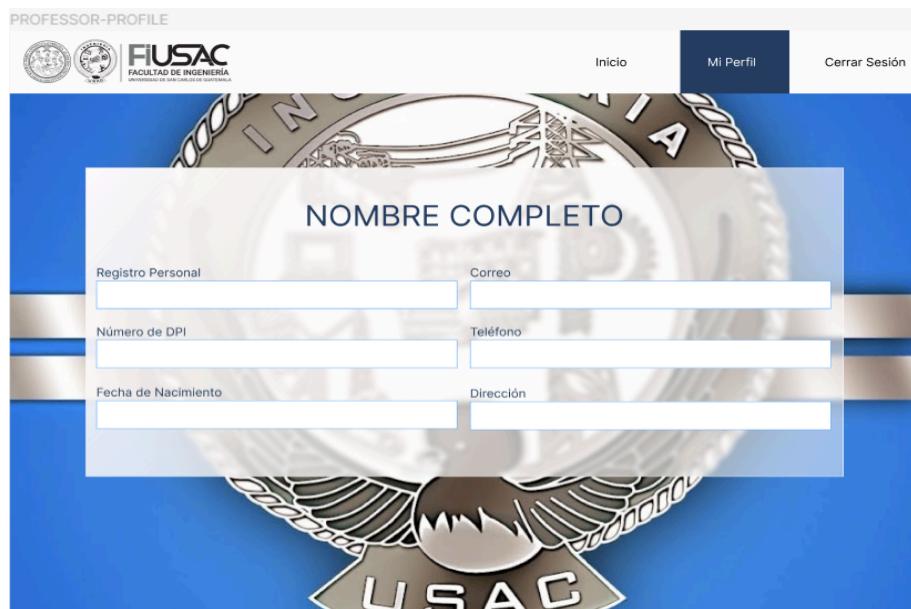
Ciclo

Sección

Generar Reporte

Ruta	http://localhost:8080/professorGenerateReport		
Componente	Objetivo	Característica Funcional	Característica No Funcional
Botón Cerrar Sesión	Cerrar sesión y redirigir a la pantalla de login	X	
Botón Mi Perfil	Redirigir a la pantalla en donde se muestra la información del docente	X	
Botón Inicio	Redirigir al menú home de docente	X	
Botón Generar Reporte	Obtener los datos del formulario y enviarlos al backend con una petición POST solicitar la creación de un acta para un curso impartido	X	
Petición de Cursos	Obtener el listado de cursos que el profesor imparte mediante una petición GET y desplegarlos en el componente select		X

#### 4. Perfil



Ruta	http://localhost:8080/professorProfile		
Componente	Objetivo	Característica Funcional	Característica No Funcional
Botón Cerrar Sesión	Cerrar sesión y redirigir a la pantalla de login	X	
Botón Mi Perfil	Redirigir a la pantalla en donde se muestra la información del estudiante	X	
Botón Inicio	Redirigir al menú home de estudiante	X	
Petición de Docente	Comunicarse con el backend con una petición GET para obtener toda la información del docente logueado y mostrarla en el formulario		X

## 5. Consultar Estudiantes Asignados

PROFESSOR-GET--ASSIGNED-STUDENTS

Ruta	http://localhost:8080/professorConsultStudents		
Componente	Objetivo	Característica Funcional	Característica No Funcional
Botón Cerrar Sesión	Cerrar sesión y redirigir a la pantalla de login	X	
Botón Mi Perfil	Redirigir a la pantalla en donde se muestra la información del docente	X	
Botón Inicio	Redirigir al menú home de docente	X	
Botón Consultar	Obtener los datos del formulario y enviarlos al backend con una petición GET para obtener el listado de estudiantes asignados a un curso en un período y sección específicos y desplegarlos en la tabla	X	
Petición de Cursos	Obtener el listado de cursos que el profesor imparte mediante una petición GET y desplegarlos en el componente select		X

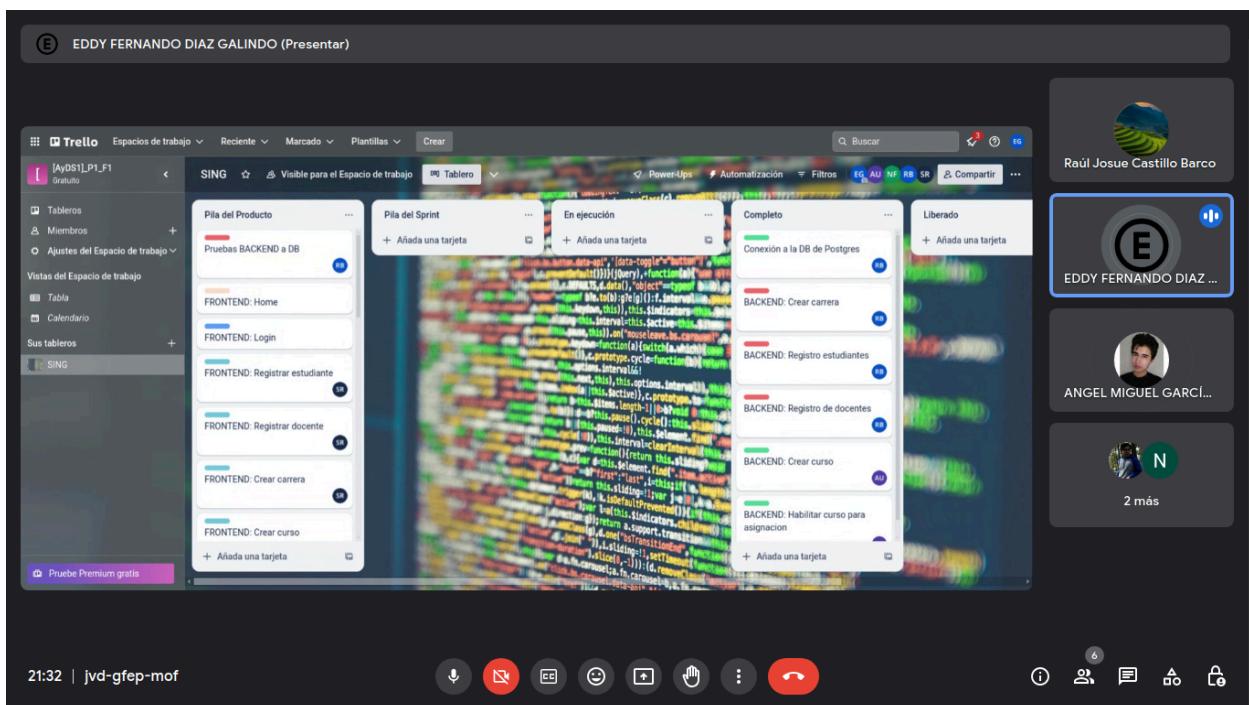
## REUNIONES DE DAILY SCRUM

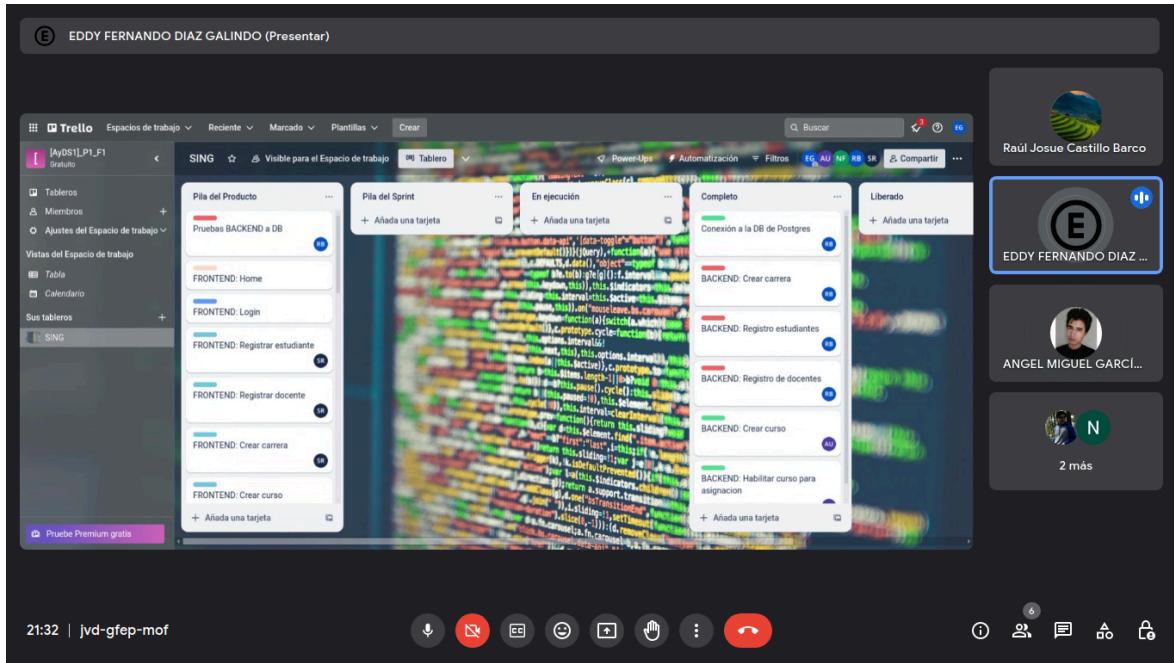
Daily 20 de marzo de 2024

Objetivo: Determinar los requerimientos del product owner en base a las historias de usuario para asignar cada uno de los product backlog de la aplicación.

¿Hay algún impedimento?

Como el product owner ya tenía definidos los requerimientos en base a las consultas al usuario, entonces para este primer daily no se tuvo ningún impedimento para llevarlo a cabo.





## Daily 21 de marzo de 2024

¿Qué hicimos ayer?

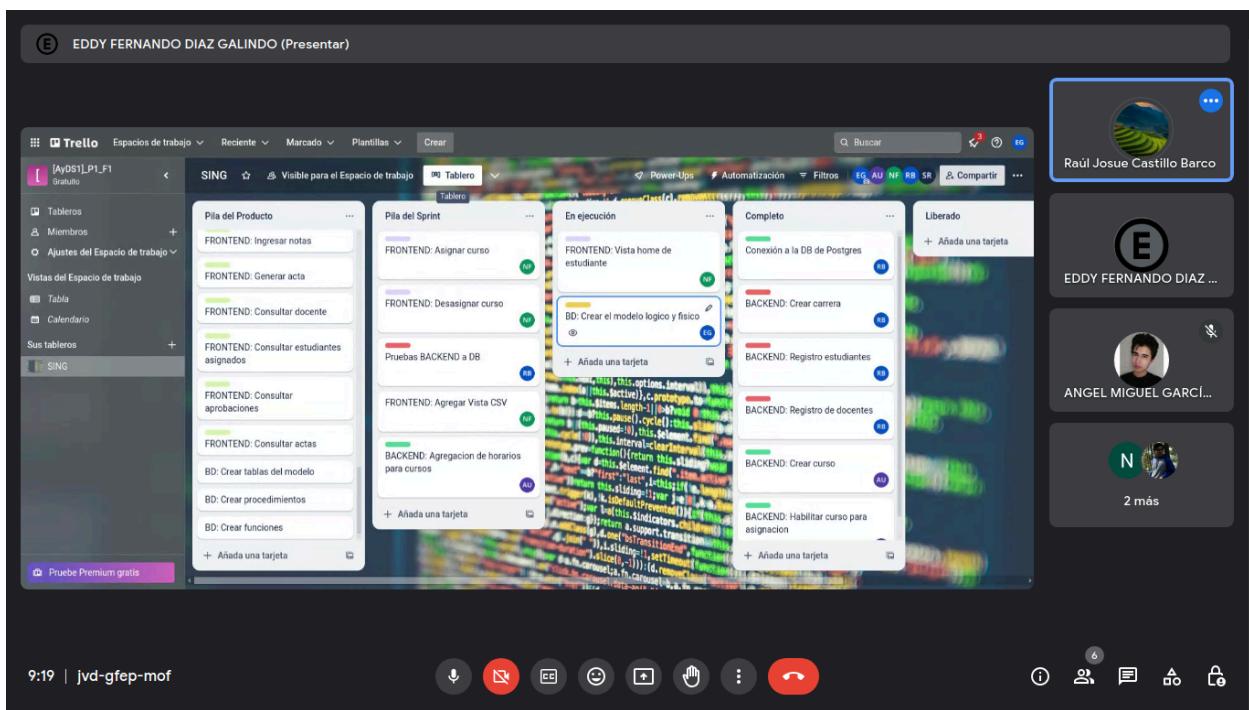
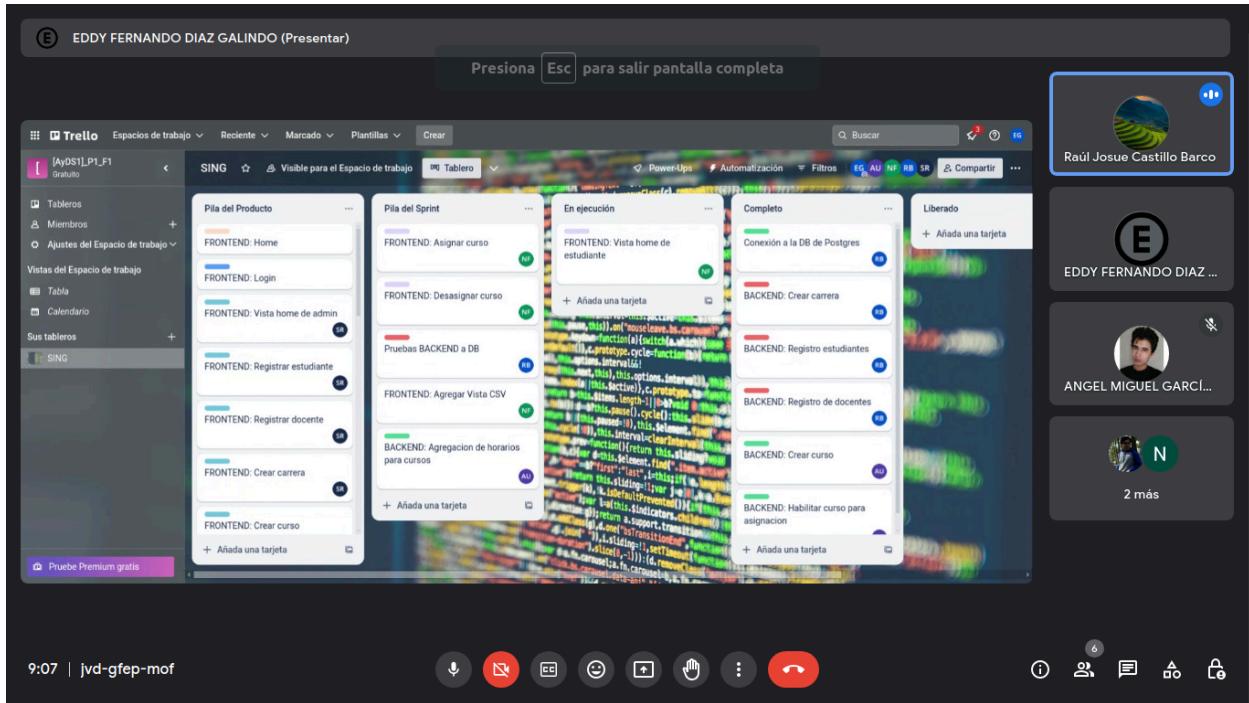
Como ya se tenían determinadas las funcionalidades en el product backlog, se asignaron las tareas de cada scrum developer y se determinaron los periodos de cada sprint.

¿Qué haremos hoy?

Una vez asignadas las tareas para cada scrum developer, se iniciará con el primer sprint. Así mismo, se definió el formato para los horarios en la base de datos, así como el formato de la data en frontend.

¿Existe algún impedimento?

La base de datos fue la primera en definirse, entonces el equipo de backend esperó a que se completara dicho sprint y de igual manera el equipo de frontend esperó a que se finalizara con el sprint de backend.



## Daily 22 de marzo de 2024

¿Qué hicimos ayer?

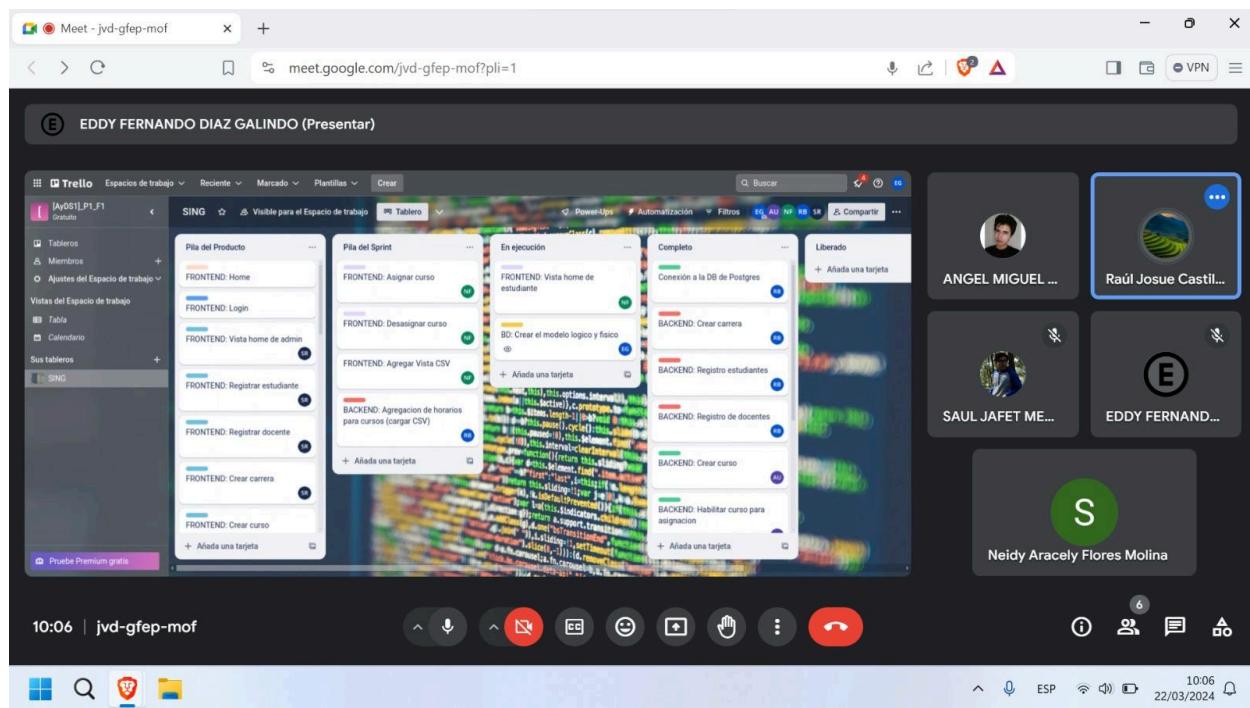
Se realizaron algunos avances con el sprint del diseño de postgres así como el de registro y login. Así mismo se corrigieron algunos formatos en la base de datos de Postgresql

## ¿Qué haremos hoy?

Continuaremos con el sprint Diseño de postgres y Login y registro. Nos basaremos en las funcionalidades propuestas por el Product Owner la cual una de ellas era crear una vista para cada usuario que inicie sesión.

## ¿Existe algún impedimento?

Esperamos las correcciones del DBA para realizar las distintas implementaciones en el backend, el equipo de frontend mencionó un error al realizar la petición POST al backend al crear usuarios, por lo que se verificarán esos errores.



## Daily 23 de marzo de 2024

¿Qué hicimos ayer?

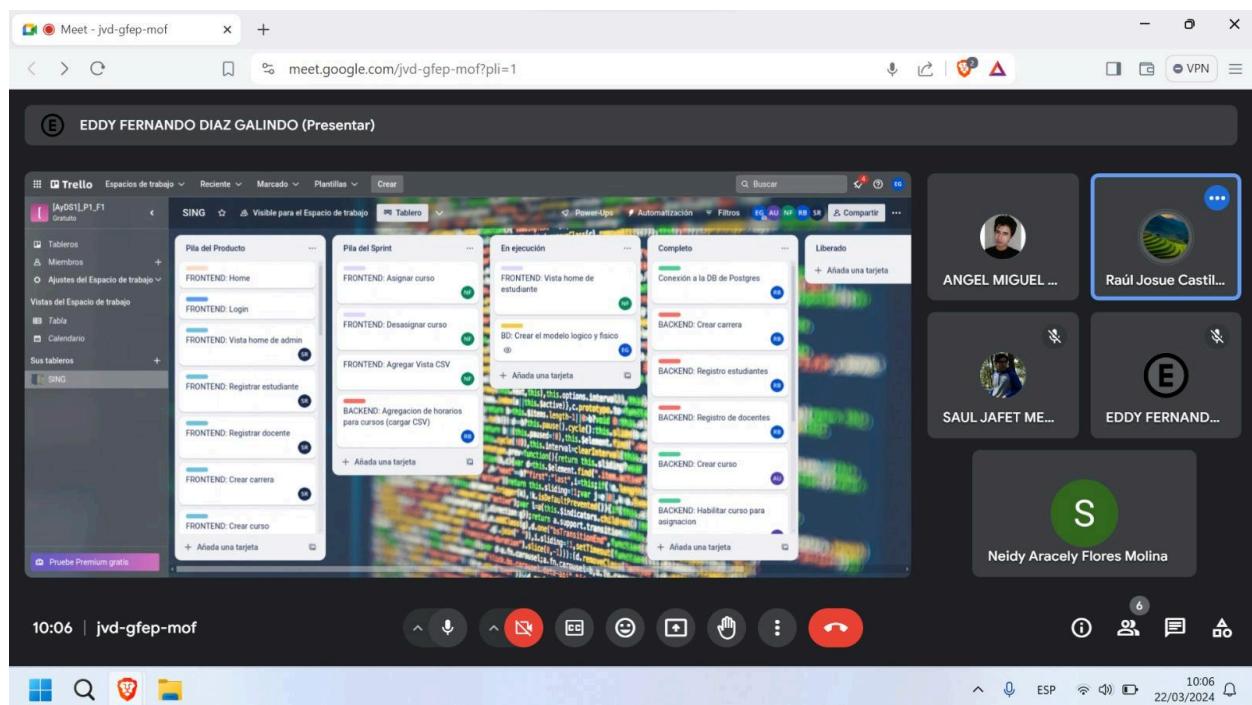
Se mencionaron las correcciones para el error sobre la creación de usuarios que arrojaba el backend al realizar dicha petición.

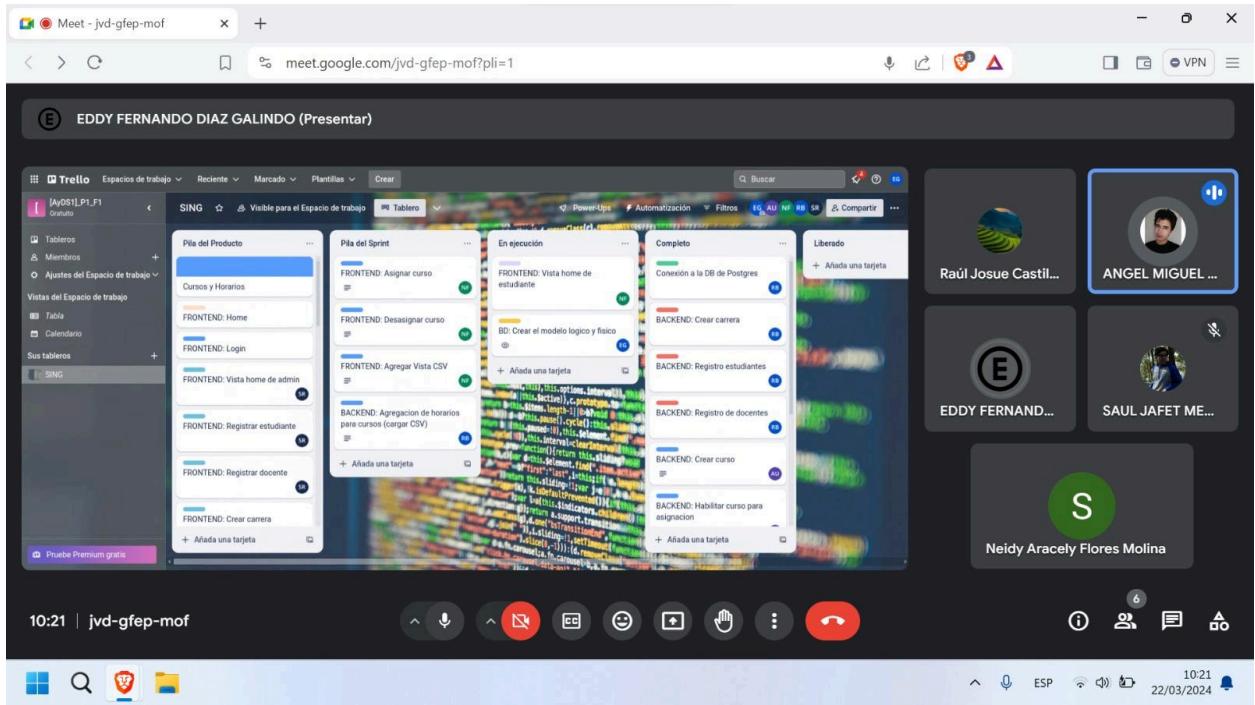
¿Qué haremos hoy?

El equipo de frontend continuará con la creación de la vista estudiante. De la misma manera se agregaron nuevas funcionalidades tales como tablas, funciones y procedimientos a la base de datos, para que el equipo de backend pueda hacer uso de ellas y posteriormente el equipo de frontend pueda implementarlo.

¿Existe algún impedimento?

Como ya se corrigieron los errores existentes, cada equipo puede trabajar en sus tareas para el avance del sprint.





## Daily 25 de marzo de 2024

¿Qué hicimos ayer?

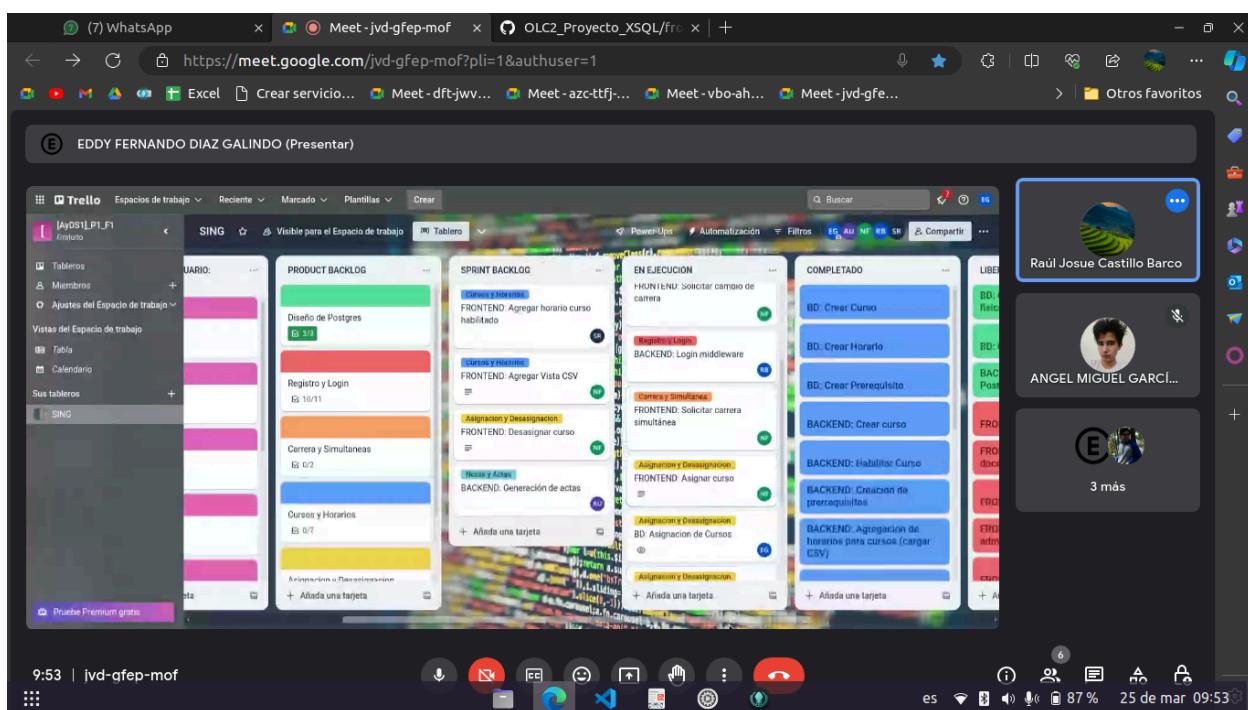
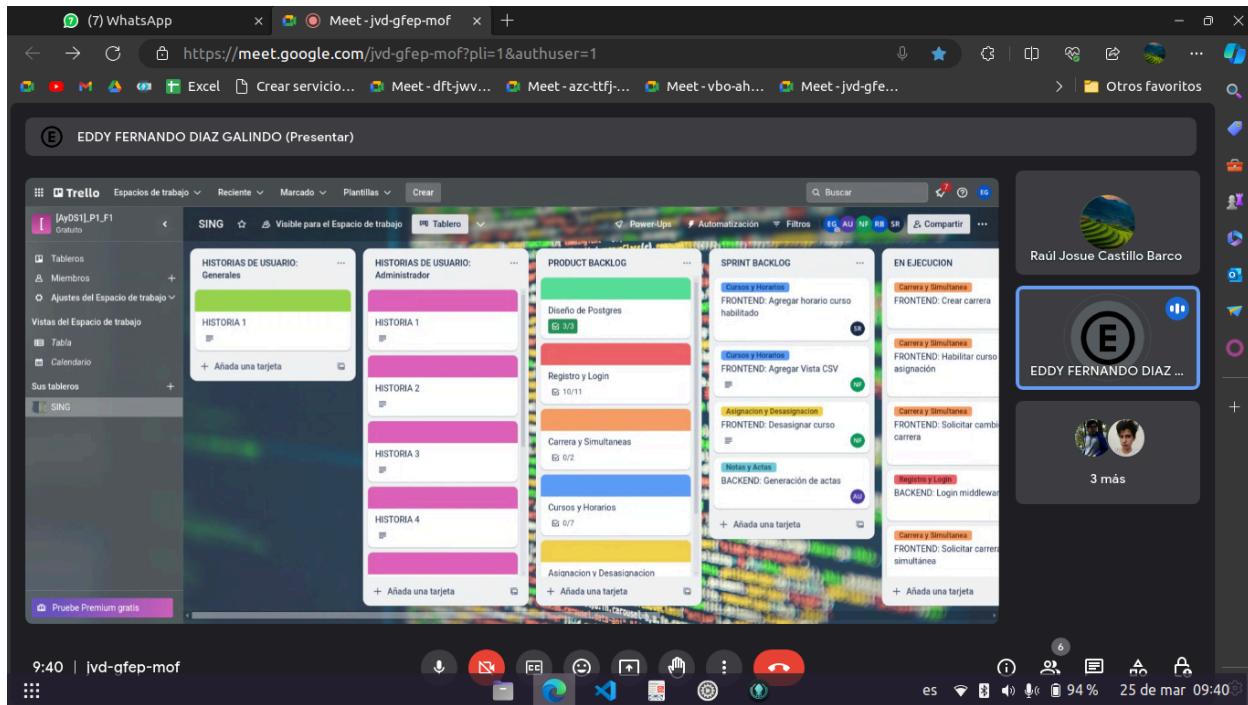
Cada equipo trabajó en sus tareas correspondientes para dar por finalizado los primeros sprint, los cuales fueron, el diseño de postgres (funcionalidades que cubrían los requerimientos de dicho sprint), así como el de registro y login.

¿Qué haremos hoy?

Se inició con un nuevo sprint el cual es el de Cursos y horarios así como Asignación y Desasignación. La funcionalidad de Cursos y horarios según lo recopilado por el producto owner en las historias de usuario, permite la creación de un curso nuevo y asignar un horario. Mientras que el segundo permite la asignación de un estudiante si y solo si el curso fue habilitado.

¿Existe algún impedimento?

No, debido a que es el inicio de un nuevo sprint.



## Daily 26 de marzo de 2024

¿Qué hicimos ayer?

Se dio inicio a un nuevo sprint por lo que se discutió sobre la distribución de tareas para la implementación de una nueva funcionalidad requerida.

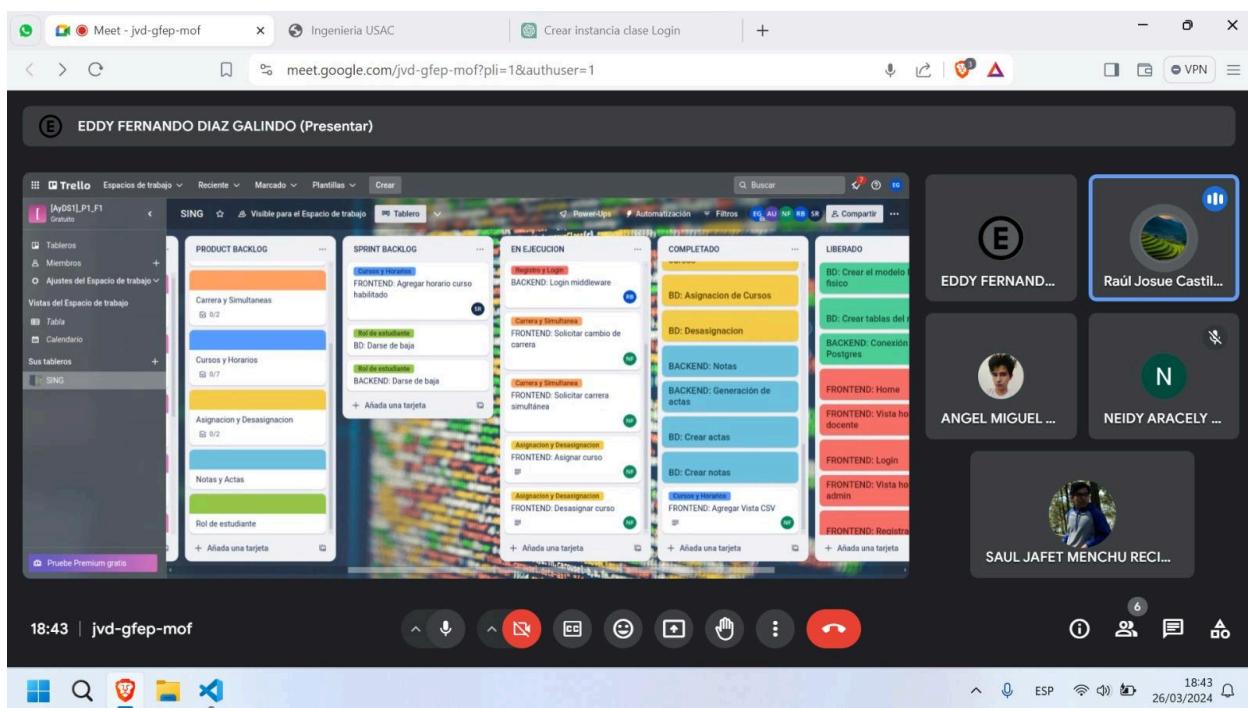
## ¿Qué haremos hoy?

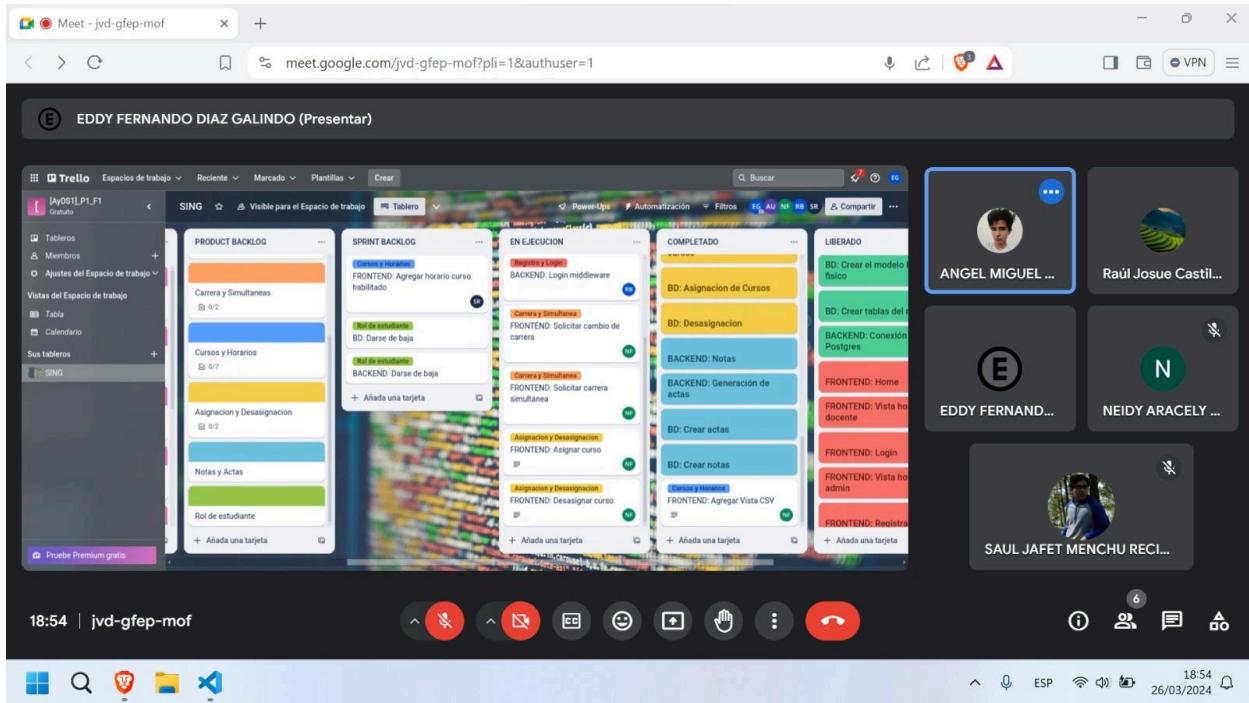
En la base de datos se discutió sobre la estructura de los datos para las nuevas implementaciones, y se coordinó con el equipo de backend para el formato de la data en json que llevará cada una de las respuestas a las peticiones hechas desde el cliente.

Así mismo se trabajará en la vista para habilitar cursos para su posterior asignación por parte de los estudiantes.

## ¿Existe algún impedimento?

Según lo mencionado por el equipo no se han presentado conflictos o problemas al momento de conectar todos los sistemas.





## Daily 27 de marzo de 2024

¿Qué hicimos ayer?

Se continuaron con las nuevas tablas, funciones y procedimientos de la base de datos, así como la continuación de cada una de las tareas de los demás equipos. Todos trabajando sobre el mismo sprint, el cual era de Cursos y horarios.

También se completó la implementación de asignar notas por parte del docente del lado del cliente.

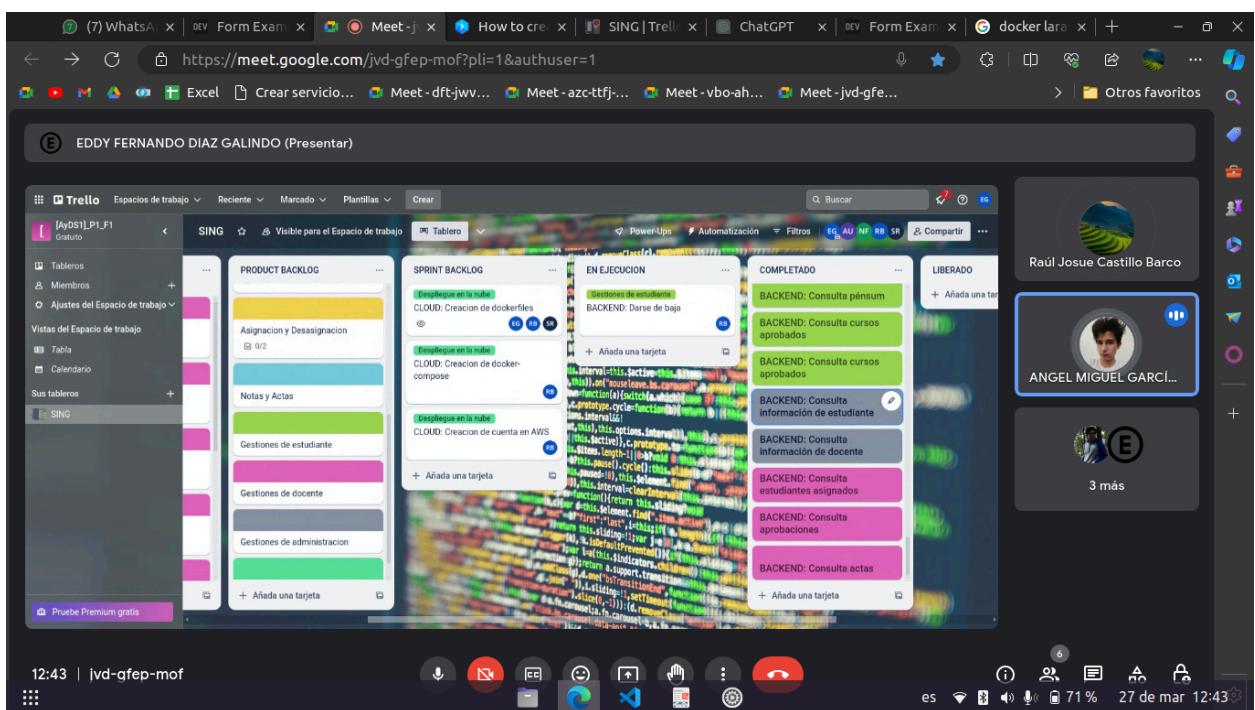
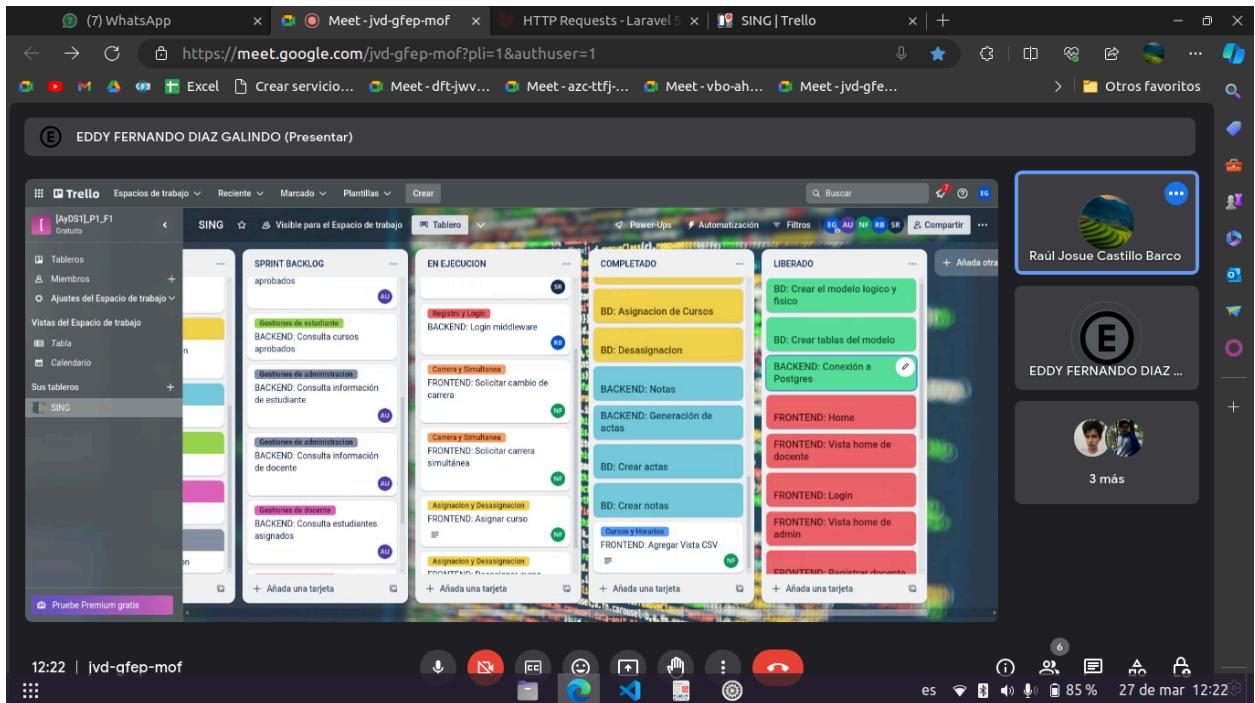
¿Qué haremos hoy?

Se corregirán errores mencionados durante el daily meet sobre el formato json de las peticiones por parte del equipo de frontend hacia el servidor de golang.

Se iniciarán nuevas funciones del sprint backlog tales como consultar de estudiantes asignados a un curso determinado. Mientras que el equipo conectará los 2 sistemas, servidor y cliente.

¿Existe algún impedimento?

Según lo informado por el equipo, no se encontró ningún impedimento para iniciar sobre otro sprint.



## Daily 28 de marzo de 2024

¿Qué hicimos ayer?

Se corrigieron los errores en el servidor para las peticiones del lado del cliente, el formato json quedó solucionado.

El equipo de frontend mientras trabajó en el inicio de un nuevo sprint sobre las gestiones del docente para que este pueda realizar cada una de sus funciones.

En la base de datos se corrigieron errores en el formato de la fecha, ya que mostraba un formato inadecuado para el cliente.

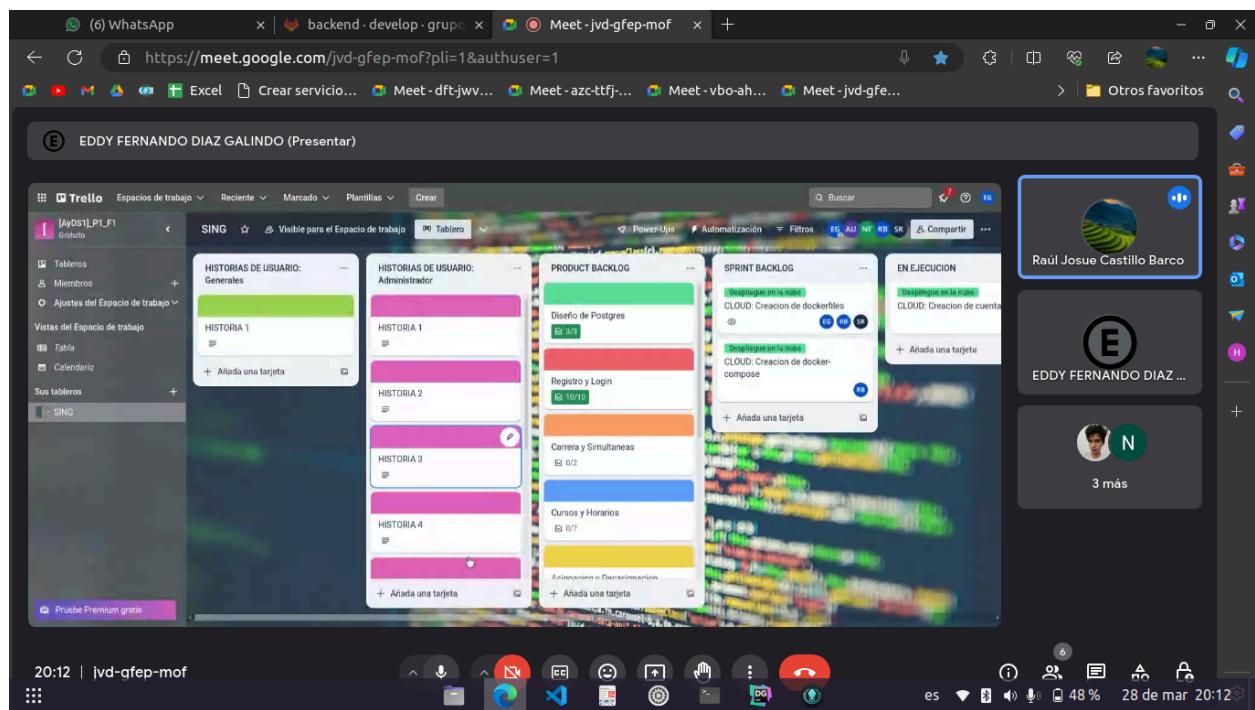
¿Qué haremos hoy?

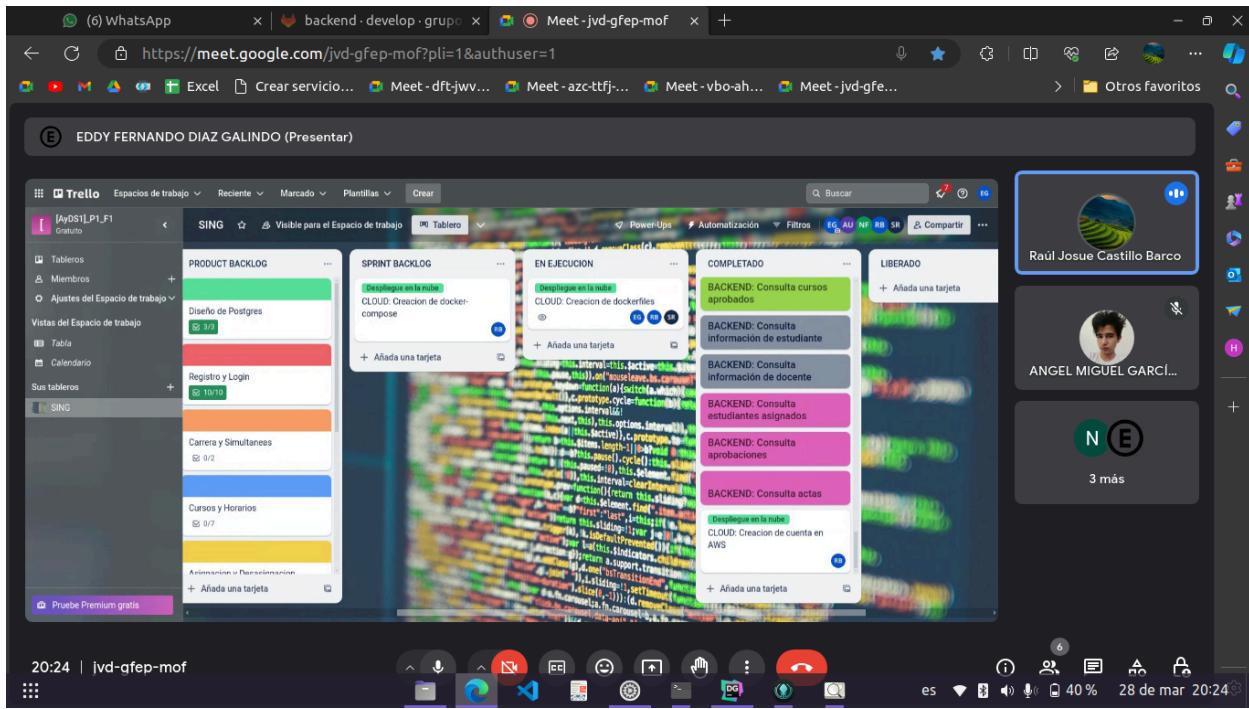
Se iniciarán con las pruebas unitarias por parte del equipo de backend, mientras que el equipo de frontend termina con cada una de las tareas para dar por finalizado el sprint.

En la base de datos se iniciarán de igual manera las pruebas para “dockerizar” e implementarla posteriormente en docker compose.

¿Existe algún impedimento?

No existe ningún impedimento.





## Daily 29 de marzo de 2024

¿Qué hicimos ayer?

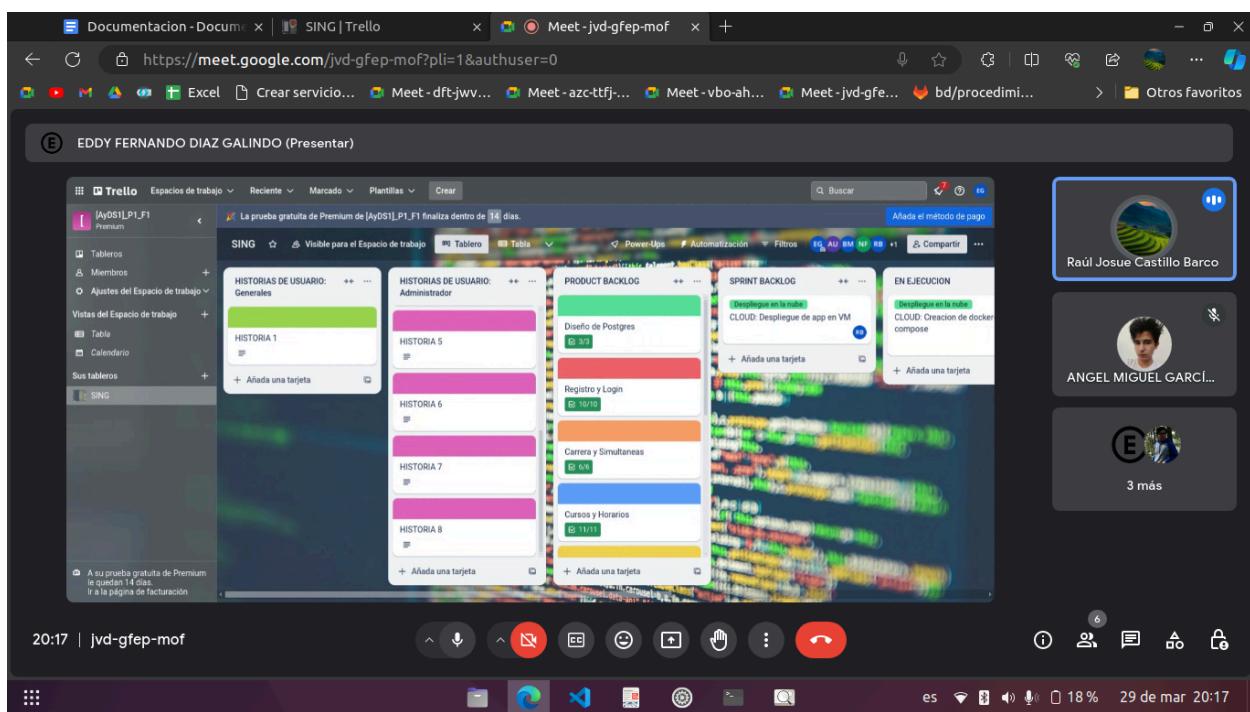
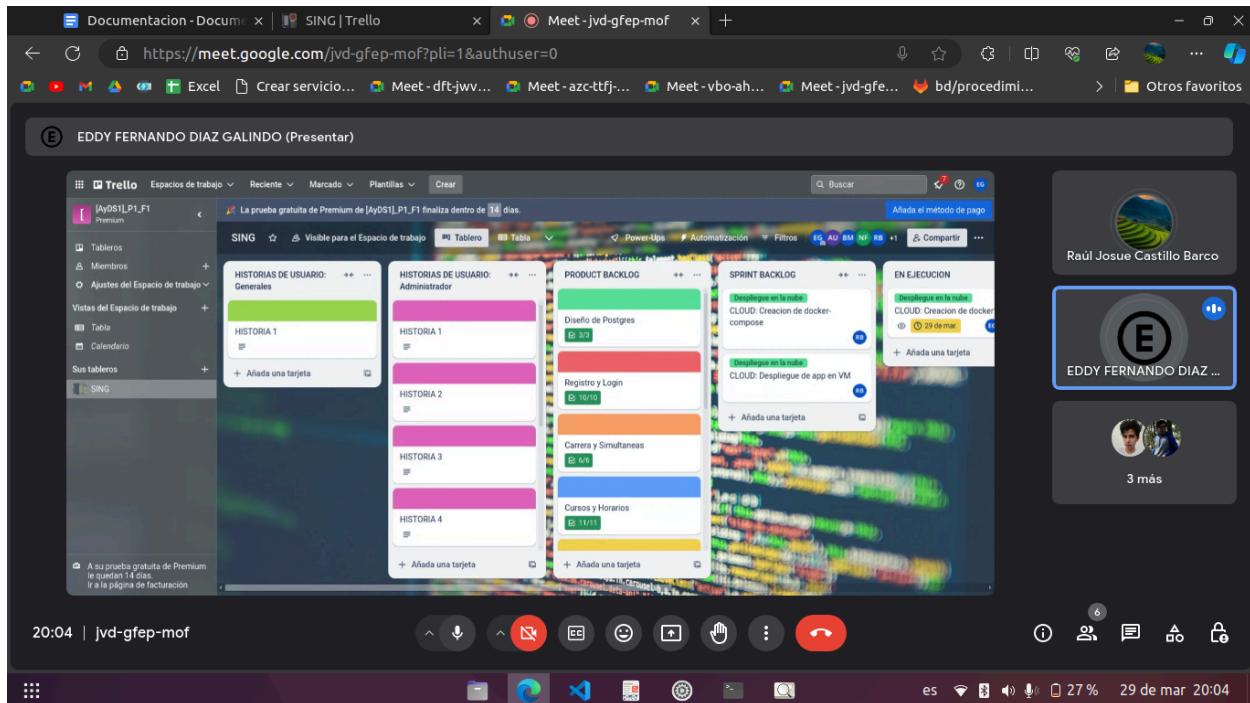
Se iniciaron las pruebas unitarias por parte del equipo backend obteniendo un 87% de test coverage. También se iniciaron las pruebas en conjunto para la unión de los 3 sistemas, cliente, servidor y base de datos

¿Qué haremos hoy?

El equipo de backend y frontend iniciarán las pruebas para “dockerizar” cada uno de sus sistemas para lograr así el despliegue de la aplicación en entorno de la nube.

¿Existe algún impedimento?

No existe ningún impedimento.



## Daily 30 de marzo de 2024

¿Qué hicimos ayer?

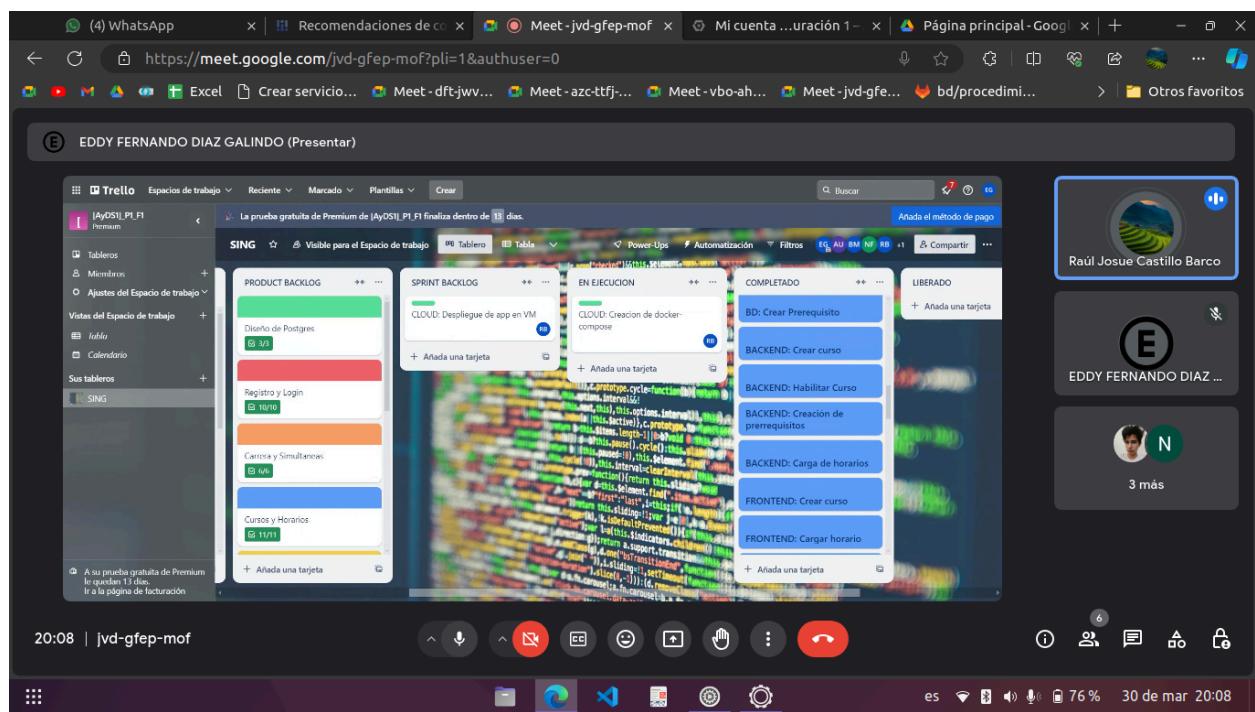
Cada equipo realizó el Dockerfile correspondiente para llevar a cabo el docker compose por medio del archivo .yml.

¿Qué haremos hoy?

Se llevará a cabo el despliegue de la aplicación en el entorno de la nube, por medio de docker compose y una máquina virtual de Amazon Web Services.

¿Existe algún impedimento?

No existe ningún impedimento.



Eddy Fernando Diaz Galindo (Presentar)

Trello Espacios de trabajo Reciente Marcado Plantillas Crear

[AyDSI]\_P1\_F1 Premium

La prueba gratuita de Premium de [AyDSI]\_P1\_F1 finaliza dentro de 13 días.

Añade el método de pago

SING Tablero Tabla Power-Ups Automatización Filtros EG AU BM NF RB +1 Compartir ...

PRODUCT BACKLOG SPRINT BACKLOG EN EJECUCIÓN COMPLETADO LIBERADO

+ Añada una tarjeta

Dirección de Postgres Registro y Login Cámara y Simultáneas Cursos y Horarios

3/3 10/10 4/6 11/11

+ Añada una tarjeta

CLOUD: Despliegue de app en VM

+ Añada una tarjeta

BD: Consulta aprobaciones BACKEND: Consulta estudiantes asignados

BACKEND: Consulta aprobaciones

BACKEND: Consulta actas

CLOUD: Creación de cuenta en AWS

CLOUD: Creación de dockerfiles

CLOUD: Creación de docker-compose

+ Añada una tarjeta

Raúl Josue Castillo Barco EDDY FERNANDO DIAZ ... ANGEL MIGUEL GARCÍ... N 2 más

20:18 | jvd-gfep-mof

6

The screenshot shows a Trello board titled "SING". The board has several columns: PRODUCT BACKLOG, SPRINT BACKLOG, EN EJECUCIÓN, COMPLETADO, and LIBERADO. The EN EJECUCIÓN column contains several cards related to database queries and cloud operations. To the right of the board, there's a sidebar with user profiles for "Raúl Josue Castillo Barco", "EDDY FERNANDO DIAZ", "ANGEL MIGUEL GARCÍ...", and "N". The bottom of the screen shows a dark interface with a timestamp "20:18" and some user icons.

## EVENTOS SCRUM

SPRINT	OBJETIVO DEL SPRINT	SPRINT PLANNING
Primer Sprint (20 - 25 de Marzo)	Implementar la lógica para asignar roles a los usuarios registrados, garantizando que se puedan distinguir entre estudiantes, docentes y administradores.	<p>División de tareas entre el equipo de desarrollo, asignando responsabilidades para la implementación de las vistas, rutas, controladores y modelos necesarios.</p> <p>Estimación del esfuerzo necesario para completar cada tarea y asignación de tiempos límite para su finalización.</p> <p>Revisión del backlog del producto y priorización de las tareas relacionadas con la autorización y gestión de perfiles de usuario.</p>
Segundo Sprint (26 - 30 de Marzo)	Realizar pruebas exhaustivas para garantizar que todas las funcionalidades estén implementadas correctamente y que el sistema sea seguro y fácil de usar.	<p>Revisión de las funcionalidades específicas que deben desarrollarse para cada rol de usuario (estudiante, docente, administrador) y su priorización.</p> <p>Estimación del tiempo y los recursos necesarios para completar cada tarea, teniendo en cuenta las lecciones aprendidas del sprint anterior y las mejoras sugeridas.</p>

El equipo ha definido el siguiente conjunto de criterios que deben cumplirse para considerar que una tarea o historia de usuario está completada al finalizar un sprint:

- Código Escrito: El código relacionado con la tarea o historia de usuario ha sido escrito y revisado por al menos un miembro del equipo.

- Pruebas Unitarias: Se han escrito pruebas unitarias para verificar el correcto funcionamiento del código implementado.
- Revisión de Código: El código ha sido revisado por al menos un miembro del equipo diferente al autor original, y se han abordado todas las sugerencias y problemas identificados durante la revisión.
- Documentación: Se ha actualizado cualquier documentación relevante, como comentarios en el código, documentación técnica o de usuario, y se ha mantenido al día.

## **SPRINT REVIEW - PRIMER SPRINT REVIEW**

Fecha: 26 de Marzo del 2024

Participantes:

- Eddy Díaz - Scrum Master
- Raul Castillo - Product Owner
- Neidy Flores, Angel García, Saul Menchu – DevTeam

Logros:

Para este sprint se logró el incremento esperado, es decir, desarrollar la lógica y la implementación de los roles establecidos (Administrador, estudiantes y docentes) dentro del producto.

Así también se definieron diferentes entidades como carreras, cursos, horarios y más lógica necesaria establecida en el funcionamiento de cada rol.

Para tener la distinción de roles también se implementó el login de usuarios, así de esa manera poder definir las rutas y vistas que corresponden a cada uno de los roles.

Conclusión:

Se logró cumplir con el objetivo del sprint y presentar un incremento satisfactorio según los criterios definidos por el Product Owner en base a las necesidades y requerimientos del cliente definidas en el Product Backlog.

## **RETROSPECTIVA DEL SPRINT - PRIMER SPRINT DE DESARROLLO DE LA APLICACIÓN**

Fecha: 26 de Marzo del 2024

Duración: 10 mins

Participantes:

- Eddy Díaz - **Scrum Master**
- Raul Castillo - **Product Owner**
- Neidy Flores, Angel García, Saul Menchu - **DevTeam**

Objetivo de la Retrospectiva:

Identificar lo que salió bien, lo que no salió tan bien y las oportunidades de mejora para el equipo durante el primer sprint de desarrollo de la aplicación.

Lo que Salió Bien:

Logramos establecer la estructura básica del proyecto y definir las tecnologías a utilizar.

El equipo demostró una gran colaboración y comunicación durante todo el sprint. Se completaron las tareas asignadas dentro del tiempo estimado. La retroalimentación inicial del Product Owner fue positiva.

Lo que No Salió Tan Bien:

Hubo cierta confusión inicial sobre los requisitos del cliente, lo que resultó en un tiempo adicional dedicado a la clarificación.

Algunos miembros del equipo experimentaron dificultades técnicas con la conexión a la base de datos.

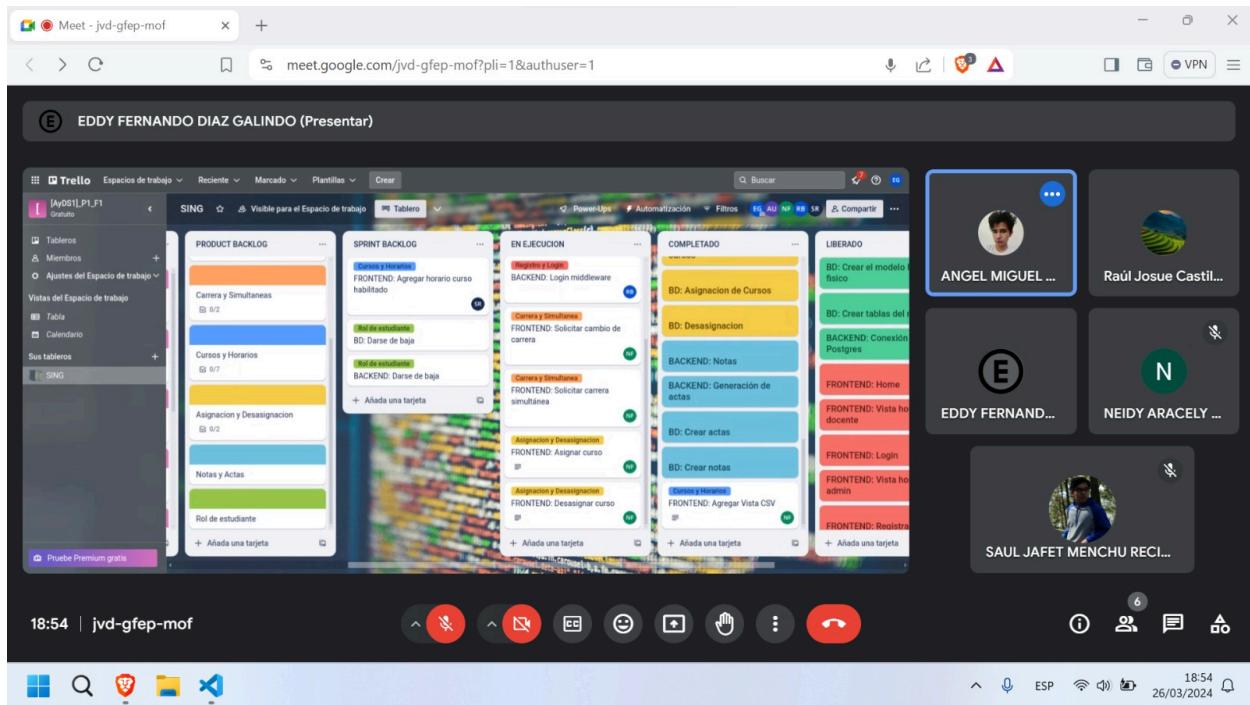
La estimación de tiempo para algunas tareas resultó ser inexacta, lo que provocó retrasos en la entrega.

Oportunidades de Mejora:

Refinar el proceso de estimación de tiempo para tareas futuras, basándonos en el aprendizaje de este sprint.

## Observaciones Finales:

En general, el equipo está satisfecho con el progreso realizado durante el primer sprint y está comprometido a abordar las áreas identificadas para mejorar en el próximo sprint.



## **SPRINT REVIEW - SEGUNDO SPRINT REVIEW**

Fecha: 30 de Marzo del 2024

Participantes:

- Eddy Díaz - **Scrum Master**
- Raul Castillo - **Product Owner**
- Neidy Flores, Angel García, Saul Menchu – **DevTeam**

Logros:

Se alcanzó el incremento esperado, este incremento terminado implica el desarrollo de la lógica y la implementación de las diferentes funcionalidades requeridas en cada uno de los roles. También asegurar el funcionamiento correcto de dichas funcionalidades.

Además se hizo el despliegue en la nube de AWS para poner en funcionamiento la aplicación y con ello cumplir con el objetivo del producto de entregar una aplicación renovada para la facultad de ingeniería.

Conclusión:

Se logró entregar el incremento esperado, completando lo establecido en el sprint backlog y cumpliendo el objetivo principal del producto.

**RETROSPECTIVA DEL SPRINT - SEGUNDO SPRINT DE DESARROLLO DE LA  
APLICACIÓN**

Fecha: 30 de Marzo del 2024

Duración: 10 mins

Participantes:

- Eddy Díaz - **Scrum Master**
- Raul Castillo - **Product Owner**
- Neidy Flores, Angel García, Saul Menchu - **DevTeam**

Objetivo de la Retrospectiva:

Reflexionar sobre el despliegue de la aplicación en AWS y Docker durante el segundo sprint e identificar oportunidades de mejora para futuros despliegues.

Lo que Salió Bien:

Logramos implementar la aplicación en contenedores Docker de manera exitosa.  
El despliegue en AWS se realizó sin problemas importantes.  
El equipo trabajó de manera colaborativa para resolver los desafíos técnicos relacionados con el despliegue.  
La documentación del proceso de despliegue fue útil y completa.

Lo que No Salió Tan Bien:

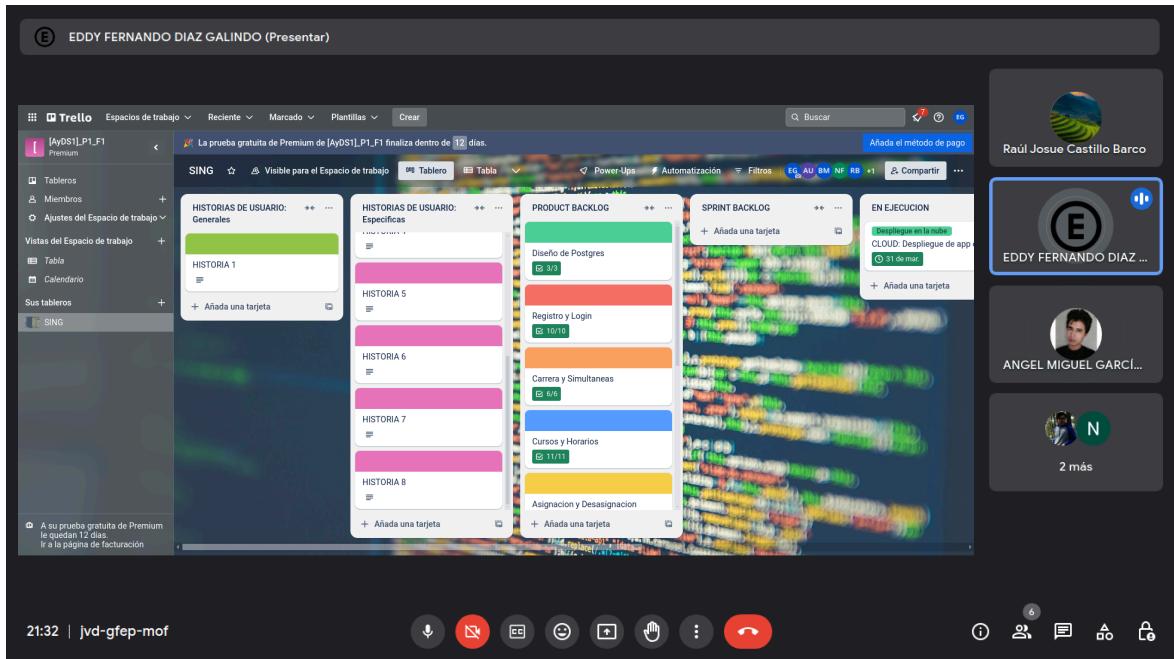
Surgieron algunos problemas de configuración durante el despliegue en AWS, lo que provocó pequeños retrasos.  
Hubo cierta confusión en torno a la gestión de las imágenes de Docker y la configuración de los contenedores.

Oportunidades de Mejora:

Investigar y adoptar herramientas de automatización de despliegue que puedan simplificar y agilizar el proceso en futuros sprints.

Observaciones Finales:

A pesar de los desafíos encontrados durante el despliegue en AWS y Docker, el equipo está comprometido a mejorar su competencia técnica y a implementar soluciones más eficientes en futuros despliegues.



## ANEXOS

Link a trello:

<https://trello.com/invite/b/TkwdUFFfJ/ATTIb286a66e50664863d63dfe6f190ab97a1B65B5ED/sing>

Link al GitFlow:

[https://drive.google.com/file/d/19ewtSZ6i4Xx0UBve\\_pvJSfOBF3Mimkxf/view?usp=sharing](https://drive.google.com/file/d/19ewtSZ6i4Xx0UBve_pvJSfOBF3Mimkxf/view?usp=sharing)

Link al Repositorio:

[https://gitlab.com/grupo\\_ayd/AyDS1\\_Proyecto1\\_F1\\_G11.git](https://gitlab.com/grupo_ayd/AyDS1_Proyecto1_F1_G11.git)

Link a Figma:

<https://www.figma.com/file/5I1IPnE1R1FHJe5Gq6IXKJ/FASE-1?type=design&node-id=0%3A1&mode=design&t=Q3j3owG27HKoNcqq-1>