



Universidad Austral de Chile

Facultad de Ciencias de la Ingeniería
Escuela de Ingeniería Civil en Informática

SCOREMED: REGISTRO PARA PRACTICANTES DEL AREA DE LA SALUD GAMIFICADO

Proyecto para optar al título de
Ingeniero Civil en Informática

PROFESOR PATROCINANTE:
CHRISTIAN A. LAZO RAMÍREZ
INGENIERO GESTION INFOTMÁTICA
D.E.A INGENIERÍA TELEMATICA
DOCTOR EN INGENIERÍA TELEMATICA

PROFESOR CO-PATROCINANTE
SCARLETTE F. LETELIER SANTIBÁÑEZ
MATRONA
MAGÍSTER EN INNOVACIÓN DE LA DOCENCIA
UNIVERSITARIA EN CIENCIAS DE LA SALUD
DIPLOMADO EN GESTIÓN DE EMPRESAS

PROFESOR INFORMANTE
MIRKO GUEREGAT HENRIQUEZ
INGENIERO CIVIL EN INFORMÁTICA

JUAN GUILLERMO CONTRERAS SEPÚLVEDA

VALDIVIA – CHILE
2020

ÍNDICE

ÍNDICE	I
ÍNDICE DE TABLAS	III
ÍNDICE DE FIGURAS	IV
RESUMEN.....	V
ABSTRACT.....	V
1. INTRODUCCIÓN	1
1.1 Objetivo general	1
1.2 Objetivos específicos.....	1
2. MARCO TEÓRICO	2
2.1 Practica de estudiantes	2
2.2 Serious Games.....	3
2.3 Gamificación	5
2.4 Trabajo en equipo e individual, actividades y/o objetivos	6
2.4.1 Trabajo en equipo.....	6
2.4.2 Trabajo individual	7
2.4.3 Objetivo.....	7
2.5 Impacto del Proyecto.....	8
2.5.1 Corto plazo	8
2.5.2 Mediano plazo	8
2.5.3 Largo plazo	8
2.6 Herramientas	8
2.6.1 Frontend	8
2.6.2 Backend.....	12
2.6.3 Base de datos.....	13
2.6.3.1 Base de datos SQL o relacionales	13
2.6.3.2 Base de datos NoSQL o No relacionales.....	13
3. ARQUITECTURA Y SOLUCION	15
3.1 Descripción de la problemática	15
3.2 Solución propuesta	15
3.3 Metodología	16
3.4 Diagramas de secuencia	17
3.4.1 Rol Profesor.	22
3.4.2 Rol Tutor	26
3.4.3 Rol Estudiante	27
4. SERVIDOR.....	31
4.1 Características	31
4.2 Dependencias	31
4.2.1 Angular/cli	31
4.2.1 Typescript.....	31
4.2.3 Angular/core.....	31
4.2.4 Angular/router.....	32
4.2.5 Angular/forms	32
4.2.6 Chartjs	32
4.2.7 Rxjs	32
4.2.8 Php	32
4.2.9 Composer	32
4.2.10 Symfony	32
4.2.11 GIT	33
4.2.12 Lexik/jwt-authentication-bundle	33
4.3 Instalación del proyecto	33

5. PLATAFORMA WEB	35
5.1 Vista Profesor.....	36
5.2 Vista Estudiante	41
6.PROBLEMAS Y SOLUCIONES	45
6.1 Problema y solución 1	45
6.2 Problema y solución 2	45
6.3 Problema y solución 3	46
7. TESTING	48
7.1 Expectativas	49
7.1.1 Expectativas profesor	49
7.1.2 Expectativas estudiante	49
7.2 Resultados	49
8. CONCLUSION	55
9.REFERENCIAS	56
ANEXOS	58
Anexo A: Entidades.	58
User.php	58
Subject.php.....	63
Mision.php	68
ActivityData.php	70
Anexo B: Manual de usuario.....	73

ÍNDICE DE TABLAS

Tabla	Pagina
Tabla 1: Extracto de la hoja que utilizan	3
Tabla 2 : Base de datos relacional vs base de datos no relacional.	14
Tabla 3: Características del servidor	31
Tabla 4: Repositorios	33
Tabla 5: Acciones para instalar el frontend	34
Tabla 6: Acciones para el backend	34

ÍNDICE DE FIGURAS

Figura	Pagina
FIGURA 1: Resultado de encuesta de stackoverflow	9
FIGURA 2: Tendencias NPM	10
FIGURA 3 : Curva de aprendizaje	11
FIGURA 4: Base de datos del proyecto	14
FIGURA 5: Imagen de Trello	17
FIGURA 6: Diagrama Sistema	18
FIGURA 7: Diagrama Registro.....	18
FIGURA 8: Diagrama Login.....	22
FIGURA 9: Crear una asignatura.....	23
FIGURA 10 : Diagrama de edición de una asignatura.....	24
FIGURA 11: Diagrama de creación de una misión.....	24
FIGURA 12: Diagrama de vinculación de una misión a una asignatura.....	25
FIGURA 13: Diagrama de validación/rechazo de las actividades	26
FIGURA 14: Diagrama de la vista inicial del tutor.....	27
FIGURA 15 : Diagrama de vinculación de un estudiante a un curso.....	28
FIGURA 16: Diagrama de los cursos de un estudiante.....	28
FIGURA 17: Diagrama para agregar una actividad	29
FIGURA 18: Diagrama para ver el progreso del estudiante	29
FIGURA 19: Clonación de repositorio	33
FIGURA 20: Home-page	35
FIGURA 21: Login-page	35
FIGURA 22: Register-page.....	36
FIGURA 23: Vista post login del profesor.....	37
FIGURA 24: Crear Asignatura	37
FIGURA 25: Edición de asignatura	38
FIGURA 26: Creación de actividad	39
FIGURA 27: Vista para agregar actividades.....	39
FIGURA 28: Validación de actividades.....	40
FIGURA 29: Ver el curso.	40
FIGURA 30: Grafico de actividad	41
FIGURA 31: Resumen de una asignatura	41
FIGURA 32: Lista de cursos.....	42
FIGURA 33: Vista de cursos inscritos	42
FIGURA 34: Vista Ingresar Actividad.....	43
FIGURA 35: Estadísticas del estudiante	43
FIGURA 36: Récord de actividades.....	44
FIGURA 37 :Feedback Editar.....	50
FIGURA 38: Feedback de la Home-page.	51
FIGURA 39: Feedback Actividades.....	51
FIGURA 40 :Feedback validación.	51
FIGURA 41: Feedback Home-page estudiante.....	52
FIGURA 42: Feedback Mis Cursos.	52
FIGURA 43: Feedback Ver Curso.	53
FIGURA 44: Feedback Ingresar actividad.	53
FIGURA 45: Feedback Récord de actividades.	53

RESUMEN

Dentro de la universidad, hay diferentes tipos de actividades que un profesor debe realizar, siendo las más comunes la planificación de una clase, la realización de ésta, y la corrección de pruebas. Sin embargo, hay algunas actividades que los profesores realizan que no están consideradas como lo es la revisión de las prácticas. Estas prácticas pueden variar dependiendo de la carrera. Por ejemplo, algunas prácticas que comienzan al final del ciclo universitario durando aproximadamente dos meses, mientras que en el área de la salud, inician tratos con pacientes en su segundo año llevando así más carga laboral para los profesores, ya que cada profesor tiene su propia estructura.

En el caso de los estudiantes, estos deben contar con un registro de su participación en los hospitales y clínicas asignados. Este registro actualmente es un papel de sus actividades que es entregado al profesor encargado, sin contar que deben llevar un histórico de las actividades que ellos han realizado.

El objetivo de este proyecto llamado *SCOREMED* es poder construir una plataforma web que permita reducir la carga laboral para los profesores y mientras tanto los estudiantes puedan tener un registro en línea de sus actividades.

ABSTRACT

There are different kinds of activities that a teacher must do inside the university. The most common ones are teaching, planning and checking tests. However, there are some activities that are not considered such as the practicum review.

Practicums can vary depending on the program/career. For example, there are some practicums that start at the end of the program, lasting two months approximately, while in the health area, students start to interact with patients during their second year carrying more work to their teachers.

In the case of the students they have to register their activities. This register consists of a paper with the activities that has to be delivered to the teacher in charge. Also, the teacher must keep a record of the activities that the students did during the practicum.

The objective of this project called *SCOREMED* is to create a web platform in order to reduce the workload of the teacher.

1. INTRODUCCIÓN

En la Universidad Austral de Chile , la cual tiene varias carreras en diferentes áreas, se le pide a los estudiantes en sus últimos años realizar sus prácticas, en algunos casos esta práctica se realiza en el último año de dicha carrera, pero en el caso del área de la salud poniendo como ejemplo la carrera de Obstetricia y Puericultura, u otra carrera de la salud, los estudiantes empiezan aproximadamente en su segundo año de universidad a realizar prácticas, procedimientos y múltiples tratamientos con pacientes, ya sea de manera presencial, acompañado de un profesor hasta poder llegar a realizarlos de manera autónoma.

En este documento, se describirá el proyecto realizado para los estudiantes del área de la salud de la Universidad Austral de Chile, donde la carrera de Obstetricia y Puericultura serán los primeros usuarios de la plataforma de registro de practicantes. En términos generales este proyecto busca poder administrar de mejor forma los datos que obtienen los estudiantes en esta plataforma, y permite bajar la carga a los profesores dándole una manera más visual de las actividades de los estudiantes en dicha asignatura.

Esta plataforma web será en una versión gamificada, la cual nos permitirá generar un ambiente visualmente más agradable, con el fin de poder incentivar a cumplir las metas u objetivos que dará el profesor a dicha asignatura.

Por lo tanto, esta plataforma web permitirá guardar la información de las practicas a los estudiantes, y respecto a los profesores poder generar una estructura más ordenada para ellos y poder regirse en base a los documentos de la Organización Mundial de la Salud (OMS).

1.1 Objetivo general

Diseñar e implementar una plataforma web que permita guardar la información de las prácticas de una forma más interactiva, que permita optimizar el tiempo de las entidades presentes en este proyecto y generar un récord de actividades para los estudiantes teniendo así un registro de todas las actividades que realizaron hasta su último año.

1.2 Objetivos específicos

- Comprender el contexto de un sistema real en un hospital a través de una toma de requisitos, con al menos un actor de cada tipo (Profesor, tutor, estudiante).
- Realizar un prototipo funcional que cumpla con los requisitos de nuestro cliente
- Validar con datos reales y mediante un testeo comprobar el funcionamiento de dicho sistema.
- Mejorar el software a través de un feedback obtenido en un período de prueba.

2. MARCO TEÓRICO

En el siguiente capítulo se realizará una descripción de las problemáticas, soluciones y estudio de artículos que aportaron en el desarrollo de la plataforma web, que serán presentadas a continuación:

- Prácticas estudiantes
- Serious Game
- Gamificación
- Trabajo en equipo, actividades y/o objetivos
- Herramientas

2.1 Practica de estudiantes

Dentro de la Universidad Austral de Chile, nos encontramos con una gran variedad de carreras. Cada una de ellas, cuenta con un proceso distinto para la formación de sus estudiantes a profesionales, y durante esta formación, los futuros profesionales deben pasar por varias etapas. Dentro de estas etapas, se destaca una que varía dependiendo de la carrera, como en el caso de ingeniería, la cual necesita dos meses de práctica.

Sin embargo, las carreras correspondientes al área de salud destacan debido a la formación de sus estudiantes, ya que estos empiezan a interactuar con pacientes desde su segundo año, ya sea de una manera presencial o prestando ayuda, para que, una vez finalizada su carrera, ellos puedan realizar un procedimiento de manera autónoma

Al hablar de prácticas, se consideran varias actividades, procedimientos y tratamientos que son realizados, pero todo esto es información que es administrada por un profesor de la universidad que generalmente almacena esta información en alguna plantilla que utilizan para su comodidad.

Un ejemplo es la tabla que ingresamos a continuación, que representa las actividades que realiza un estudiante de obstetricia. El profesor adapta esta tabla a sus requerimientos con el fin de optimizar su trabajo y hacerlo más eficiente. La imagen que se apreciará (Ver tabla 1)¹ es un extracto ya que por lo general está conformada de varias hojas.

¹ Extracto de tabla adaptada

Tabla 1: Extracto de la hoja que utilizan

RÉCORD DE ACTIVIDADES DE ESTUDIANTES DE OBSTETRICIA Y PUERICULTURA ENFM 122 – 2019

NOMBRE:

Actividad		ARO	Puerperio y Puericultura	Partos y RNI	Emergencia Obstétrica	APS	Total
Administración y preparación de medicamentos	Vía EV						
	Vía Intramuscular						
	Administración subcutánea						
	Vacuna dTPA						
	Vacuna BCG						
	Vacuna Hepatitis B						
	Vitamina K						

Muchos profesores del área de la salud tienen su propia plantilla, cada una normalizada a su propio estilo y comodidad, pero cada asignatura, se rige por lo que dicta la OMS². En otras palabras, se puede decir que hay una estructura que deben seguir, pero la acomodan de manera independiente, considerando que el tiempo que empeñan los profesores de diferentes áreas varía entre cada profesor.

Una solución que se considera es la normalización de esta información y darle una estructura cómoda para reducir el tiempo y facilitar el trabajo a cada profesor. Otra facilidad que se tomará en consideración, es que toda la información que le llegaba a un profesor de forma física, una hoja de papel que posteriormente es traspasada a una plantilla, será reducida o eliminada con el fin de ir eliminando la carga académica extra de la entidad.

2.2 Serious Games

Los *Serious Games* o “juegos serios” son juegos diseñados con un propósito formativo más que para fines de entretenimiento. La expresión “serio” se refiere a aquellos

² Organización mundial de la salud

videojuegos que se utilizan en el sector educativo, científico, en la atención médica, planificación urbana, ingeniería y política, principalmente.

El término fue acuñado por Clark C. Abt (1970)³ y afirma que “Los *Serious Games* o juegos serios son juegos diseñados con un propósito formativo más que para fines de entretenimiento. La expresión serio se refiere a aquellos videojuegos que se utilizan en el sector educativo, científico, en la atención médica, planificación urbana, ingeniería y política, principalmente y así explorar las diferentes formas en las que los juegos se pueden incluir en el proceso de enseñanza-aprendizaje sin eliminar la diversión y el placer.

Los también conocidos como juegos formativos son especialmente eficaces para el aprendizaje de habilidades concretas, por ejemplo, para aprender un idioma o matemáticas. En la formación profesional también ha proliferado su uso ya sea para el desarrollo de competencias de comunicación y liderazgo, como para el desarrollo de proyectos de innovación.

A pesar de que hay un gran prejuicio sobre el uso de videojuegos, las estrategias lúdicas propias de los *Serious Games*, del aprendizaje basado en juegos o la gamificación, permiten a los docentes atraer a sus alumnos con los contenidos educativos, de esta manera los alumnos aprenden divirtiéndose.”

Dentro de la investigación, un artículo cuenta que utiliza una plataforma lúdica llamada *Mad Science* (Casper Harteveld y Steven Sutherland, 2017), que en pocas palabras realiza actividades lúdicas y rescata lo siguiente:

Los entornos tipo juego tienen un enorme potencial para avanzar en la ciencia social y del comportamiento. Se discutieron que dichos entornos pueden:

- Sumergir a las personas en situaciones auténticas
- Recuperar datos de comportamiento ricos de una manera discreta pero controlada
- Involucrar a una audiencia global amplia durante un período de tiempo sostenido.

Sin embargo, también se dijo que dichos entornos tienen una necesidad de personalización para realmente involucrar a los participantes y no aplicar un enfoque único de ajuste de coincidencias. Desarrollaron varias teorías de motivación que pueden ayudar a personalizar las tecnologías lúdicas, y se concluyó que la motivación es un proceso dinámico complejo que requiere considerar varias perspectivas del jugador.

Dentro de las primeras instancias, el concepto de *Serious Games* logro llamar demasiado la atención, pero al entrar más en este concepto y leer artículos, un *Serious Games* en palabras simple ayuda a enseñar, como por ejemplo, que nos dice que *Reduct* (Ian Arawjo, Cheng-Yao Wang, Andrew C. Myers, Erik Andersen, François Guimbretièrre, 2017), es un juego educativo que incorpora un nuevo enfoque de comprensión para enseñar a los novatos conceptos básicos de programación que incluye funciones, booleanos, igualdad, condicionales y funciones de mapeo sobre conjuntos y los datos recopilados dicen que los

³ Investigador estadounidense y autor del libro *Serious Games*

estudio realizados del laboratorio que demuestra que los principiantes pueden aprender conceptos de programación jugando el juego. También, presentan los resultados de una implementación en línea que muestra que el juego fue bien recibido. Sin embargo, hay pruebas no concluyentes del impacto del desvanecimiento de la concreción.

En el artículo *Developing a Serious Game for Nurse Education* se menciona que la futura educación en enfermería, tiene el desafío de desarrollar programas innovadores y efectivos que se alineen con los cambios actuales en el cuidado de la salud y para educar a las enfermeras con un alto nivel de habilidades de razonamiento clínico, conocimiento basado en evidencia y autonomía profesional. Los juegos serios son simulaciones basadas en computadora que combinan el desarrollo de conocimientos y habilidades con aspectos de juegos de video para permitir un aprendizaje activo, experiencial, situado y basado en problemas (Hege Mari Johnsen, Fossum, Mariann, Vivekananda-Schmidt, Pirashanthie, Fruhling, Ann, Slettebø, Åshild, 2018).

2.3 Gamificación

La Gamificación, es una técnica de aprendizaje que traslada la mecánica de los juegos al ámbito educativo-profesional con el fin de conseguir mejores resultados, ya sea para absorber mejor algunos conocimientos, mejorar alguna habilidad, o bien recompensar acciones concretas, entre otros muchos objetivos.

Este tipo de aprendizaje gana terreno en las metodologías de formación debido a su carácter lúdico, que facilita la interiorización de conocimientos de una forma más divertida, generando una experiencia positiva en el usuario.

El modelo de juego realmente funciona porque consigue motivar a los alumnos, desarrollando un mayor compromiso de las personas, e incentivando el ánimo de superación. Se utilizan una serie de técnicas mecánicas y dinámicas extrapoladas de los juegos.

La técnica mecánica, es la forma de recompensar al usuario en función de los objetivos alcanzados. Algunas de las técnicas mecánicas más utilizadas son las siguientes:

- Acumulación de puntos: se asigna un valor cuantitativo a determinadas acciones y se van acumulando a medida que se realizan.
- Escalando de niveles: se define una serie de niveles que el usuario debe ir superando para llegar al siguiente.
- Obtención de premios: a medida que se consiguen diferentes objetivos se van entregando premios a modo de colección
- Regalos: bienes que se dan al jugador o jugadores de forma gratuita al conseguir un objetivo.
- Clasificaciones: clasificar a los usuarios en función de puntos u objetivos logrados, destacando los mejores en una lista o ranking.
- Misiones o retos: conseguir resolver o superar un reto u objetivo planteado, ya sea solo o en equipo.

Las técnicas dinámicas hacen referencia a la motivación del propio usuario para jugar y seguir adelante en la consecución de sus objetivos. Algunas de las técnicas dinámicas más utilizadas son las siguientes:

- Recompensa: obtener un beneficio merecido.
- Estatus: establecerse en un nivel jerárquico social valorado.
- Logro: como superación o satisfacción personal.
- Competición: por el siempre afán de competir e intentar ser mejor que los demás.

La idea de la Gamificación no es crear un juego, sino valerse de los sistemas de puntuación-recompensa-objetivo que normalmente componen a los mismos.

Dentro de otros estudios se aplica elementos empleados en una clase, con el objetivo de hacerla más didáctica y el aprendizaje logrado dice que dentro del uso se vio una mejora con lo aprendido en clases. Incluso hasta mejoraron con la metodología de estudio que ellos tenían, pero abarcar todos los contenidos fue complicado y quedaron espacios y conocimientos por llenar, pero fue una grata experiencia (Patrícia Gomes Fernandes Matsubara y Caroline Lima Corrêa Da Silva ,2017).

A comparación de *OneUp* (Darina Dicheva, Keith Irwin, Christo Dichev, 2019), que es una plataforma de gamificación de cursos altamente configurable que permite a los instructores adaptar las funciones de gamificación para adaptarlas a sus preferencias, lo cual permite a una asignatura ser gamificada con el objetivo de que mediante la práctica en línea conduciría a un mayor autoaprendizaje y práctica de los estudiantes.

Los elementos del juego utilizados:

- Insignia
- Tabla de clasificación
- Moneda virtual
- Paneles de aprendizaje

2.4 Trabajo en equipo e individual, actividades y/o objetivos

Posteriormente luego de varios artículos se capturaron una variedad de conceptos claves, esperando que actúen de manera positiva, estos conceptos serán explicados a continuación.

2.4.1 Trabajo en equipo

Trabajar en equipo, es el esfuerzo integrado de un conjunto de personas para la realización de un proyecto.

Trabajar en equipo implica la coordinación de 2 a más personas orientadas para el alcance de objetivos comunes. Cada miembro debe aportar para la realización de una parte del trabajo.

La denominación trabajo, en equipo surgió después de la Primera Guerra Mundial. Hoy en día, es una forma de trabajo eficiente que entrega varios puntos de vista y que sirven para resolver más eficientemente conflictos que surgen en el ámbito económico, político y social.

Trabajar en equipo es una forma de realizar proyectos o alcanzar metas específicas. En este sentido, la definición de objetivos, como en cualquier proyecto, debe ser conocida de antemano por todos los miembros que integrarán el grupo de trabajo, también se recomienda definir claramente la división de las tareas. La asignación de tareas es generalmente hecha en conjunto. Durante la exposición de los objetivos del proyecto, cada uno de los integrantes podrá saber en las áreas en que podrá ser más eficiente o dar un mejor aporte.

No hay discurso actual en el campo de la educación o de la salud en el que no sea señalada la importancia del trabajo en equipo como una alternativa o demanda en el mundo contemporáneo para lograr metas en menor tiempo y con mayor alcance, a través del trabajo que realizan comunidades que comparten ideales y objetivos comunes. Es decir, se trata de una herramienta fundamental para el desarrollo de actividades.

Muchos juegos están aplicando este concepto del trabajo en equipo con diferentes roles a cumplir, juegos como *Defense of the Ancients* o *League of Legends* son un ejemplo claro de diferentes habilidades y compromisos que se deben cumplir para poder lograr una meta en común, la victoria.

2.4.2 Trabajo individual

Un trabajo individual, es un cuando una persona trabaja sola y no tiene ayuda de nadie, solo de sus conocimientos. Un trabajo individual le enseña a la persona a ser independiente en lo que hace, haciéndose de manera personas autocríticas buenas o malas, o eso es lo que se espera.

En sentido general, los trabajos que son hechos por una sola persona tienen identidades, cursa su propio objetivo, éxitos y fracasos. Se establecen características específicas, toma de decisiones propias y desempeña diferentes acciones de acuerdo a su conocimiento y características personales.

2.4.3 Objetivo

Como objetivo, se denomina el fin al que se desea llegar o la meta que se pretende lograr. Es lo que impulsa al individuo a tomar decisiones.

2.5 Impacto del Proyecto

Una vez que se comprendan los conceptos anteriores se espera que den forma a una plataforma con el fin obtener resultados positivos.

Si se tuviera que dividir el impacto que se quiere lograr serían los de corto, mediano y largo plazo.

2.5.1 Corto plazo

Se ha analizado los conceptos vistos previamente y al fusionar esos conceptos, se logró captar que dentro de un período corto de tiempo se podrían reducir los tiempos de revisión de los profesores, reducir la carga académica a los alumnos y dar una estructura más ordenada a las asignaturas.

2.5.2 Mediano plazo

Dentro del mediano plazo, el impacto que se espera es que las carreras de la salud empiecen a utilizar esta plataforma, para que todas tengan una estructura más clara y ordenada. Para que en un futuro no muy lejano, toda la universidad pueda tener acceso a esta plataforma e incluso otras universidades de la región de Los Ríos.

2.5.3 Largo plazo

Se estipuló que en un período largo de tiempo, la aplicación pueda utilizarse a nivel nacional con el fin de poder regular las prácticas de todos los estudiantes, para que todos ellos salgan con un promedio similar de actividades realizadas independientemente de la institución que estudiaron, región o ciudad, para que posteriormente, permita mejorar la experiencia de los futuros profesionales del área de la salud.

2.6 Herramientas

2.6.1 Frontend

El frontend es una de las separaciones que existen al momento de la programación, esta es conocida como la capa de presentación o más conocida como la interfaz de usuario. Este concepto nace aproximadamente el año 2008, ya que en ese tiempo empezaron a aparecer tecnologías que iniciarían este concepto, estas tecnologías serían:

- Html5
- Css3
- Json
- AngularJs
- Ember
- Backbone

- Rest
- Nodejs

En 2019, no es práctico imaginar el desarrollo web sin HTML, CSS y Javascript. Los lenguajes para destacar y los cuales se compararon fueron:

- Angular
- Vue.js
- React.js

Se considero una comparación mediante el factor de popularidad y tendencias, el resultado que obtuvo fue que React.js es el más popular en el 2019, seguido por Vue.js y finalmente Angular (Ver figura 1)⁴.

Most Loved, Dreaded, and Wanted Web Frameworks

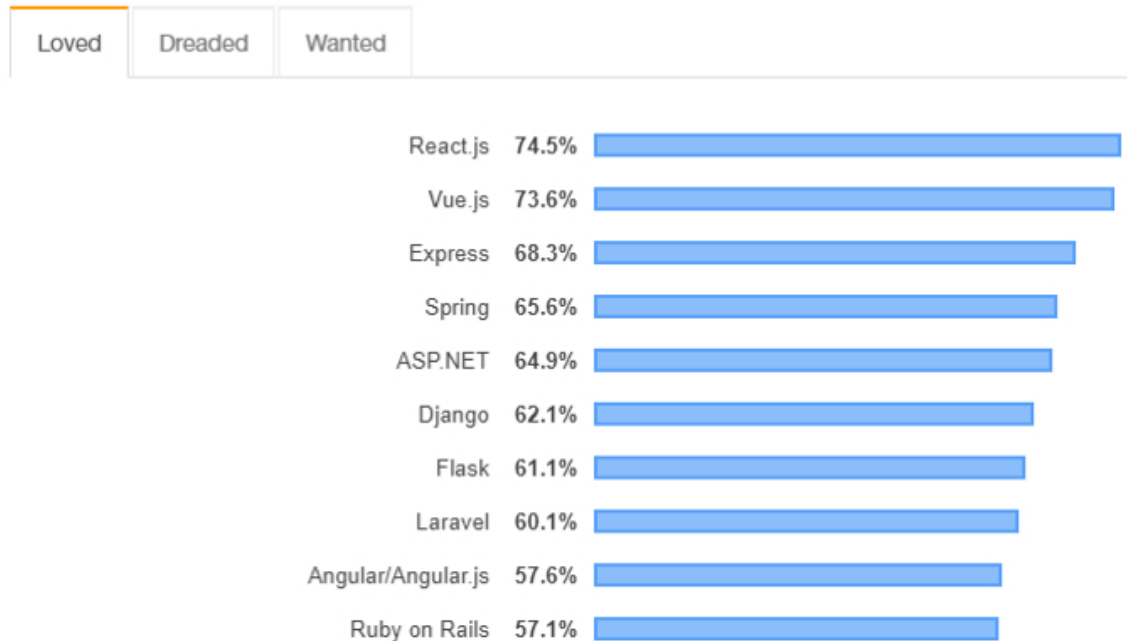


FIGURA 1: Resultado de encuesta de stackoverflow.

Pero a continuación se muestra una representación gráfica de las tendencias NPM⁵ para las descargas y estrellas de Angular vs React vs Vue (Ver figura 2)⁶.

⁴ Extraído de <https://medium.com/somoswigou/angular-vs-react-vs-vue-cu%C3%A1l-es-la-mejor-opci%C3%B3n-941a207951c7>

⁵ Node package manager

⁶ Extraído de <https://medium.com/somoswigou/angular-vs-react-vs-vue-cu%C3%A1l-es-la-mejor-opci%C3%B3n-941a207951c7>



FIGURA 2: Tendencias NPM⁷.

Muestra claramente React en la parte superior, seguido de angular y luego Vue. Pero cuando miramos las estrellas, la más baja se convierte en angular y la más alta en Vue, seguida de React.

Una vez que se realizó la búsqueda de popularidad por los desarrolladores, un factor que tomó en cuenta fue que instituciones trabajan con estas herramientas.

En el caso de Angular, este fue presentado por el reconocido líder del mercado Google. Además, la organización usa este Framework para sus *Adwords* (que consiste en brindar a las empresas la posibilidad de incluir sus anuncios en los resultados de las búsquedas que realizan las personas en Internet). Esto se suma a la autenticidad de Angular que lo proporciona fuerte apoyo de la comunidad desde su lanzamiento.

⁷ Extraído de <https://medium.com/somoswigou/angular-vs-react-vs-vue-cu%C3%A1l-es-la-mejor-opci%C3%B3n-941a207951c7>

Continuando con React fue desarrollado por Facebook para apoyar el mantenimiento y la codificación de sus campañas publicitarias. Mientras los ingenieros de Facebook trabajan constantemente en su mejora.

Vue, en comparación a las herramientas mencionadas anteriormente no tenía el respaldo de empresas de alto nivel, pero su popularidad en la comunidad de código abierto fue creciendo.

Se mencionó popularidad y prácticamente el apoyo que han recibido estas herramientas, pero ¿Qué tan difícil es aprenderlos? React es el preferido por los desarrolladores para usar y aprender, seguido de Vue y luego Angular.

La terminología se llama curva de aprendizaje por lo que Angular tiene una mayor curva de aprendizaje (Ver figura 3).

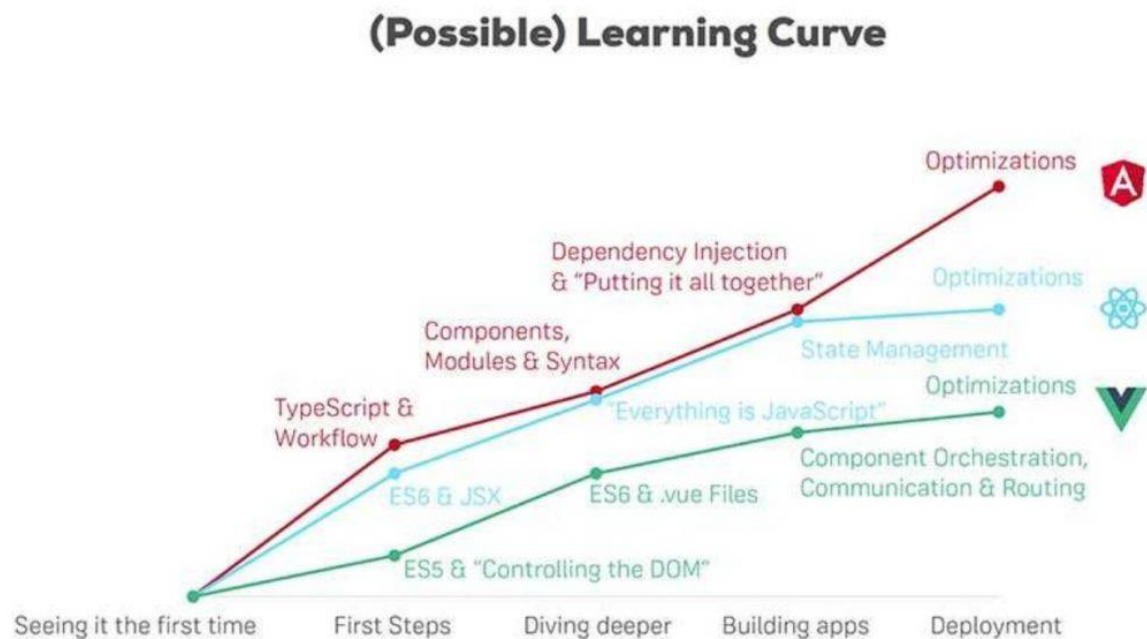


FIGURA 3 : Curva de aprendizaje⁸.

Dentro de la flexibilidad que ofrecen y como están estructurados, Angular ofrece todo lo que se necesita, pero no es muy flexible, React es más flexible y Vue no es muy flexible.

De esta comparación se llegó a la siguiente conclusión la cual es que:

- Vue y React ofrecen un mejor rendimiento y flexibilidad que Angular.
- Vue y React son más adecuados para aplicaciones livianas y angular es el mejor para aplicaciones de interfaz de usuario grandes.
- Angular es muy rígido y, a diferencia de Vue y React, ofrece todo desde enrutamiento, plantillas hasta pruebas de utilidades en su paquete.

⁸ Extraído de <https://www.openinnova.es/angular-vs-react-vs-vue-espanol-cual-elegir/>

- Vue es el framework más popular, querido y en crecimiento de Javascript.

2.6.2 Backend

Continuando con el backend es la otra separación de interés, pero esta viene siendo la capa de acceso a datos, el concepto fue apareciendo en el año 2008. Y como el frontend ha ido evolucionando, anteriormente existía un backend gigantesco que siempre se va alimentando, pero en la actualidad aparecen el concepto de microservicios con lo que se puede dividir esta lógica en pequeños servicios.

Symfony 4 es el framework desarrollado por *SensioLabs*, cofundada por Fabien Potencier, y desde su salida en 2011 ha sido de los frameworks más sólidos de PHP. Los componentes que forman parte de este lo hacen un framework robusto y ágil. Con respecto a su código, tendrá que al crear un nuevo proyecto con la versión 4, Symfony comienza con la estructura de un micro-framework. Elimina la mayoría de los componentes que en versiones anteriores se incluyen por defecto.

Symfony también desarrolló un plugin el cual se llama *Symfony flex*, que es una simplificación de los comandos *require*, *update* y *remove*. Básicamente permite instalar los módulos de Symfony mediante un alias, además de registrarlo automáticamente en la configuración del framework, evitándonos la necesidad de hacerlo manualmente.

Symfony fue comparado con otro framework llamado Laravel, el cual es un proyecto creado por Taylor Otwell, desarrollado principalmente por él con alguna ayuda de la comunidad. Laravel, ha acaparado mucha atención de los desarrolladores debido a la gran variedad de componentes y paquetes con los que cuenta. Otro punto positivo de Laravel, es la versatilidad con la que está codificado haciéndolo mucho más fácil de aprender y de utilizar. La versión de PHP⁹ 7.2 incluyó un nuevo algoritmo de encriptación llamada Argon2i. Laravel añadió el soporte para este algoritmo desde su componente de encriptación. Ahora, podemos encriptar las contraseñas dentro de nuestra app con un algoritmo más reciente y seguro.

Sin embargo, se llegó a una misma pregunta, la cual nos permite comparar estos framework en sus versiones nuevas y estables y así poder decidir ¿Cuál es el mejor?, se respondió a esta pregunta con que Laravel es mejor. Laravel tiene una curva de aprendizaje bastante cómoda, además de una comunidad muy grande dispuesta a ayudar. No es necesario ser un experto para programar con él. Además, te da la libertad de crear una estructura con base a tus necesidades. Su gran cantidad de componentes y paquetes hace que te enfoques en lo que es más importante para ti: la lógica de negocio de tu proyecto.

Pero Symfony es un excelente framework en proyectos más grandes. Con su nueva actualización la creación de microservicios será excepcional

⁹ Hypertext Preprocessor

Al momento de tomar la decisión sobre cuál de los dos es mejor, se basará el posible impacto que podrá tener la plataforma.

2.6.3 Base de datos

Para el almacenamiento de la información es necesario tener en cuenta que tipo de modelo utilizaremos en este proyecto, por lo que se analizarán un par de opciones de las cuales se tendrán base de datos relacionales y no relacionales o SQL¹⁰ o NoSQL¹¹ respectivamente.

2.6.3.1 Base de datos SQL o relacionales

Las bases de datos relacionales (BDR) es un tipo de base de datos que almacena datos relacionados entre sí. Este tipo de base de datos (BD) permite almacenar y representar los datos en tablas. Las tablas ya mencionadas estas compuestas por filas, y la estructura de estas filas es un registro con un identificador único, conocido como clave o llave. Tras ser postuladas sus bases en 1970 por Edgar Frank Codd, uno de los laboratorios IBM¹² en San José ,California, no tardó en consolidarse como un nuevo paradigma en los modelos de base de datos.

Al presentar proyectos que tienen varias entidades, se forman un conjunto de tablas que contienen datos en categorías predefinidas. Cada tabla tiene una o más categorías de datos en las columnas, mientras que cada fila tiene una instancia única de datos para las categorías definidas por las columnas.

2.6.3.2 Base de datos NoSQL o No relacionales

Principalmente, las bases de datos no relacionales se emplean para almacenar datos no estructurados, o de los que no se tienen ningún esquema o información predefinida

Las bases de datos NoSQL, es una amplia clase de sistemas de gestión de bases de datos que salen del modelo clásico de SGBDR (Sistema de Gestión de Bases de Datos Relacionales) en aspectos importantes, siendo el más destacado que no usan SQL como lenguaje de consultas. Los datos almacenados no requieren estructuras fijas como tablas, normalmente no soportan operaciones *JOIN*, ni garantizan completamente ACID (atomicidad, consistencia, aislamiento y durabilidad) y habitualmente escalan bien horizontalmente.

Se recopiló la información anteriormente mencionada donde se obtuvo lo siguiente (Ver tabla 2).

¹⁰ Structured Query Language

¹¹ No structured Query Language

¹² International Business Machines Corporation

Tabla 2 : Base de datos relacional vs base de datos no relacional¹³.

Base de datos relacionales	Base de datos no relacionales
Se organiza en forma estructurada, en tablas.	No conserva la forma estructurada y utiliza generalmente documentos.
Lenguaje principal de consulta SQL.	Su lenguaje principal para consultas no es el SQL.
Se recomienda cuando el volumen de datos es constante o crece poco a poco.	La escalabilidad es mayor en una base de datos no relacional, y también están preparadas para soportar mayor volumen de datos.

Una vez realizada la comparación entre los modelos mencionados, el que se empleará será el modelo relacional ya que por su forma estructurada de almacenar, la información en tablas nos permitirá organizar el proyecto, aparte del conocimiento que se tiene de SQL facilitara la realización de consultas a nuestra base de datos.

A continuación (Ver figura 4 y Anexo A), se aprecia la estructura que deberá tener la base de datos para *ScoreMed*, la cual consta de 7 tablas en total. En la tabla *user*, se tendrá todas las características y credenciales del usuario. Mediante la columna roles podremos identificar a nuestro usuario para asignar las acciones correspondientes. Mientras tanto, la tabla *subject* y *mission* se quedará con la información de la asignatura y las actividades a realizar respectivamente.

Como tablas secundarias, pero no menos importantes son las que surgen mediante la relación varios es a varios, y estas tablas permitirán tener una información de unión entre las tablas.

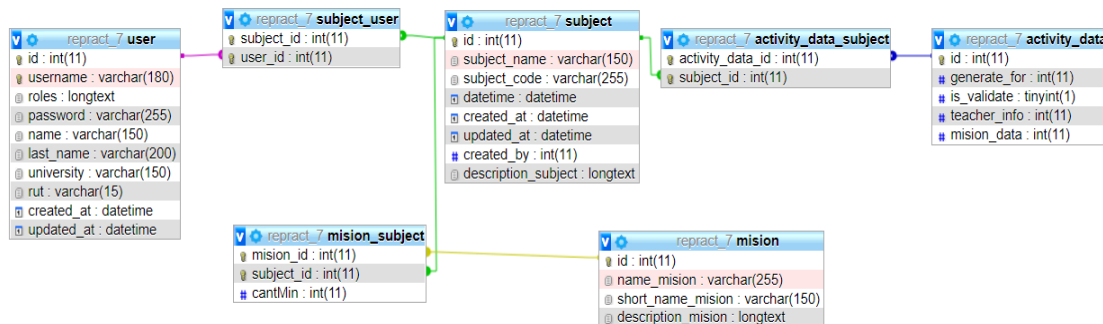


FIGURA 4: Base de datos del proyecto¹⁴

¹³ Elaboración propia

¹⁴ Elaboración propia

3. ARQUITECTURA Y SOLUCION

En este capítulo, se mostrará lo referente al proceso de desarrollo para obtener nuestra solución propuesta, documentación, comparación de tecnologías y las tecnologías utilizadas.

3.1 Descripción de la problemática

Los profesores de la Universidad Austral de Chile, específicamente en el área de la salud, se han visto envueltos en una acumulación de material físico de las prácticas de sus estudiantes. Estos nos llevan a que hay demasiada información que se recopila y que tiene que ser revisada en un tiempo relativamente corto. Con toda esta información, se necesitó realizar una evaluación y surgieron las siguientes preguntas ¿Se podrá normalizar esta información ?, ¿Los estudiantes realizan todas las actividades que pide la OMS¹⁵ antes de titularse ?, ¿Los estudiantes cumplen los procedimientos mínimos que pide una asignatura?

Actualmente, se ha generado una estructura en la que se almacena la información, pero aun así es tediosa y consume mucho tiempo para el profesor, ya que tiene que realizar esta acción con el fin de poder finalizar la asignatura. Se comprendió como era el crecimiento de las habilidades que deberían adquirir los alumnos en el paso de dicha carrera, con el objetivo de poder realizar un procedimiento de manera autónoma, y este proceso ya tiene una estructura sólida.

La solución propuesta, consiste en realizar una plataforma web que permita almacenar la información de las prácticas de los estudiantes, con el propósito de reducir el tiempo de las entidades presentes, que los estudiantes realicen las actividades que pide la OMS antes de titularse, que cumplan las actividades solicitadas por una asignatura en su totalidad, poder almacenar la información normalizada y así poner a la vista de una forma más agradable toda esta información.

3.2 Solución propuesta

Se propone desarrollar una plataforma web en una versión gamificada, que sea capaz de almacenar la información de las prácticas de los estudiantes y que permita generar un récord de actividades para esta entidad y poder ver de una forma visual y más dinámica su información, con el fin de generar el registro de todo lo realizado en sus estudios y en sus asignaturas. Continuando con otra entidad, el profesor, se verá beneficiado con un mejor manejo de la información, con la cual podrá dar una estructura a su asignatura y realizar un mejor seguimiento a los estudiantes, y poder ver si realizaron el mínimo de actividades, en cuanto al tiempo que esto implicaba, se reducirá bastante ya que el profesor deberá crear su curso y dar los objetivos que se deben cumplir.

¹⁵ Organización Mundial de la Salud

La información que llegará al sistema, serán las actividades realizadas en la práctica por los estudiantes, y esta información vendrá con la validación de un tutor asignado que rectifica que la actividad fue realizada por el estudiante.

3.3 Metodología

La metodología de trabajo para el proyecto, se basa en las etapas que esta lleva como lo es la investigación, desarrollo, pruebas o testeo, validaciones, análisis de datos y conclusiones. Para este proyecto, se utilizará *Focus Group* para la toma de información y para el desarrollo un Kanban modificado.

El método *Focus Group*, consta de reunión con los participantes donde se exponen opiniones sobre el producto. En esta etapa, se realizará una presentación enseñando un posible producto presentado por el estudiante, y por la parte del cliente, que serán profesores de obstetricia los cuales se espera que darán un feedback esperado para así capturar mejor los requerimientos y tener una mejor captura de lo que necesitan.

En el caso del método Kanban, se puede definir como un sistema que coordina la entrega de cada objetivo o tarea, con el fin de crear una especie de armonía en la creación del producto, generalmente se utilizan tarjetas y hay plataformas que facilitan este uso como lo es Trello (Ver figura 5). En este caso, se vio modificado, ya que las tareas principales fueron extraídas del método utilizado anteriormente y cada objetivo o tarjeta tendrá su check-list para ir avanzando gradualmente.

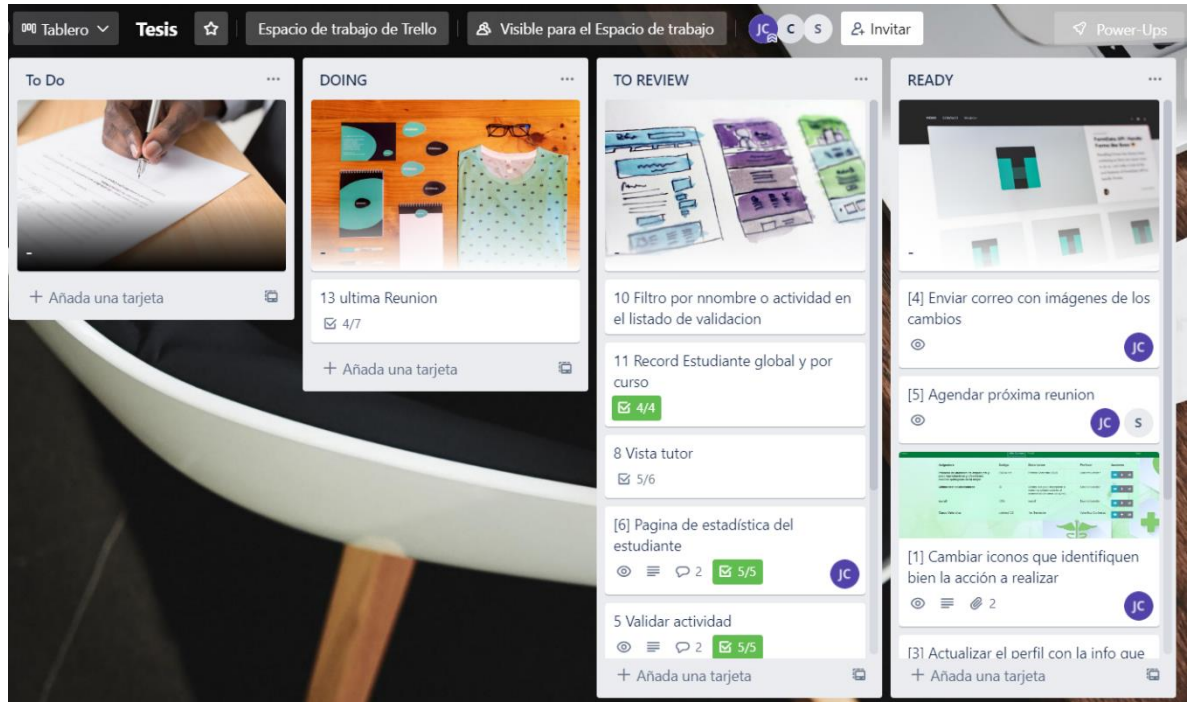


FIGURA 5: Imagen de Trello¹⁶

3.4 Diagramas de secuencia

En esta sección del capítulo, se dará el detalle de los diagramas de secuencias que deberían seguir las entidades presentes en nuestra plataforma.

A continuación, se explicará cómo funciona nuestro sistema (Ver figura 6), antes de continuar con las actividades que le dan forma a esta estructura.

El sistema captura la data ingresada por nuestro usuario llamando así a una función, en el backend comprobamos el token que da el permiso necesario para ejecutar esta función, con la que se obtendrá o verificará la información en la base de datos que enviará una respuesta de los datos existentes, se deberá comprobar que dicha información o dato existe en la base de datos, el backend enviará una respuesta al frontend que permite que el usuario pueda ver la información solicitada por su acción específica (de ahora en adelante cada vez que se mencione el sistema se debe tener en mente este diagrama).

¹⁶ Elaboración propia

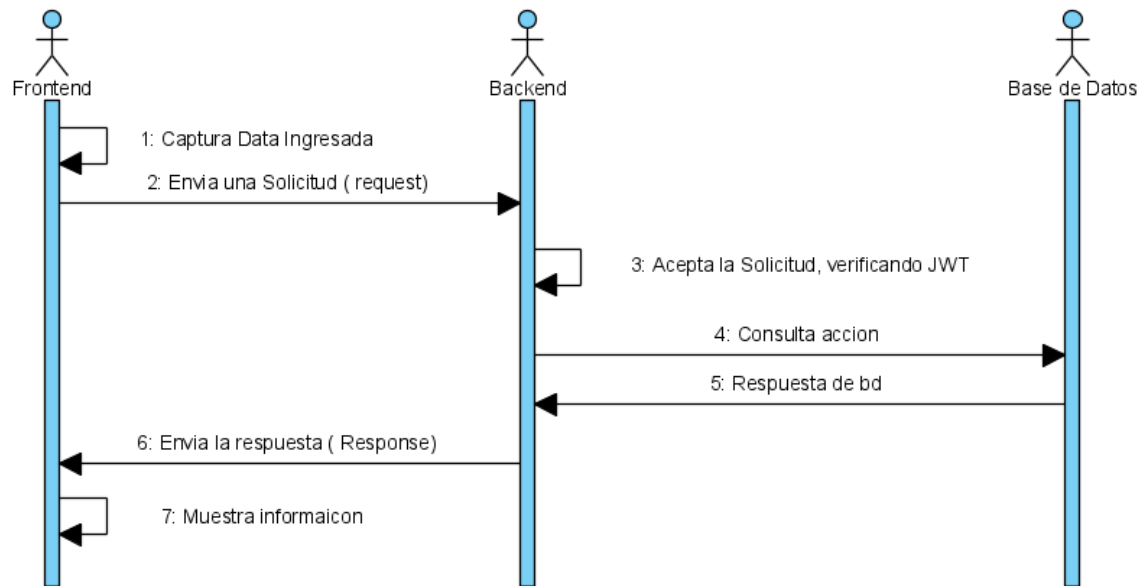


FIGURA 6: Diagrama Sistema¹⁷

Una acción que permite utilizar la plataforma, es la de registrar a un usuario (Ver figura 7), esta acción dice que si un usuario no está registrado, tendrá que enviar sus datos correspondientes, al ser ingresados el sistema consulta si esos datos ya existen, y si no existen agrega esta información a la base de datos, posteriormente se tomaran los datos *username* y *password* para que acceda al inicio de esta plataforma web.

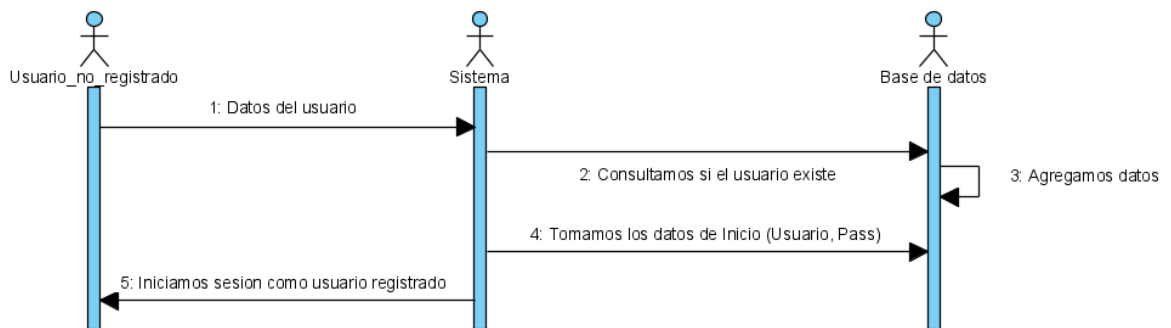


FIGURA 7: Diagrama Registro¹⁸

A modo de ejemplo, se mostrará parte del código hecho para realizar esta acción de registro la cual corresponde al backend y que fue creado siguiendo un ejemplo (Restfull API con Symfony + JWT,2020).

¹⁷ Elaboración propia

¹⁸ Elaboración propia


```

1. @Rest\Post ("/register", name="user_register")

2. public function registerAction (Request $request,
    UserPasswordEncoderInterface $encoder) {
3. $serializer = $this->get('jms_serializer');
4. $em = $this->getDoctrine()->getManager();

5. $user = [];
6. $message = "";

7. try {
    a. $code = 200;
    b. $error = false;

    c. $name = $request->request->get('name');
    d. $username = $request->request->get('username');
    e. $roles = $request->request->get('roles');
    f. $password = $request->request->get('password');
    g. $lastName = $request->request->get('lastName');
    h. $rut = $request->request->get('rut');
    i. $university = $request->request->get('university');
    j. $password = $request->request->get('password');
    k. $user = new User();

    l. $user->setUsername($username);
    m. $user->setName($name);
    n. $user->setRoles($roles);
    o. $user->setRut($rut);
    p. $user->setLastName($lastName);
    q. $user->setUniversity($university);
    r. $user->setPassword($encoder->encodePassword($user,
        $password));

    s. $em->persist($user);
    t. $em->flush();

8. } catch (Exception $ex) {
    a. $code = 500;
    b. $error = true;
    c. $message = "An error has occurred trying to register the user
        - Error: {$ex->getMessage()}";
9. }

10. $response = [
    a. 'code' => $code,
    b. 'error' => $error,
    c. 'data' => $code == 200 ? $user : $message,
11. ];

12. return new Response($serializer->serialize($response,
    "json"));
13. }

```

Y utilizando los servicios llamamos a nuestra función de registro

```

1. public registerIssues (user: User): Observable<any> {
2. return this.http.post ('http://localhost/api/register', user);}

```

para aplicarlo en el frontend

```

1. registerForm: FormGroup;

2. constructor (private fb: FormBuilder,private issueService:
   IssueService,private _router:Router,private _activateRouter:
   ActivatedRoute) {

3. this.registerForm = this.fb.group(
   a. {
   b. username: ['', Validators.required],
   c. name: ['', Validators.required],
   d. password: ['', Validators.required],
   e. confirmarPassword: ['', Validators.required],
   f. lastName: ['', Validators.required],
   g. rut: ['', Validators.required],
   h. university: ['', Validators.required],
   i. roles: ['', [Validators.required]]
4. },
5. {
   a. validators: validarQueSeanIguales,
6. }
7. );

8. }

9. onSubmit() {
   a. let usernameLog = this.registerForm.get('username').value;
   b. let passLog = this.registerForm.get('password').value;
                                   i. return
                                   this.issueService
                                   .registerIssues(t
                                   his.registerForm.
                                   value).subscribe(
                                   response=>{if
                                   (response.code ==
                                   200 )

   c. {
       i. if (confirm('Desea iniciar Sesion ?')) {
       ii. this.issueService.loginIssues(usernameLog,passLog)
       iii. .subscribe( response2 => {
       iv. this.data = response2.data;
       v. console.log(this.data);
       vi. StatusLogin.clave = 'Bearer ' + response2.token;

       vii. localStorage.setItem('id', this.data.id)
       viii. localStorage.setItem('rol', this.data.roles)
       ix. this.issueService.ckeckLoggedIn(true);
       x. if (this.data.roles == 'ROLE_TEACHER' ) {
       xi. this._router.navigateByUrl('/homeT');
       xii. }
       xiii. else if(this.data.roles == 'ROLE_STUDENT') {

```

```

xiv. this._router.navigateByUrl('/homeST');
xv. }
xvi. else if(this.data.roles == 'ROLE_TUTOR') {
xvii. this._router.navigateByUrl('/homeTutor');
xviii. }
xix. });
d. }
e. }
10. });

11. }
12. checarSiSonIguales(): boolean {

a. return this.registerForm.hasError('noSonIguales') &&
b. this.registerForm.get('password').dirty &&
c. this.registerForm.get('confirmarPassword').dirty;
13. }

14. // Choose Role using select dropdown
15. changeRoles(e) {
16. this.roles;
17. }

18. // Getter method to access formcontrols
19. get roles() {
20. const role = this.registerForm.get('roles');
21. return role.value;
22. }

23. changeUniversity(e){
24. this.universidad
25. }

26. get universidad() {
27. const universidad = this.registerForm.get('university');
a. return universidad.value;
28. }

29. }

30. export const validarQueSeanIguales: ValidatorFn = (
a. control: FormGroup
31. ): ValidationErrors | null => {
a. const password = control.get('password').value;
b. const confirmarPassword = control.get('confirmarPassword').value;

c. return password === confirmarPassword
d. ? null
e. : { noSonIguales: true };
32. };

```

Si el usuario ya se encuentra registrado, podrá realizar la siguiente secuencia, en la que puede acceder con su *username* y *password*. Luego de hacer click, nuestro sistema comprueba si la información agregada existe en nuestra base de datos validando la

autenticación, para posteriormente, generar nuestro token que ayudará al usuario interactuar con nuestra plataforma y finalizamos esta acción guardando el rol que ha ingresado (Ver figura 8).

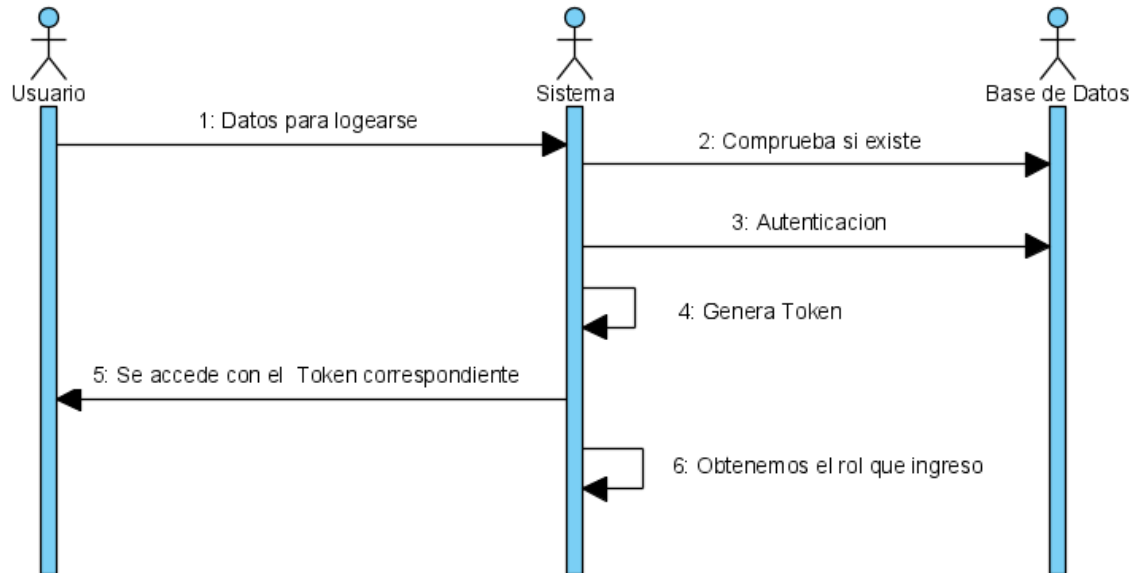


FIGURA 8: Diagrama Login¹⁹

Ya explicado como registrarse y como tener acceso a la plataforma es el momento en que se consideraron las principales que podrán utilizar por roles.

3.4.1 Rol Profesor.

Este rol, podrá realizar las acciones que se describen a continuación, estas acciones son claves para generar la estructura de una asignatura con los objetivos que deberán cumplir los estudiantes.

Al retomar el diagrama anterior (Figura 6), el usuario, una vez iniciada su sesión, guardara su rol con él fue ingresado, para así hacer uso de ese rol para comprobar que funciones se habilitan, como la siguiente que se llama Crear una asignatura (Ver figura 9).

Una vez enviado el rol que se guardara, el sistema comprueba que sea el rol correcto para poder habilitar la función, una vez habilitado el usuario accede a ella e ingresa los datos de su asignatura que tiene campos como el nombre y el código, se continúa agregando estos datos a la base de datos y el sistema pregunta si desea agregar a los profesores clínicos que serán los otros profesores que participarán en la asignatura. Posteriormente, el sistema cambia la vista desplegando una lista de las asignaturas hechas por ese profesor.

¹⁹ Elaboración propia

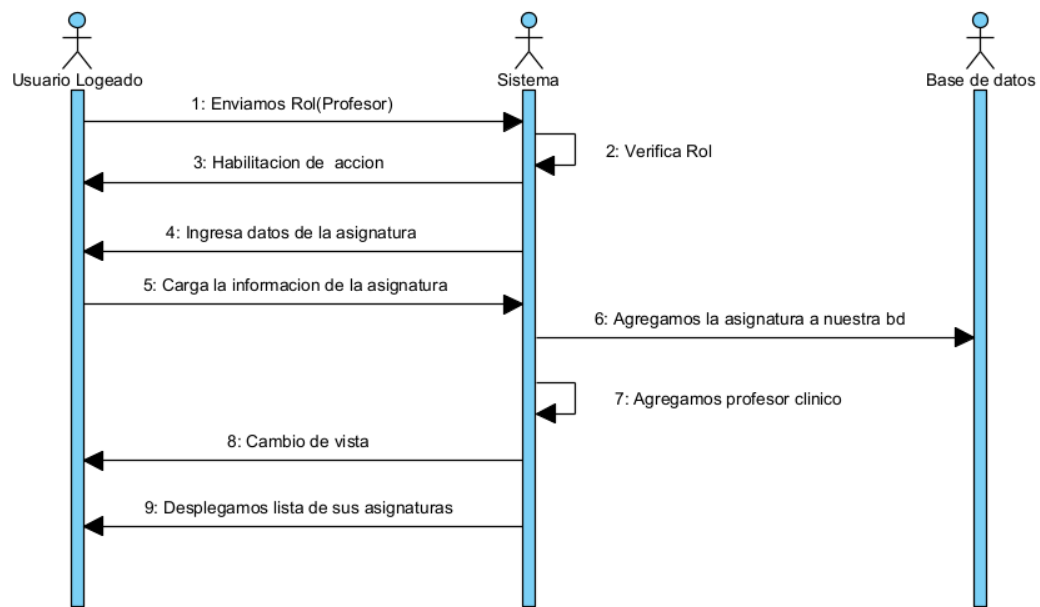


FIGURA 9: Crear una asignatura²⁰

En el caso de edición de una asignatura (Ver figura 10), una vez obtenido el rol, ya que solo el profesor puede editar la asignatura, comprobará y habilitará dicha acción, el sistema carga en un formulario los datos de la asignatura que necesita ser editada, con el modo de edición habilitado, se ingresa la nueva información para que posteriormente sea almacenada en la base de datos. Ya almacenada, se realiza un cambio de vista donde mediante una función del sistema y gracias al identificador del usuario visualizamos nuevamente la lista de la asignatura de dicho profesor.

²⁰ Elaboración propia

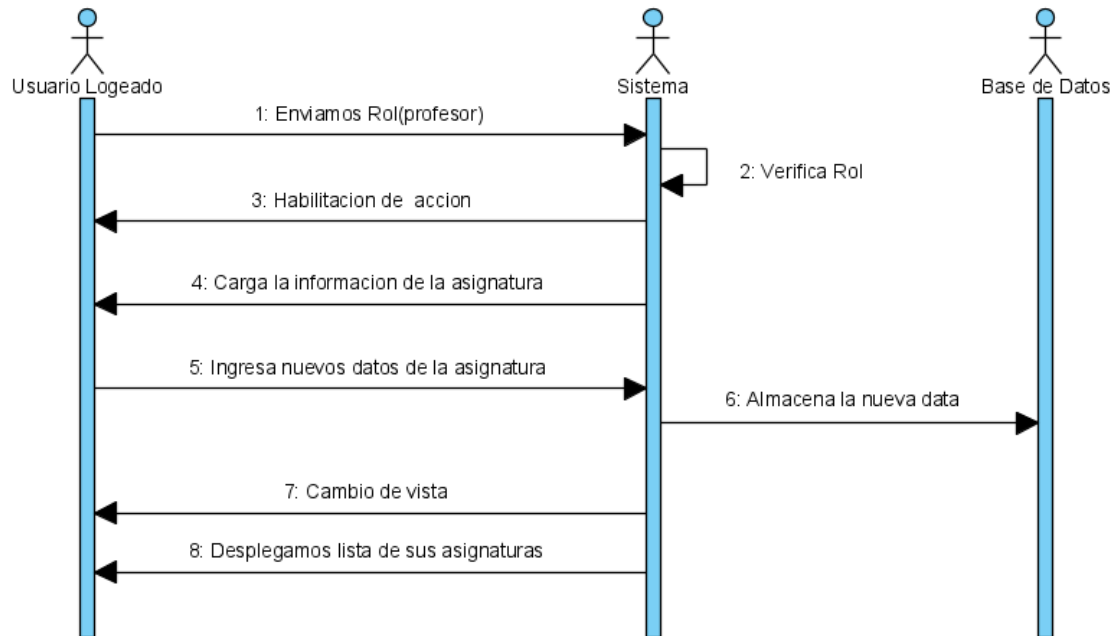


FIGURA 10 : Diagrama de edición de una asignatura²¹

El profesor para poder crear una misión (Ver figura 11) una vez logeado se procede verifica el rol con el que habilitará la acción que permite crear la misión que va a utilizar más adelante, los datos son ingresados por el usuario y esta nueva información es agregada a la base de datos en su tabla correspondiente, para pasar por un cambio de vista y desplegar la lista de misiones.

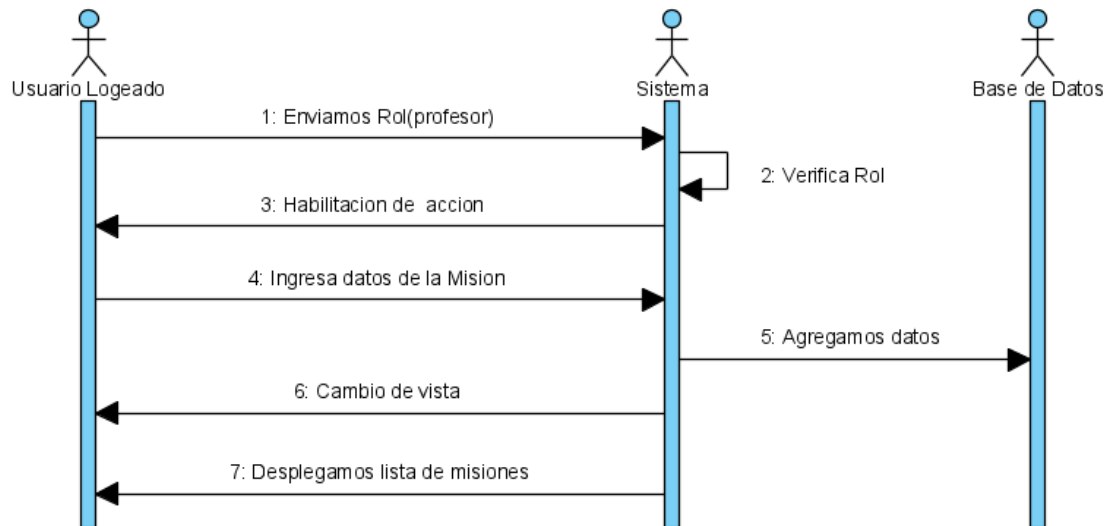


FIGURA 11: Diagrama de creación de una misión²²

²¹ Elaboración propia

²² Elaboración propia

Un procedimiento que le da una estructura a esta plataforma es la vinculación de una misión a una asignatura (Ver figura 12). Esta acción que es realizada solo por el profesor, está conformada de la siguiente forma; el rol del profesor es ingresado al sistema que verifica a esta entidad, habilitada la acción el usuario ingresa a una opción que le permite ver la lista de misiones. En el transcurso de ese procedimiento obtendrá el identificador de la asignatura para poder realizar la vinculación. Ya desplegada la lista de misiones, el usuario puede elegir que misiones podrá tener esta asignatura y al hacer click, esta vinculación es agregada a la base de datos a su tabla correspondiente, para posteriormente, mostrar las misiones seleccionadas en la asignatura asignada.

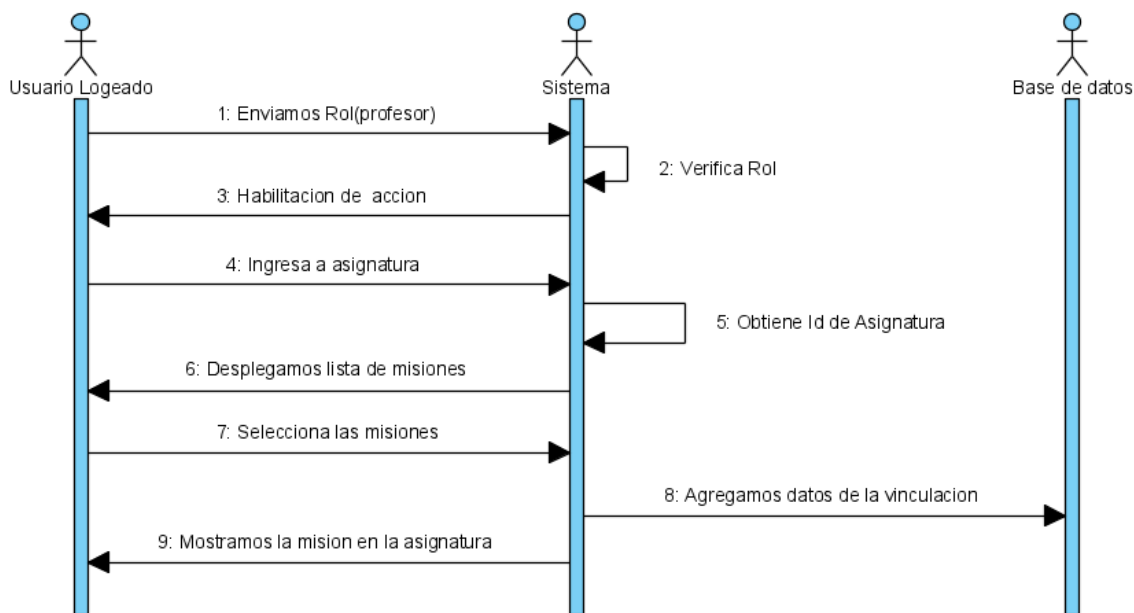


FIGURA 12: Diagrama de vinculación de una misión a una asignatura²³

Los estudiantes enviarán sus actividades a un profesor relacionado con esa actividad para después, ser validada por el profesor. Esto lo explica el siguiente diagrama (Ver figura 13), en la vista del profesor y el tutor, se apreciará un icono que vendrá con un número, lo cual significa la cantidad de actividades que tiene que validar el profesor correspondiente. Esto se realiza mediante una query que nos trae ese resultado y muestra la información en la plataforma.

Al momento de acceder a esta sección, el profesor deberá tener todas las actividades en las que él ha sido mencionado, y podrá apreciar la actividad y que alumno lo ha mencionado. Además, el profesor tendrá la acción de aceptar o rechazar esa acción, y luego de haber realizado la acción, se actualizarán la base de datos, y el sistema va a ir refrescado la lista una vez realizada esta validación.

²³ Elaboración propia

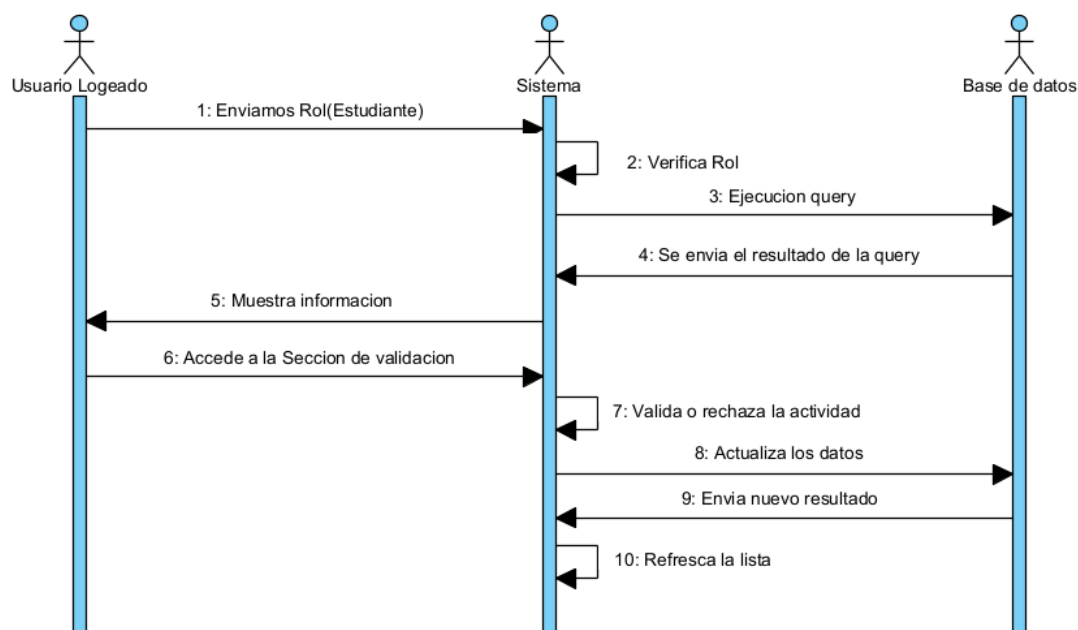


FIGURA 13: Diagrama de validación/rechazo de las actividades²⁴

3.4.2 Rol Tutor

El siguiente rol, será como el de un profesor, pero con menos permisos, por lo que no podrá realizar algunas acciones dentro de esta plataforma como lo es un CRUD²⁵ de asignaturas ni de misiones.

Una vez que el tutor ingresa a la plataforma el sistema ejecuta una query (Ver figura 14) que nos facilita la información de las asignaturas que tendrá el tutor, que será desplegada en una tabla.

²⁴ Elaboración propia

²⁵ Create, Read, Update, Delete

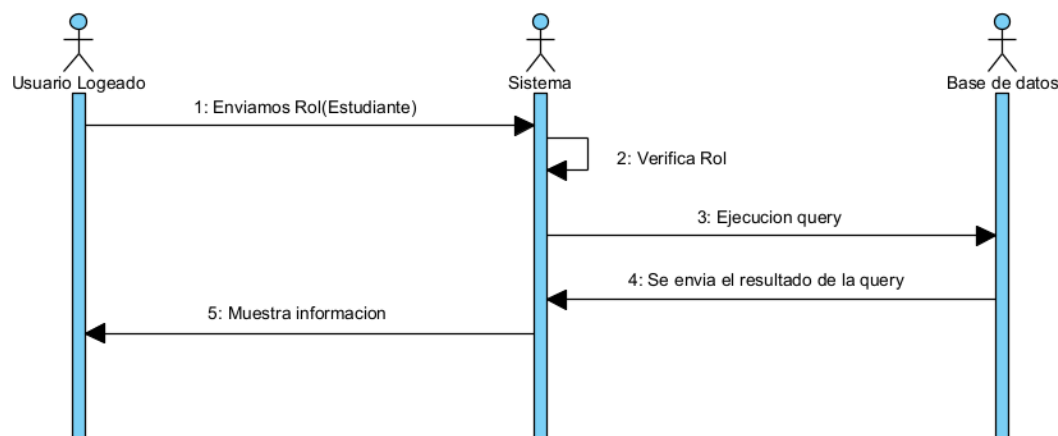


FIGURA 14: Diagrama de la vista inicial del tutor²⁶

El tutor también deberá realizar la acción de validación que se explicó anteriormente (Ver figura 13), ya que el también estará presente en las actividades que deberá realizar el estudiante.

3.4.3 Rol Estudiante

El siguiente rol se encargará de la elección de su curso, que vendrá con actividades que el usuario deberá cumplir, todas las acciones que se realizarán a continuación serán luego de realizar el login (Ver figura 8).

El estudiante ya logeado, deberá poder vincularse o inscribirse a un curso (Ver figura 15) que ha sido creado por el profesor (Ver figura 9). Esta acción, debe iniciar con el usuario logeado. Se comprobará el rol, que será estudiante, se despliega la lista de todos los cursos existentes que han sido creados. Al ver los cursos existentes el usuario podrá inscribirse a la asignatura deseada, una vez que realice esta acción el sistema recibirá la información de que estudiante se está vinculando y a que asignatura, el sistema enviará esta información a la tabla correspondiente a la base de datos, al actualizar dicha tabla, los datos del sistema se actualizan, permitiendo que el usuario aprecie la acción que realizó que fue, inscribirse o vincularse a una asignatura.

²⁶ Elaboración propia

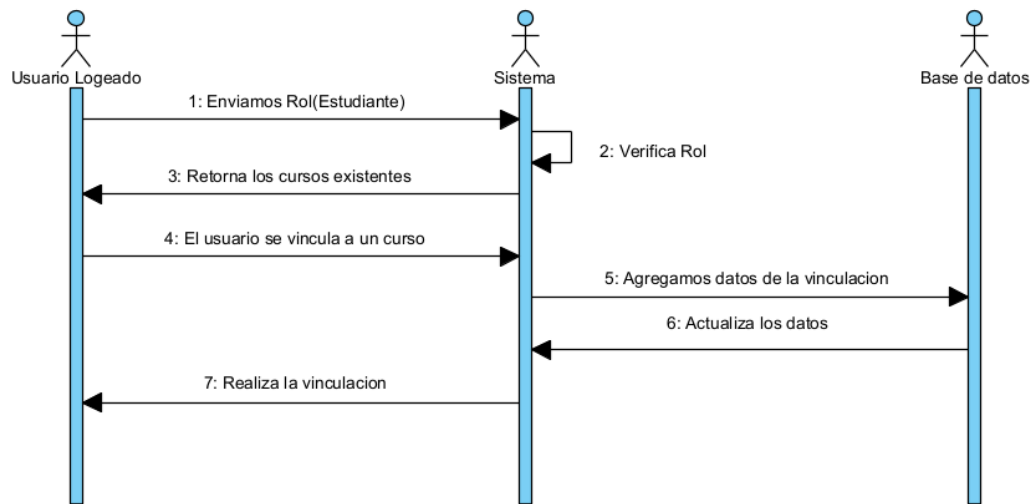


FIGURA 15 : Diagrama de vinculación de un estudiante a un curso²⁷

Continuando con el siguiente diagrama, este tiene relación con el anterior (Ver figura 15). dentro de las pestañas que ofrece la plataforma, hay una que se llama Mis cursos, que funciona de la siguiente forma (Ver figura 16); da inicio con la verificación del rol. Cuando se realice esa verificación, el sistema ejecutará un query que utiliza los datos del estudiante para así extraer la información de los cursos a los que el estudiante se ha inscrito, y al enviar estos datos el sistema muestra una lista de los cursos con nuevas acciones a realizar.

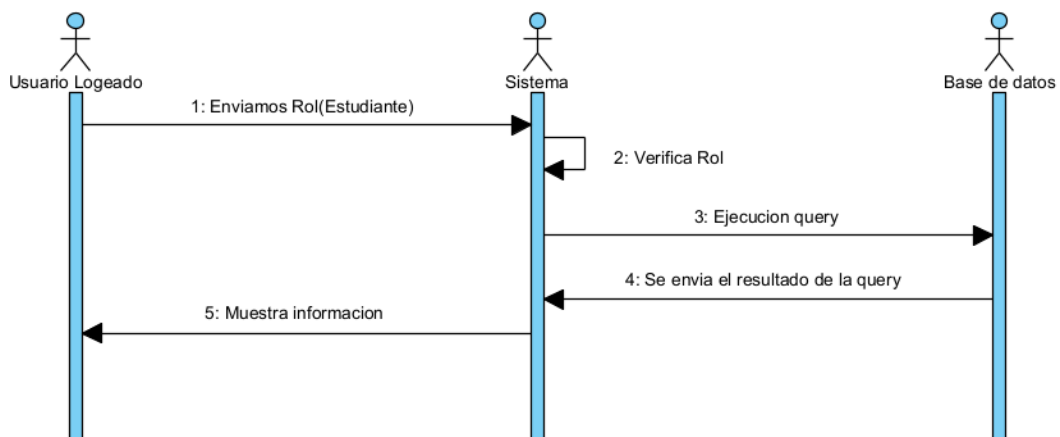


FIGURA 16: Diagrama de los cursos de un estudiante²⁸

Siguiendo con el próximo diagrama, este hace referencia a una de las acciones que alimentará a las asignaturas con la información. El diagrama que se presenta continuación (Ver figura 17), representa como se agrega una actividad realizada por un estudiante. Cuando ya está verificado el rol, y el estudiante hace click en agregar actividad, que está

²⁷ Elaboración propia

²⁸ Elaboración propia

representada por el icono más (+), su vista cambiará, y la nueva visualización traerá información de las actividades que deben de realizar en esa asignatura. Luego de completar con la información solicitada, que es la actividad realizada y con que profesor o tutor, ésta información será enviada a nuestra base de datos y será almacenada en una tabla que espera la verificación de los profesores o tutores.

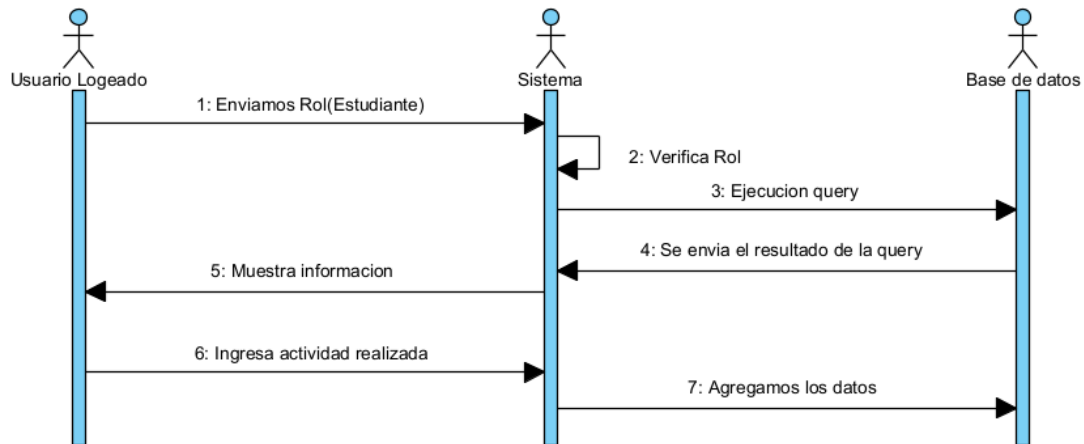


FIGURA 17: Diagrama para agregar una actividad²⁹

Cuando el profesor o tutor valida una actividad que es enviada por el estudiante (Ver figura 15), en la pestaña de mis cursos podrá acceder a una opción llamada mi progreso. En esta sección validamos el rol, y el sistema ejecuta un query que envía como resultado la actividad que debe de realizar el estudiante, con un mínimo que deben igualar o superar y las actividades que han sido validadas por los profesores que han sido mencionada en esta, esa información que se mostrará en el sistema se verá reflejado en un gráfico de radar, con las actividades mínimas y las que van realizadas en ese momento.

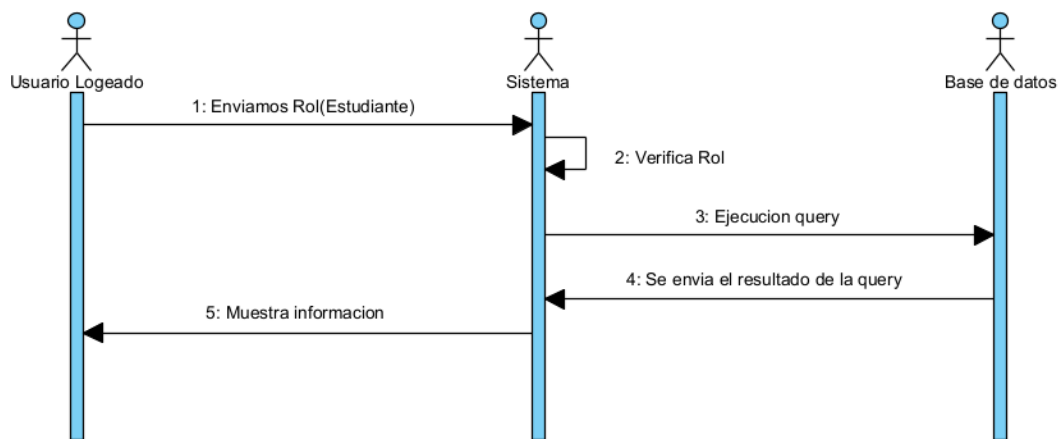


FIGURA 18: Diagrama para ver el progreso del estudiante³⁰.

²⁹ Elaboración propia

³⁰ Elaboración propia

Finalmente, en la barra de navegación el estudiante tendrá una pestaña llamada mi récord, al hacer click dentro de esta pestaña, el estudiante tendrá el registro de todas las actividades que han sido aprobadas por sus profesores y tutores, todo esto se realiza mediante una query, para hacer referencia la secuencia es igual a la anterior (Ver figura 18).

4. SERVIDOR

Un servidor, es un computador que forma parte de una red informática y facilita determinados servicios al resto de los computadores. Dicho computador, debe contar con características específicas y ser capaz de atender las peticiones de los distintos clientes. Escrito lo siguiente, se especificarán las características del servidor que se utilizará en este proyecto (Ver tabla 3).

4.1 Características

Tabla 3: Características del servidor³¹

Requerimiento	Especificación
Sistema Operativo	Ubuntu 18.04
Procesador	2 Cores
Memoria	2 GB
Disco	5 GB Disponibles
Puertos	Puerto de acceso remoto: 22 Puerto Backend: 8000 Puerto Frontend: 4200 Puerto Http: 80

Una vez que el servidor este corriendo, se tendrá que instalar algunas dependencias para poder realizar la compilación del proyecto, de las cuales estas dependencias serán descritas en forma breve.

4.2 Dependencias

4.2.1 Angular/cli

Con la versión 7.1.4 Angular CLI, es intérprete de línea de comandos de Angular que permitirá iniciar el proyecto y la creación del esqueleto de todo tipo de artefactos necesarios para el desarrollo de aplicaciones.

4.2.1 Typescript

Con la versión 3.1.1 Typescript es un lenguaje de programación fuertemente tipado que se basa en JavaScript, lo que le brinda mejores herramientas a cualquier escala.

4.2.3 Angular/core

Utilizando la versión 7.1.0 esta dependencia implementa la funcionalidad principal, los servicios de bajo nivel y las utilidades de Angular. Define la infraestructura de clases para componentes, jerarquías de vista, detección de cambios, renderizado y manejo de eventos y a la vez define la infraestructura para la inyección de dependencias (Angular/core,2020).

³¹ Elaboración propia

4.2.4 Angular/router

La versión para utilizar es la 7.1.0 esta dependencia ayudará a implementar el servicio Angular Router, que permite la navegación de una vista a la siguiente a medida que los usuarios realizan tareas de la aplicación (Angular/router,2020)

4.2.5 Angular/forms

Nuevamente con la versión 7.1.0 nos ayudará a crear formularios y modelos de datos, y proporcionar validación a sus formularios. Los formularios de angular nos permitirán capturar el valor actual y el estado de validación de un formulario, realizar un seguimiento de los cambios en el modelo de datos del formulario, Valide la exactitud de la entrada del usuario, crear validadores personalizados y elementos de entrada (Angular/forms,2020).

4.2.6 Chartjs

Utilizando la versión 2.9.3 esta librería de gráficos, como su propio nombre indica, generan distintos tipos de gráficos para representar datos en base a una información dada, ya sea en forma de fichero externo, o en forma de query (Chartjs,2020).

4.2.7 Rxjs

RxJS proporciona una implementación del tipo Observable (Es aquello que queremos observar, que será implementado mediante una colección de eventos o valores futuros), La biblioteca también proporciona funciones de utilidad para crear y trabajar con observables. Estas funciones de utilidad se pueden utilizar para la conversión de código existente para operaciones asíncronas en observables, iterando a través de los valores en una secuencia, asignación de valores a diferentes tipos y redactar múltiples transmisiones (Angular/rx-library,2020).

4.2.8 Php

Php (acrónimo recursivo de PHP: Hypertext Preprocessor) es un lenguaje interpretado de código abierto, de propósito general ampliamente usado, diseñado especialmente para desarrollo web y que puede ser incrustado dentro de código HTML³² (¿Qué es PHP?,2020).

4.2.9 Composer

Con la versión 2.0.9 un manejador de paquetes para PHP que proporciona un estándar para administrar, descargar e instalar dependencias y librerías. Similar a NPM en Node.js y Bundler en Ruby (Composer, Basic usage,2020).

4.2.10 Symfony

Continuando con Symfony la versión a utilizar es la 4.3.4, Symfony es un entorno de trabajo estandarizado (framework PHP) que se utiliza para el desarrollo de aplicaciones web y es de los más utilizados en el entorno de desarrolladores de apps. En otras palabras, es una herramienta para desarrolladores para crear aplicaciones en PHP (Installing & Setting up the Symfony Framework,2020).

³² HyperText Markup Language

4.2.11 GIT

Para controlar todas las versiones de nuestro proyecto usaremos git, que es una herramienta que permite un control de versiones, es un sistema que registra los cambios realizados en un archivo o conjunto de archivos a lo largo del tiempo, de modo que podremos recuperar y controlar mejor la versión del proyecto (Git Reference,2020).

4.2.12 Lexik/jwt-authentication-bundle

Con la versión 2.6.5 de JWT³³ que define un mecanismo compacto y autocontenido de transmitir información de forma segura entre dos partes mediante un objeto JSON³⁴. Esta información puede ser verificada y confiada mediante una firma digital, normalmente usando algoritmos basados en RSA (las siglas representan a Rivest, Shamir y Adleman, y es un sistema criptográfico de clave pública que utiliza factorización de números enteros) o HMAC (código de autenticación de mensajes en clave-hash o código de autenticación de mensaje basado en hash). Básicamente permite enviar información autocontenido de una forma lo suficientemente compacta como para enviarse en una cabecera HTTP y donde todos los datos están contenidos en la transmisión sin necesidad de acceder a una fuente externa como una base de datos (Json Web Tokens en Symfony,2020).

4.3 Instalación del proyecto

Para iniciar la instalación del proyecto, hay que descargar el código de los siguientes repositorios que están disponibles en GitHub (Ver tabla 4).

Tabla 4: Repositorios³⁵

Desarrollo	Enlace
Backend	https://github.com/Blacky91/frontend2
Frontend	https://github.com/Blacky91/Repract_7

Para la descarga de estos repositorios se necesitará tener instalado previamente git, y para realizar esta acción en el bash ejecutamos la siguiente acción para ambos repositorios que están actualizados (Ver Figura 19).

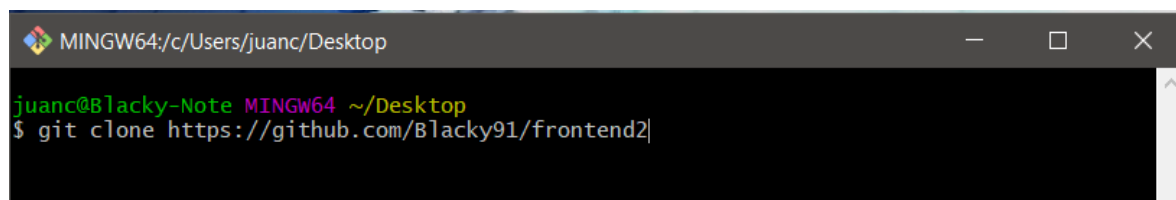
A screenshot of a terminal window with a dark background. The title bar at the top shows the path 'MINGW64:/c/Users/juanc/Desktop' and standard window controls. The terminal text shows the user 'juanc@Blacky-Note' in a green prompt, the shell 'MINGW64' in yellow, and the current directory '~/Desktop' in yellow. The command '\$ git clone https://github.com/Blacky91/frontend2' is entered in white text.

FIGURA 19: Clonación de repositorio³⁶

Luego de descargado el repositorio, en este caso el frontend, accedemos a la carpeta y realizamos las siguientes acciones en el terminal (Ver tabla 5).

³³ JSON Web Token

³⁴ JavaScript Object Notation

³⁵ Elaboración propia

³⁶ Elaboración propia

Tabla 5: Acciones para instalar el frontend³⁷

Paso	Codigo
1	Npm install
2	Ng build –prod
3	cd /var/www/html/
4	sudo rm -r *
5	cd /home/vps190/frontend2/dist/frontend2/
6	sudo cp -r * /var/www/html/

La primera acción, nos permite preparar el ambiente y la instalación de dependencia del proyecto. Luego se crea el producto angular de nuestro proyecto, este generará una carpeta llamada dist, en el tercer paso vamos a la carpeta donde almacenaremos nuestro producto y borraremos alguna antigua versión, si es que la hay, luego se volverá a la ruta y se copiará el producto a la ruta asignada.

Siguiendo con el backend se clonará el repositorio correspondiente, de manera similar al anterior (Ver figura 19), una vez clonado se realizará lo siguiente (Ver tabla 6).

Tabla 6: Acciones para el backend³⁸

Paso	Codigo
1	php bin/console server:stop
2	Git pull
3	composer install
4	Php bin/console server:star *:8000

Los primeros dos pasos son en caso de que el backend necesite alguna actualización, por lo que, al momento de especificar dichas acciones, es detener el backend que está en ejecución. Luego, realizar un pull de repositorio, para realizar un composer install para la instalación de las dependencias de nuestro backend, y una vez que se tenga todas las dependencias instaladas, se ejecuta el cuarto comando que nos permita ejecutar el backend en el puerto que se nos ha asignado.

³⁷ Elaboración propia

³⁸ Elaboración propia

5. PLATAFORMA WEB

La plataforma tendrá como nombre *ScoreMed*, en la página de inicio podemos apreciar que tendrá dos botones antes de iniciar la fase de login, el botón azul para el profesor y el botón verde para el estudiante, a la vez inicia con una breve reseña de nuestra plataforma (Ver figura 20).

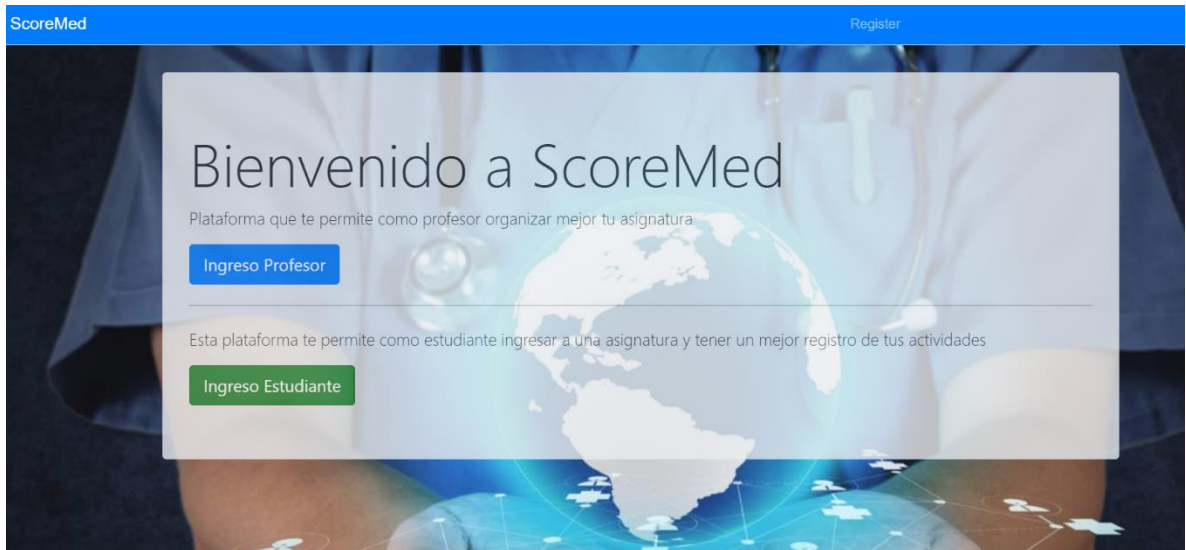


FIGURA 20: Home-page³⁹

Una vez que se accede como profesor o estudiante, se ingresará a la sección de login en donde el usuario podrá ingresar sus credenciales y podrá acceder a los beneficios que le facilitará esta plataforma (Ver figura 21).

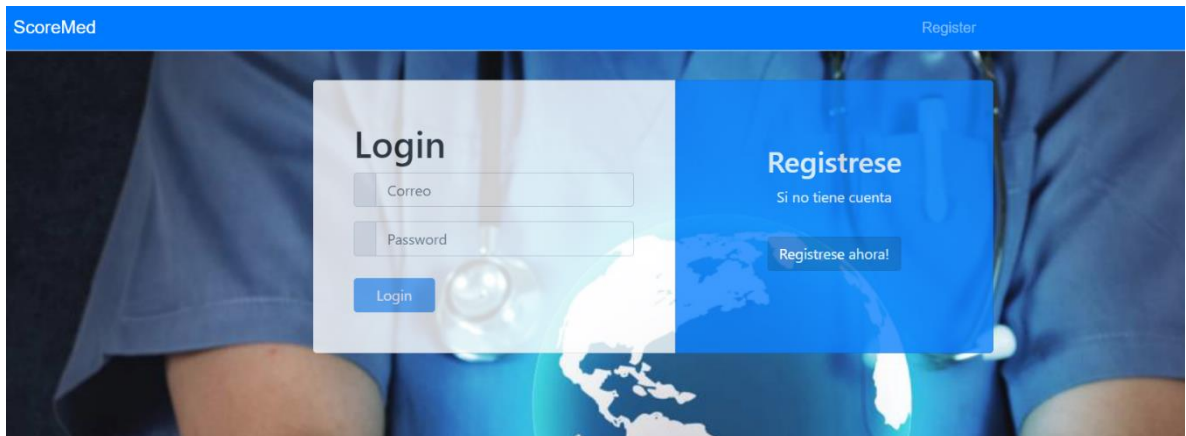


FIGURA 21: Login-page⁴⁰

³⁹ Elaboración propia

⁴⁰ Elaboración propia

En la barra de navegación, hay una opción de registro, y aparte en la sección del login también permite registrarse a esta plataforma. El formulario de registro es el que apreciamos a continuación (Ver figura 22), que nos permitirá capturar los datos de las entidades.

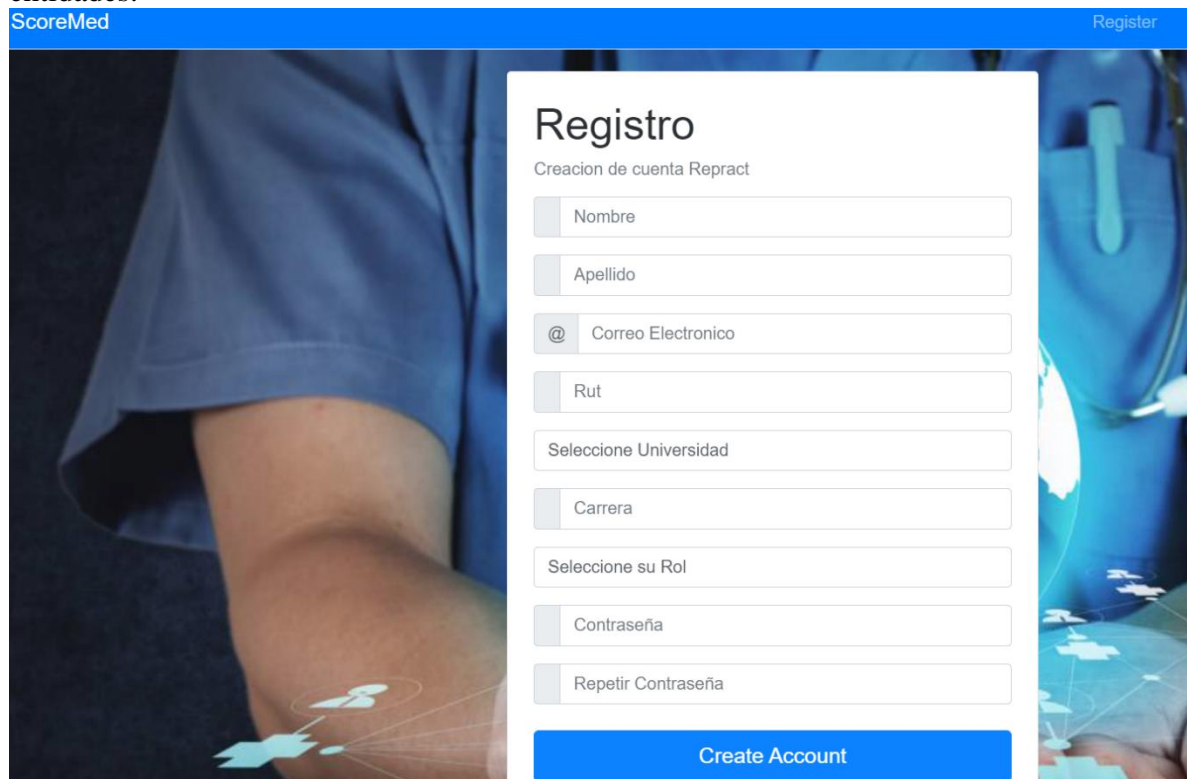


FIGURA 22: Register-page⁴¹.

5.1 Vista Profesor

Luego de pasar la fase de login, en el caso del profesor, ésta será la vista que tendrá (Ver figura 23). El botón mis cursos, son los cursos que el profesor ha creado y el botón colaborador en, serán las asignaturas en las que estará participando como profesor adjunto. En esta vista, se puede apreciar las diferencias de acciones, ya que un profesor colaborador solo podrá ver el curso, agregar actividades a esa asignatura y validar, mientras que el profesor creador del curso podrá ver el curso, editarlo, agregar actividades, validar y eliminar el curso.

⁴¹ Elaboración propia

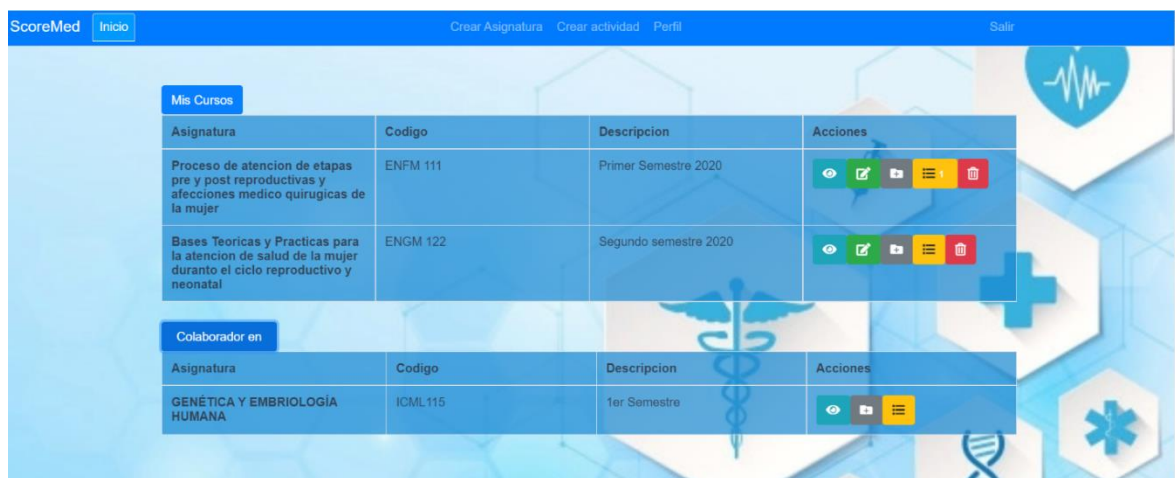


FIGURA 23: Vista post login del profesor⁴²

Para cuando un profesor se registre en la plataforma y es el encargado de un curso tendrá que llenar un formulario (Ver figura 24) en el cual podrá crear dicho curso, dándole así el nombre, el código correspondiente y alguna descripción. Posteriormente, luego de agregar dicha información podrá ir directamente mediante una pregunta si se quiere ingresar profesor clínico o tutor, y en caso de no tener claro en la sección de edición podrá agregar los profesores que participarán en el curso.

ScoreMed Inicio **Crear Asignatura** Crear actividad Perfil Salir

Creacion de **Asignatura**

Nombre Asignatura

Codigo Asignatura

Descripcion

FIGURA 24: Crear Asignatura

En el caso de querer editar el curso, por algún detalle al momento de crearlo se debe habilitar el *checkbox* para poder realizar esta acción, para luego actualizar la información del curso, y en caso de agregar a los profesores clínicos (Ver figura 25), se tendrá un filtro que facilita la búsqueda del profesor que se querrá incorporar, esa lista muestra a todos los profesores que estarán usando la plataforma.

⁴² Elaboración propia

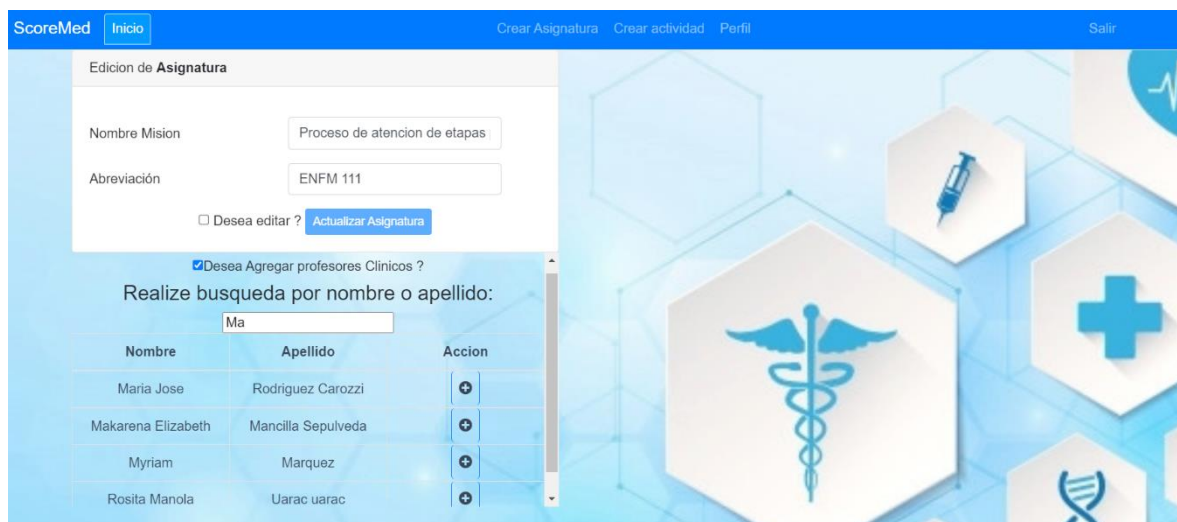


FIGURA 25: Edición de asignatura⁴³

La sección del filtro (Ver Figura 25), estará conformada por esta sección de código, la cual realiza la búsqueda por el nombre del profesor o el apellido que este ingrese en la plataforma.

```

1. import { Pipe, PipeTransform } from '@angular/core';
2. @Pipe({
3.   name: 'filter'
4. })
5.
6. export class FilterPipe implements PipeTransform {
7.   transform(value: any, arg: any): any {
8.     if(arg == "" || arg.length < 2){ return value}
9.     const resultadoTeacher =[];
10.    for (const teacherData of value){
11.      a. if(teacherData.name.toLowerCase().indexOf(arg.toLowerCase())
12.         > -1) {
13.        b. resultadoTeacher.push(teacherData);
14.        c. }
15.      d. else
16.        if(teacherData.lastname.toLowerCase().indexOf(arg.toLowerCase())
17.         > -1) {
18.        e. resultadoTeacher.push(teacherData);
19.        f. }
20.    }
21.    return resultadoTeacher;
22.  }
23. }
24.

```

Otra vista es la creación de actividad (Ver figura 26) en la que será muy similar a la creación de asignatura, pero el código, será brindado por la plataforma manteniendo una estructura para las actividades.

⁴³ Elaboración propia

FIGURA 26: Creación de actividad⁴⁴

Dentro de las opciones que deberá estar presente luego de logearse (Ver figura 21), está la opción de agregar actividades (Ver figura 27) que podrán realizarla las entidades con el rol de profesor. Además, se podrá apreciar un filtro que facilitará la búsqueda tanto por código de la actividad o el nombre de esta. Y en esta misma vista, se podrá vincular al curso, editar la actividad o incluso eliminarla.

Actividad	Codigo	Descripcion	Acciones
Curacion plana	ACME-0001	Limpieza de heridas	[+][✎][🗑]
Especuloscopia	ACME-0002	realizar exámenes o procedimientos diagnósticos y terapéuticos de cavidades corporales manteniendo abiertos sus orificios de entrada.	[+][✎][🗑]
Examen Fisico de mamas	ACME-0003	Examen	[+][✎][🗑]

FIGURA 27: Vista para agregar actividades⁴⁵

Volviendo al inicio (Ver figura 23) el botón amarillo, servirá para poder tener acceso a lo que es la sección de validación de las actividades (Ver figura 28), este botón tendrá un detalle que se podrá apreciar un número que representará una notificación y permitirá avisarles a los profesores cuantas actividades debe de validar. Con un filtro incluido que permite facilitar la búsqueda y una vez aceptada, la plataforma confirmará esta acción para luego refrescar la vista y continuar con las otras validaciones.

⁴⁴ Elaboración propia

⁴⁵ Elaboración propia



FIGURA 28: Validación de actividades⁴⁶

Para poder ver quiénes y cómo esta armado el curso, tenemos que dar una opción que les permita ver el curso (Ver figura 29). La primera pestaña de actividad, dará la posibilidad de ingresar la cantidad mínima que debe realizar un alumno de esa actividad, la pestaña de profesores y alumnos nos permitirá ver quienes son los participantes de este curso, y para finalizar, en la sección de estadística nos llevará a una próxima vista. Y en el icono que estará situado al lado derecho representará el resumen de todo el curso.

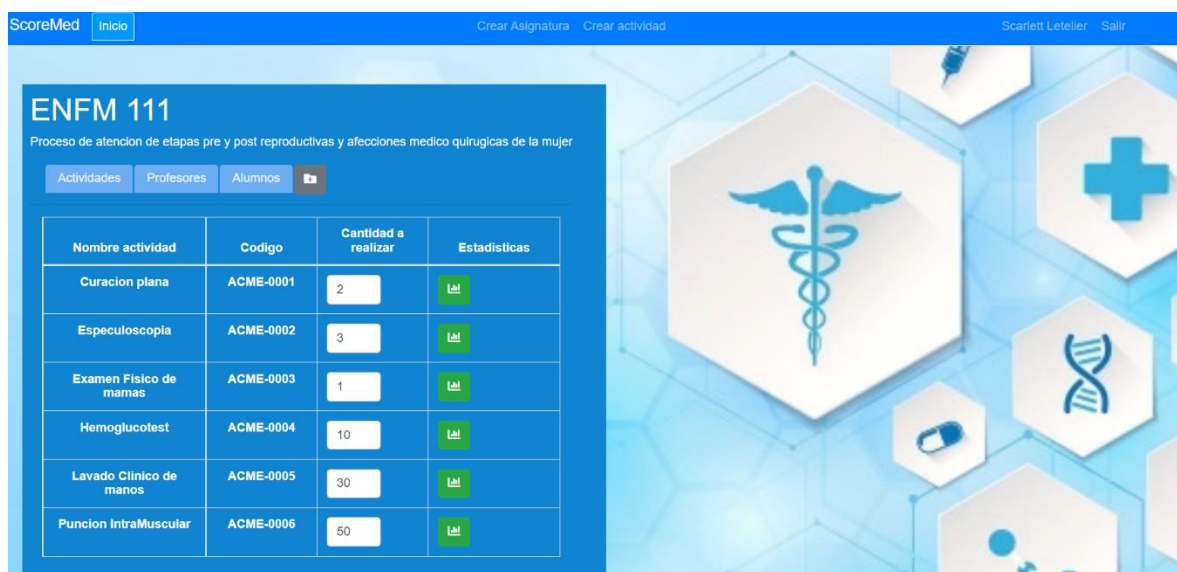


FIGURA 29: Ver el curso⁴⁷.

El botón de estadística, los llevará a una vista que estará conformada por un gráfico de barras (Ver figura 30), lo cual permitirá el siguiente análisis con las variables que se utilizaron. En el caso de la línea azul, representará la cantidad mínima ingresada por el profesor con la que debe cumplir el alumno, y la segunda variable que estará presente,

⁴⁶ Elaboración propia

⁴⁷ Elaboración propia

será los estudiantes y sus cantidades que fueron validadas. Luego de tener un mínimo de tres alumnos con actividades validadas se podrá visualizar el gráfico.



FIGURA 30: Gráfico de actividad⁴⁸

Anteriormente se hace referencia a un resumen, este resumen deberá tener como finalidad presentar a todos los estudiantes de la asignatura en filas (Ver figura 31), y las columnas serán las actividades por realizar, y estará relacionada de la siguiente forma:

- Completo: Es cuando un alumno supero o igualo el mínimo que asigno el profesor.
- Incompleto: Es cuando el alumno ingreso actividad, pero no logró el mínimo requerido.
- 0: Cuando el alumno no ha ingresado ninguna actividad al sistema.

ScoreMed Inicio Crear Asignatura Crear actividad Scarlett Letelier Salir						
ENFM 111						
Resumen	Curacion plana	Especulosopia	Examen Fisico de mamas	Hemoglucotest	Lavado Clinico de manos	Puncion IntraMuscular
Manuel Hugo	Completo	Completo	Completo	Incompleto	Incompleto	Incompleto
Richard Aranguiz	Completo	Incompleto	Completo	Incompleto	Incompleto	Incompleto
Emily Gallardo	Incompleto	0	0	0	0	0
Nicolas Sepulveda	Completo	Incompleto	0	Incompleto	Incompleto	0

FIGURA 31: Resumen de una asignatura⁴⁹

5.2 Vista Estudiante

Para el estudiante, luego de realizar su fase de login, deberá poder ver la lista de cursos a la que se podrá inscribir, y como se mencionó con anterioridad el objetivo es que el

⁴⁸ Elaboración propia

⁴⁹ Elaboración propia

alumno, pueda inscribir su curso, y esto podrá realizarse al hacer click al botón que estará presente (Ver figura 32).



Asignatura	Código	Descripción	Profesor	Acciones
Proceso de atención de etapas pre y post reproductivas y afecciones médicas quirúrgicas de la mujer	ENFM 111	Primer Semestre 2020	Scarlett Letelier	
Bases Teóricas y Prácticas para la atención de salud de la mujer durante el ciclo reproductivo y neonatal	ENGM 122	Segundo semestre 2020	Scarlett Letelier	
GENÉTICA Y EMBRIOLOGÍA HUMANA	ICML115	1er Semestre	Valentina Contreras	

FIGURA 32: Lista de cursos⁵⁰

Continuando con la vista de la plataforma, los estudiantes tendrán una sección donde podrán ver sus cursos. Dentro de ellas tendrán la opción ver curso, agregar actividad y mis estadísticas (Figura 33). Ver curso, que es la primera de estas acciones, es similar a la del profesor con diferencia de permisos, como por ejemplo el permiso para editar la cantidad mínima a realizar (Ver figura 29).



Asignatura	Código	Descripción	Profesor	Acciones
Proceso de atención de etapas pre y post reproductivas y afecciones médicas quirúrgicas de la mujer	ENFM 111	Primer Semestre 2020	Scarlett Letelier	
Bases Teóricas y Prácticas para la atención de salud de la mujer durante el ciclo reproductivo y neonatal	ENGM 122	Segundo semestre 2020	Scarlett Letelier	
GENÉTICA Y EMBRIOLOGÍA HUMANA	ICML115	1er Semestre	Valentina Contreras	

FIGURA 33: Vista de cursos inscritos

La segunda acción es ingresar actividad, esta permitirá al estudiante dar la información y con que profesor fue que realizó dicha actividad, en el caso que se muestra se verá que el alumno ingresa la actividad de especuloscopia e ingresa al profesor que deberá aceptar o rechazar esta acción (Ver figura 34).

⁵⁰ Elaboración propia

FIGURA 34: Vista Ingresar Actividad⁵¹

El dentro del objetivo principal, es poder tener una visualización de las actividades que permita tener en cuenta que cosa esta haciendo él como estudiante y que le falta por realizar, el gráfico de radar que se muestra a continuación, les pondrá las metas mínimas a realizar que es lo que pide el profesor, que son ingresados en la sección anterior y que es demostrado por el color rojo (Ver figura 35), mientras tanto el color azul serán las actividades que son aprobadas por el profesor o tutor correspondiente.

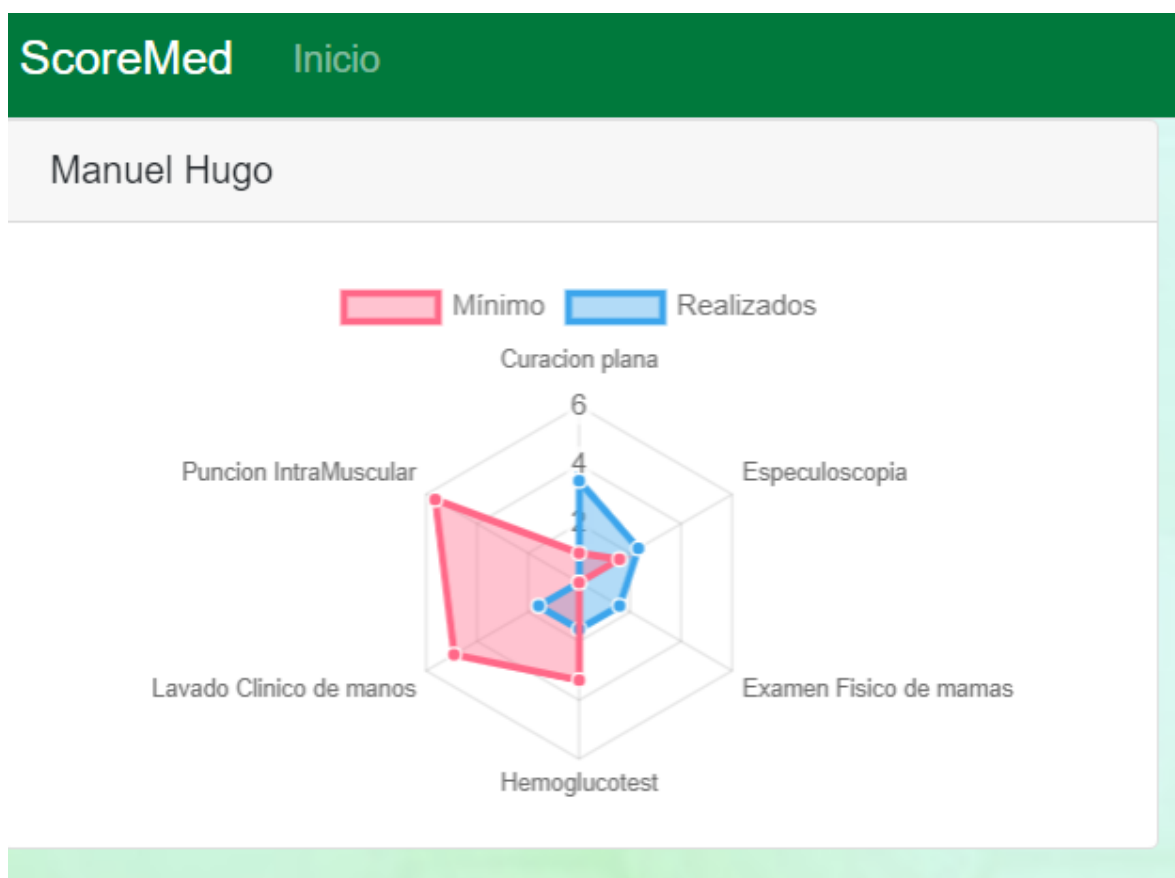


FIGURA 35: Estadísticas del estudiante⁵²

⁵¹ Elaboración propia

⁵² Elaboración propia

Dentro de las secciones de la barra de navegación se tendrá que el estudiante podrá ver todas sus actividades realizadas en diferentes en los cursos que él tiene inscrito, para una vez finalizada su carrera tendrá todos sus datos registrados de una manera más práctica y mucho más agradable de manipular (Ver figura 36). En esta vista, podrá ver el nombre de la actividad, la descripción de esta y la cantidad de veces que realizó esta actividad aprobada por el profesor y en la misma vista podrá ver cuáles fueron las rechazadas.

Actividad	Descripción	Cantidad Realizada
Curacion plana	Limpieza de heridas	2
Especulosopia	realizar exámenes o procedimientos diagnósticos y terapéuticos de cavidades corporales manteniendo abiertos sus orificios de entrada.	2
Hemoglucotest	edir lacantidad de glucosa en la sangre de una persona embarazada	1
Lavado Clínico de manos	Es la técnica básica utilizada para prevenir la transmisión de infecciones por vía contacto manual, eliminando arrastre los microorganismos que quedan en ellas	1

FIGURA 36: Récord de actividades⁵³.

⁵³ Elaboración propia

6.PROBLEMAS Y SOLUCIONES

6.1 Problema y solución 1

Uno de los problemas fue el manejo de información de cómo usar la extracción de roles y manejo de parámetros e identificadores de las asignaturas, alumnos y ese tipo de información que estará interactuando constantemente.

La solución que se utilizó fue utilizar una propiedad de angular llamada *localStorage*, el *localStorage* es una propiedad que acceden al objeto Storage y tienen la función de almacenar datos de manera local, Además, almacena la información de forma indefinida o hasta que se decida limpiar los datos del navegador, que para nuestro caso, será el logout o el fin de la sesión. Para guardar los datos, se podrán guardar de la siguiente manera;

```
1. Opción 1 -> localStorage.setItem(name, content)
2. Opción 2 -> localStorage.name = content
3. name = nombre del elemento
4. content = Contenido del elemento

5. localStorage.setItem('Nombre', Juan Guillermo')
6. localStorage.Apellido = 'Contreras Sepulveda'
```

Para nuestra opción a utilizar fue la opción 1, utilizando el `setItem`.

6.2 Problema y solución 2

Para la realización del gráfico de radar, adjuntaremos parte del código que demuestra la interacción de los datasets, pero a la vez se presentará un problema. El crecimiento del gráfico de radar es lineal, por lo que, si un alumno quisiera ingresar 100 actividades y en otra solo realizó 10, el gráfico quedará muy desproporcionado, por lo que para el usuario, no será una manera grata de ver si información representada en el gráfico de radar.

La solución aplicada a este problema, fue traemos los valores representados en números enteros, luego pasan por una función que los representa esos valores, en valores logarítmicos, después, esos valores son graficados por lo que el gráfico a utilizar no se verá tan desproporcionado y para ver el resultado correspondiente, el usuario solo debe pasar el puntero por su actividad y el tooltip le mostrará su valor original representado en el valor inicial.

```
1. ngOnInit() {
2. const id = this.route.snapshot.paramMap.get('id');
3. let result;
4. let resultName;
5. let resultMin;
6. let resultEstudiante;
7. this.issueService.myActivityValidate(this.id_user,id).subscribe(r
  es3 => {
    a. this.myValidates = res3.data;
    b. this.name = res3.data[1].name;
```

```

        c. this.lastname = res3.data[1].last_name;
8. });
9. //Usamos el id de la asignatura para traer la info de sus
   actividades
10.   this.issueService.misionInSubject(id).subscribe(res2 => {
      a. //necesitamos el servicio que trae la informacion del
         estudiante para crear su campo en el grafico
      b. // LA info que necesitamos es La cantidad que tiene de cada
         mision donde su campo es true o 1

      c. this.misionData = res2.data;
      d. result = this.misionData.map(a => a.shortNameMision);
      e. resultName = this.misionData.map(a => a.nameMision);
      f. resultMin = this.misionData.map(a => Math.log (a.minimo)/
         Math.log (2));
         i. resultEstudiante = this.myValidates.map(b => Math.log
            (b.realizado)/ Math.log (2));

      g. this.misionData={
      h. 'label': resultName,
      i. datas: [{
         i. 'data': resultMin,
         ii. 'label' : 'Mínimo'
         iii. }
         iv. ,
         v. {
         vi. 'data': resultEstudiante,
         vii. 'label': 'Realizados'
         viii. }
         ix. ],
      j. options: {
         i. tooltips: {
         ii. callbacks: {
         iii. label: function (tooltipItems, data) {
         iv. return Math.pow(2, tooltipItems.yLabel).toFixed(2);
         v. },
         vi. title:function(tooltipItems, data) {
         vii. return " El Valor es: ";
         viii. }
         ix. }
         x. }
         k. },
         l. }
      m. this.chartReady=true;
      n. return [this.misionData,this.myValidates];
11.   });
12.   }

13.   public radarChartType = 'radar';

```

6.3 Problema y solución 3

Una vez actualizado el servidor con el backend y frontend, correspondiente, se presentó el siguiente problema dentro de la aplicación:

Error 1055: SELECT list is not in GROUP BY clause and contains nonaggregated column
incompatible with sql_mode=only_full_group_by

Este problema se originó por lo siguiente, el servidor actualizó a otra versión de mysql a la versión 5.7, la cual ha presentado algunos problemas en el uso del *group by*.

Dentro de las soluciones que se encontraron fueron las siguiente:

- Volver a una versión anterior de MySQL lista para usar con el software de la aplicación que utiliza.
- Cambiar el `sql_mode` su servidor para deshacerse del modo *ONLY_FULL_GROUP_BY* recién configurado.

La primera opción no se utilizó por la idea de perder la data que se tenía en la base de datos, y que se presenten más errores, y eran dos queries que se veían afectadas por este error. La opción dos fue la utilizada, pero encontrar el archivo `.init` fue lo complicado, debido a que se presentaron archivos vacíos, o archivos que no utilizaban el `sql_mode`, dentro de varias sintaxis se reiniciaba el mysql y el cambio no tenía efecto.

Una opción que dio efecto pero fue poca práctica era antes de ejecutar la query era realizar lo siguiente en mysql.

```
SET sql_mode =  
'STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_AUTO_CREATE_USER,NO_ENGINE_SUBSTITUTION'
```

Pero es poco practico realizarla en cada consulta, por lo que la opción que funciono de mejor forma fue la siguiente

```
1. #!/bin/bash  
2. if [ -e /etc/mysql/conf.d/disable_strict_mode.cnf ]; then  
3. echo "Disable strict mode config already exists"  
4. else  
5. touch /etc/mysql/conf.d/disable_strict_mode.cnf  
6. echo "Creating file..."  
7. echo "Adding config..."  
8. printf  
   "[mysqld]\nsql_mode=IGNORE_SPACE,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERR  
   OR_FOR_DIVISION_BY_ZERO,NO_AUTO_CREATE_USER,NO_ENGINE_SUBSTITUTIO  
   N\n" > /etc/mysql/conf.d/disable_strict_mode.cnf  
9. echo "Restarting MySql..."  
10.    service mysql restart  
11.    echo "Done"  
12.    fi
```

En pocas palabras, el script revisa si existe el modo estricto, y en caso de no existir, crea un archivo que omite la información de *ONLY_FULL_GROUP_BY*, permitiendo así poder realizar la consulta de manera normal, esta solución se ve replicada en la plataforma (Ver Figura 23).

7. TESTING

El testing o prueba de software, es el proceso en el cual se lleva a cabo una evaluación y verificación del producto a entregar. En este caso será una plataforma web, con el fin de saber si realiza lo que se debe y si es de fácil manejo para el usuario.

Hay muchos tipos diferentes de pruebas de software, cada una con objetivos y estrategias específicos:

- Pruebas de aceptación: Verificar si todo el sistema funciona según lo previsto.
- Pruebas de integración: Se asegura de que los componentes o funciones del software operen juntos.
- Examen de la unidad: Valida que cada unidad de software funcione como se esperaba. Una unidad es el componente comprobable más pequeño de una aplicación.
- Pruebas funcionales: Verifica las funciones emulando escenarios comerciales, basados en requisitos funcionales. Las pruebas de caja negra son una forma común de verificar funciones.
- Pruebas de rendimiento: Prueba el rendimiento del software en diferentes cargas de trabajo. Las pruebas de carga, por ejemplo, se utilizan para evaluar el rendimiento en condiciones de carga reales.
- Pruebas de regresión: Verifica si las nuevas funciones rompen o degradan la funcionalidad. Las pruebas de cordura se pueden utilizar para verificar menús, funciones y comandos a nivel de superficie, cuando no hay tiempo para una prueba de regresión completa.
- Pruebas de estrés: Prueba cuánta tensión puede soportar el sistema antes de que falle. Está considerada como un tipo de prueba no funcional.
- Pruebas de usabilidad: Valida qué tan bien un cliente puede usar un sistema o una aplicación web para completar una tarea.

En sí, se tiene una idea de los problemas principales que se presentan en los testing. A continuación, se presentará un pequeño grupo de posibles problemas o detalles luego de realizar un testing, como:

- Defectos arquitectónicos.
- Decisiones de diseño deficientes.
- Funcionalidad no válida o incorrecta.
- Vulnerabilidades de seguridad.
- Problemas de escalabilidad.

La prueba que se realizó fue la de usabilidad y de aceptación. En este caso un profesor del área de Informática Christian A. Lazo Ramírez y un profesor del área de Obstetricia y Puericultura Scarlett Letelier S. utilizaron la plataforma con el fin de ver si se pueden cumplir las expectativas esperadas.

7.1 Expectativas

Previo al testing surge la incógnita ¿Qué se espera de las pruebas?, y dentro de las expectativas es poder saber que tan intuitiva es la plataforma tanto para profesores y estudiantes, y que tan agradable es para el usuario. Todas las acciones estarán especificadas en el Manual de Usuario (Ver anexo B).

7.1.1 Expectativas profesor

Se espera que un profesor pueda realizar las siguientes pruebas:

- Registrarse a la plataforma.
- Logearse a la plataforma.
- Crear una asignatura.
- Editar una asignatura.
- Eliminar una asignatura.
- Crear actividades.
- Agregar actividades a una asignatura.
- Validar actividades de los estudiantes.
- Ver gráficos por actividad.
- Ver resultados generales de la asignatura.
- Vincular a profesores clínicos a una asignatura.

7.1.2 Expectativas estudiante

Dentro de las expectativas que se esperan que pueda realizar un estudiante están las siguientes:

- Registrarse a la plataforma.
- Logearse a la plataforma.
- Vincularse/Inscribirse a una asignatura.
- Ver las actividades a realizar en una asignatura.
- Ingresar una actividad con un profesor asignado.
- Ver récord de actividades aceptadas y rechazadas.
- Ver gráficos.

7.2 Resultados

A continuación Scarlett Letelier S junto a Christian Lazo Ramírez dieron su retroalimentación sobre la prueba de uso en *ScoreMed*.

“1.- Registro e ingreso plataforma.

De acuerdo con las etapas y pasos referidos a la utilización de la plataforma de *ScoreMed* descritas en el manual de uso de la misma, cabe señalar que en la etapa de registro es

posible la creación de usuarios tanto en el rol de profesores, como estudiantes, siendo factible además el ingreso sin dificultad de usuarios registrados anteriormente.

2.- Ingreso con rol profesor.

Tal como lo describe el Manual de usuario, al ingresar en la plataforma bajo el rol de profesor en la página de “Inicio” es posible crear asignatura y docentes para cada signatura, sin embargo, no es posible crear actividad, ya que en la página que se despliega, específicamente en el recuadro para el código actividad no es posible sobrescribir (Ver figura 37).



FIGURA 37 :Feedback Editor⁵⁴

En la misma página de inicio es posible visualizar tabla con las asignaturas creadas, así como los botones para visualizar las asignaturas asociadas al docente tanto como responsable (“Mis cursos”) como, “Colaborador”.

En cualquiera de ambos roles docentes es posible visualizar el listado de asignaturas donde cada profesor tiene injerencia (Ver figura 38), cada asignatura posee sus íconos o botones interactivos los cuales permiten el despliegue de una pantalla por cada uno, con información correspondiente a: las características del curso, listado de profesores, estudiantes y actividades clínicas asociadas a la asignatura con la cantidad mínima a realizar por cada estudiante y estadística por curso; botón para vincular actividades clínicas (Ver figura 39) a la asignatura (también como colaborador), además de botón para desplegar pantalla para la validación de actividades o score realizadas por los estudiantes (Ver figura 40), o rechazo de las mismas, destaca que en dicho ícono se señala el número de actividades pendientes de validación. En este sentido, se valora que el desarrollo logró que el docente responsable de la asignatura tenga la posibilidad de ver a los docentes de su asignatura y las actividades que tienen pendientes de validación.

⁵⁴ Elaboración del cliente.



FIGURA 38: Feedback de la Home-page⁵⁵.



FIGURA 39: Feedback Actividades⁵⁶.



FIGURA 40 :Feedback validación⁵⁷.

Cabe sugerir dada esta las características de esta pantalla, que este botón back quede operativo pues regresa a la página de inicio mostrando el listado de “Mis cursos” y que sería de utilidad replicar el ícono back en las otras pantallas para volver a la página anterior sobre todo cuando se pretende volver al listado de cursos como “Colaborador”.

⁵⁵ Elaboración del cliente.

⁵⁶ Elaboración del cliente.

⁵⁷ Elaboración del cliente.

3.- Ingreso con rol estudiante.

Al ingresar como estudiante nuevo se visibilizan los cursos (Ver Figura 41) y es factible en el icono acciones anexar asignaturas al perfil, visibilizándose luego en “Mis cursos”



FIGURA 41: Feedback Home-page estudiante⁵⁸.

En la pantalla de cursos cada asignatura posee sus íconos o botones interactivos los cuales permiten el despliegue de una pantalla por cada uno (Ver figura 42), con información correspondiente a las características del curso, con el listado de profesores y actividades clínicas asociadas a la asignatura con la cantidad mínima exigible y un gráfico de su progreso (Ver figura 43).



FIGURA 42: Feedback Mis Cursos⁵⁹.

⁵⁸ Elaboración del cliente.

⁵⁹ Elaboración del cliente.

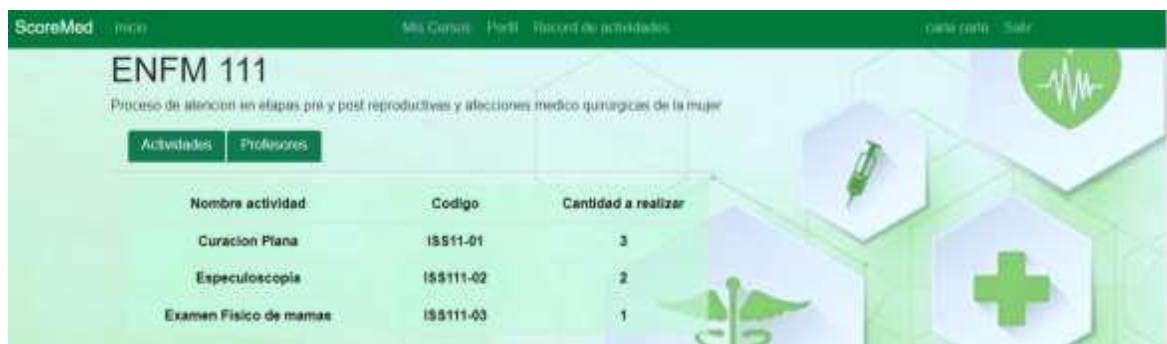


FIGURA 43: Feedback Ver Curso⁶⁰.

Además, se logró adecuadamente en el desarrollo que en la pantalla de Récord de actividades el estudiante ingrese al docente y actividad a validar (Ver figura 44), se sugiere agregar fecha en el que se realizó la actividad, incluso sería ideal la hora de la misma. Se valora la modificación sugerida con relación a que mediante cuadros de texto estudiantes y docentes sean informados de que sus acciones en la plataforma fueron realizadas.



FIGURA 44: Feedback Ingresar actividad⁶¹.

El estudiante puede verificar las actividades clínicas que le fueron aprobadas, así como, rechazadas (Ver Figura 45), ante esto sería perfectible este recurso para el estudiante si además visualizara aquellas pendientes.



FIGURA 45: Feedback Récord de actividades⁶².

Cabe señalar que las pruebas de la plataforma se realizaron previo al conocimiento del Manual de usuario siendo bastante intuitivo su uso.

⁶⁰ Elaboración del cliente.

⁶¹ Elaboración del cliente.

⁶² Elaboración del cliente.

Finalmente, se concluye que la plataforma desarrollada cumple con el objetivo de llevar un registro escrito, de manera virtualizada y aun cuando es perfectible, permite desde la perspectiva docente llevar un registro básico de las actividades clínicas efectuadas por cada estudiante, un curso y contrastar, el número de actividades realizadas con los mínimos exigibles, el cumplimiento de las metas en este sentido de cada asignatura, y permite al estudiante visibilizar su progreso y hacer gestión del mismo.

Prof. Scarlett Letelier”

8. CONCLUSION

La idea inicial al comenzar este proyecto fue disminuir la carga laboral a los profesores del área de la salud, y se demostró que organizando la información y adquiriendo una buena comunicación con el cliente se logró administrar la información para profesores y dentro del trayecto del desarrollo una mejor administración de las actividades para los estudiantes.

Gracias a la creación de la plataforma web se ha permitido adquirir más conocimientos y experiencia al momento de desarrollar con un cliente de un área específica, y a la vez se aplicó el concepto de que una plataforma web debe ser sencilla, interactiva e intuitiva para los usuarios.

La decisión elegida de ir por una plataforma gamificada, es que esta técnica incorpora la mecánica de los juegos al ámbito educativo-profesional con el fin de conseguir mejores resultados, en este caso, sería establecer un mínimo para el estudiante y con el fin de que pueda igualarlo o en el mejor de los casos superar la marca impuesta por el profesor.

Si bien se logró realizar una serie de pruebas con usuarios específicos durante la fase de desarrollo, aún queda pendiente ejecutar una prueba con un curso durante un semestre completo para así poder llevar a cabo otro tipo de pruebas como la de estrés a la plataforma.

Este proyecto de título puede ser considerado un prototipo, ya que dentro del impacto a largo plazo, se espera que pueda ser utilizada a nivel nacional por todas las universidades considerando todas las ramas de la salud y lograr así un mejor análisis dentro del país. Gracias a las tecnologías utilizadas como lo son Symfony y Angular, las cuales fueron las más destacadas de este proyecto, las mejoras que se esperan a largo plazo no deberían ser un gran problema, lo que si realizar un buen análisis va a ser clave para seguir mejorando la plataforma.

9.REFERENCIAS

Angular/core. Accedido el 15 de enero, 2020, desde <https://angular.io/api/core>.

Angular/forms. Accedido el 17 de enero, desde 2020, <https://angular.io/api/forms>.

Angular/router. Accedido el 15 de enero, 2020, desde <https://angular.io/api/core>.

Angular/rx-library. Accedido el 25 de enero,2020, desde <https://angular.io/guide/rx-library>.

Arawjo, I., Wang, C.-Y., Myers, A. C., Andersen, E., & Guimbretière, F. (2017). Teaching Programming with Gamified Semantics. In Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems. <https://doi.org/10.1145/3025453.3025711>.

Casper Hartevelde Northeastern University, Boston, MA, USA, & Steven C. Sutherland University of Houston-Clear Lake, Houston, TX, USA. (n.d.). Personalized Gaming for Motivating Social and Behavioral Science Participation. ACM Conferences. Retrieved March 28, 2022, from <https://dl.acm.org/doi/abs/10.1145/3039677.3039681>.

Clark C. Abt (1970). Serious Game. Lanham, Maryland, Estados Unidos de América: Viking Press.

ChartJs. Accedido el 25 de enero,2020, desde <https://www.chartjs.org/docs/latest/>.

Composer,Basic usage, Accedido el 10 de enero,2020 desde <https://getcomposer.org/doc/01-basic-usage.md>.

Darina Dicheva Winston-Salem State University, Winston Salem, NC, USA, Keith Irwin Winston Salem State University, Winston Salem, NC, USA, & Christo Dichev Winston-Salem State University, Winston Salem, NC, USA. (n.d.). *OneUp*. ACM Conferences. Retrieved March 28, 2022, from <https://dl.acm.org/doi/abs/10.1145/3287324.328748>.

Git Reference, Accedido el 04 de enero del 2020, desde <https://git-scm.com/docs>.

Hege Mari Johnsen; Fossum, Mariann; Vivekananda-Schmidt, Pirashanthie, Fruhling, Ann, Slettebø, Åshild(2018). Developing a Serious Game for Nurse Education. (n.d.). Retrieved March 28, 2022,from

<https://www.proquest.com/openview/abf7f50cb3ef07e2400781600edd28a1/1?pq-origsite=gscholar&cbl=47810>.

Installing & Setting up the Symfony Framework. Accedido el 08 de enero del 2020, desde <https://symfony.com/doc/4.3/setup.html>.

Json Web Tokens en Symfony, Accedido el 4 de enero del 2020, <https://www.kroatoan.es/articulos/json-web-tokens-en-symfony/>

Patrícia Gomes Fernandes Matsubara, Caroline Lima Corrêa Da Silva. (2017). Developing a Serious Game for Nurse Education. (n.d.). Retrieved March 28, 2022, from <https://www.proquest.com/openview/abf7f50cb3ef07e2400781600edd28a1/1?pq-origsite=gscholar&cbl=47810>.

Restfull API con Symfony + JWT, Accedido el 5 de enero del 2020, <https://medium.com/francisco-ugalde/restful-api-con-symfony-4-jwt-parte-2-112cc23e335d>.

¿Qué es PHP?, Accedido el 10 de enero,2020, desde <https://www.php.net/manual/es/intro-what-is.php>.

ANEXOS

Anexo A: Entidades.

User.php

```
use Doctrine\Common\Collections\Collection;
use Doctrine\ORM\Mapping as ORM;
use Symfony\Component\Security\Core\User\UserInterface;
use Doctrine\Common\Collections\ArrayCollection;
use DateTime;
use JMS\Serializer\Annotation as Serializer;

/**
 * @ORM\Entity(repositoryClass="App\Repository\UserRepository")
 * @ORM\HasLifecycleCallbacks()
 */
class User implements UserInterface
{
    /**
     * @ORM\Id()
     * @ORM\GeneratedValue()
     * @ORM\Column(type="integer")
     */
    private $id;

    /**
     * @ORM\Column(type="string", length=180, unique=true)
     */
    private $username;

    /**
     * @ORM\Column(type="json")
     */
    private $roles = [];

    /**
     * @var string The hashed password
     * @ORM\Column(type="string")
     */
    private $password;

    /**
     * @ORM\Column(type="string", length=150)
     */
    private $name;

    /**
     * @ORM\Column(type="string", length=200)
     */
    private $lastName;

    /**
     * @ORM\Column(type="string", length=150)
     */
}
```



```

    */
private $university;

/**
 * @ORM\Column(type="string", length=15)
 */
private $rut;

/**
 * @ORM\ManyToOne(targetEntity="App\Entity\Subject",
 * mappedBy="relationUser")
 */
private $relationSubjects;

/**
 * @ORM\Column(name="created_at", type="datetime")
 */
private $createdAt;

/**
 * @ORM\Column(name="updated_at", type="datetime")
 */
private $updatedAt;

public function __construct()
{
    $this->relationSubjects = new ArrayCollection();
}

public function getId(): ?int
{
    return $this->id;
}

/**
 * A visual identifier that represents this user.
 *
 * @see UserInterface
 */
public function getUsername(): string
{
    return (string) $this->username;
}

public function setUsername($username): self
{
    $this->username = $username;

    return $this;
}

/**
 * @see UserInterface
 */
/**
 * Get roles

```

```

*
* @return array
public function getRoles(): array
{
    $roles = $this->roles;
    // guarantee every user at least has ROLE_USER
    //$roles[] = 'ROLE_USER';

    return array_unique($roles);
}
/**
 * Set roles
 *
 * @param array $roles
 *
 * @return User
 */
public function setRoles($roles)
{
    $this->roles = $roles;

    return $this;
}

/**
 * @see UserInterface
 */
public function getPassword(): string
{
    return (string) $this->password;
}

public function setPassword(string $password): self
{
    $this->password = $password;

    return $this;
}

/**
 * @see UserInterface
 */
public function getSalt()
{
    // not needed when using the "bcrypt" algorithm in
    security.yaml
}

/**
 * @see UserInterface
 */
public function eraseCredentials()
{
    // If you store any temporary, sensitive data on the user,
    clear it here
    // $this->plainPassword = null;

```

```

    }

    public function getName(): ?string
    {
        return $this->name;
    }

    public function setName( $name): self
    {
        $this->name = $name;

        return $this;
    }

    public function getLastName(): ?string
    {
        return $this->lastName;
    }

    public function setLastName( $lastName): self
    {
        $this->lastName = $lastName;

        return $this;
    }

    public function getUniversity(): ?string
    {
        return $this->university;
    }

    public function setUniversity($university): self
    {
        $this->university = $university;

        return $this;
    }

    public function getRut(): ?string
    {
        return $this->rut;
    }

    public function setRut( $rut): self
    {
        $this->rut = $rut;

        return $this;
    }

    /**
     * @return Collection|Subject[]
     */
    public function getRelationSubjects(): Collection
    {
        return $this->relationSubjects;
    }

```

```

    }

    public function addRelationSubject(Subject $relationSubject): self
    {
        if (!$this->relationSubjects->contains($relationSubject)) {
            $this->relationSubjects[] = $relationSubject;
            $relationSubject->addRelationUser($this);
        }

        return $this;
    }

    public function removeRelationSubject(Subject $relationSubject):
        self
    {
        if ($this->relationSubjects->contains($relationSubject)) {
            $this->relationSubjects->removeElement($relationSubject);
            $relationSubject->removeRelationUser($this);
        }

        return $this;
    }
    /**
     * @return mixed
     */
    public function getCreatedAt()
    {
        return $this->createdAt;
    }
    /**
     * @param mixed $createdAt
     * @return self
     */
    public function setCreatedAt($createdAt)
    {
        $this->createdAt = $createdAt;

        return $this;
    }
    /**
     * @return mixed
     */
    public function getUpdatedAt()
    {
        return $this->updatedAt;
    }
    /**
     * @param mixed $updatedAt
     * @return self
     */
    public function setUpdatedAt($updatedAt)
    {
        $this->updatedAt = $updatedAt;

        return $this;
    }
}

```

```

/**
 * @ORM\PrePersist
 * @ORM\PreUpdate
 */
public function updatedTimestamps()
{
    $dateTimeNow = new DateTime('now');
    $this->setUpdatedAt($dateTimeNow);
    if ($this->getCreatedAt() === null) {
        $this->setCreatedAt($dateTimeNow);
    }
}

```

Subject.php

```

use DateTime;
use Doctrine\Common\Collections\ArrayCollection;
use Doctrine\Common\Collections\Collection;
use Doctrine\ORM\Mapping as ORM;

/**
 * @ORM\Entity(repositoryClass="App\Repository\SubjectRepository")
 * @ORM\HasLifecycleCallbacks()
 */
class Subject
{
    /**
     * @ORM\Id()
     * @ORM\GeneratedValue()
     * @ORM\Column(type="integer")
     */
    private $id;

    /**
     * @ORM\Column(type="string", length=150)
     */
    private $subjectName;

    /**
     * @ORM\Column(type="string", length=255)
     */
    private $subjectCode;

    /**
     * @ORM\Column(type="datetime")
     */
    private $datetime;

    /**
     * @ORM\ManyToMany(targetEntity="App\Entity\User",
     *                 inversedBy="relationSubjects")
     */
    private $relationUser;
}

```

```

/**
 * @ORM\Column(type="datetime")
 */
private $createdAt;

/**
 * @ORM\Column(type="datetime")
 */
private $updatedAt;

/**
 * @ORM\Column(type="integer")
 */
private $createdBy;

/**
 * @ORM\ManyToOne(targetEntity="App\Entity\Mision",
    mappedBy="relationSubject")
 */
private $misions;

/**
 * @ORM\Column(type="text", nullable=true)
 */
private $descriptionSubject;

/**
 * @ORM\ManyToOne(targetEntity="App\Entity\ActivityData",
    mappedBy="Id")
 */
private $activityData;

public function __construct()
{
    $this->relationUser = new ArrayCollection();
    $this->misions = new ArrayCollection();
    $this->activityData = new ArrayCollection();
}

public function getId(): ?int
{
    return $this->id;
}

public function getSubjectName(): ?string
{
    return $this->subjectName;
}

public function setSubjectName(string $subjectName): self
{
    $this->subjectName = $subjectName;

    return $this;
}

```

```

public function getSubjectCode(): ?string
{
    return $this->subjectCode;
}

public function setSubjectCode(string $subjectCode): self
{
    $this->subjectCode = $subjectCode;

    return $this;
}

public function getDatetime(): ?\DateTimeInterface
{
    return $this->datetime;
}

public function setDatetime(\DateTimeInterface $datetime): self
{
    $this->datetime = $datetime;

    return $this;
}

/**
 * @return Collection|User[]
 */
public function getRelationUser(): Collection
{
    return $this->relationUser;
}

public function addRelationUser(User $relationUser): self
{
    if (!$this->relationUser->contains($relationUser)) {
        $this->relationUser[] = $relationUser;
    }

    return $this;
}

public function removeRelationUser(User $relationUser): self
{
    if ($this->relationUser->contains($relationUser)) {
        $this->relationUser->removeElement($relationUser);
    }

    return $this;
}

/**
 * @return mixed
 */
public function getCreatedAt()
{
    return $this->createdAt;
}

```

```

/**
 * @param mixed $createdAt
 * @return self
 */
public function setCreatedAt( $createdAt)
{
    $this->createdAt = $createdAt;

    return $this;
}
/**
 * @return mixed
 */
public function getUpdatedAt()
{
    return $this->updatedAt;
}
/**
 * @param mixed $updatedAt
 * @return self
 */
public function setUpdatedAt($updatedAt)
{
    $this->updatedAt = $updatedAt;

    return $this;
}
/**
 * @ORM\PrePersist
 * @ORM\PreUpdate
 */
public function updatedTimestamps()
{
    $dateTimeNow = new DateTime('now');
    $this->setUpdatedAt($dateTimeNow);
    if ($this->getCreatedAt() === null) {
        $this->setCreatedAt($dateTimeNow);
    }
}

public function getCreatedBy(): ?int
{
    return $this->createdBy;
}

public function setCreatedBy(int $createdBy): self
{
    $this->createdBy = $createdBy;

    return $this;
}
/**
 * @return Collection|Mision[]
 */
public function getMisions(): Collection

```



```

{
    return $this->misions;
}

public function addMision(Mision $mision): self
{
    if (!$this->misions->contains($mision)) {
        $this->misions[] = $mision;
        $mision->addRelationSubject($this);
    }

    return $this;
}

public function removeMision(Mision $mision): self
{
    if ($this->misions->contains($mision)) {
        $this->misions->removeElement($mision);
        $mision->removeRelationSubject($this);
    }

    return $this;
}

public function getDescriptionSubject(): ?string
{
    return $this->descriptionSubject;
}

public function setDescriptionSubject(?string $descriptionSubject):
    self
{
    $this->descriptionSubject = $descriptionSubject;

    return $this;
}

/**
 * @return Collection|ActivityData[]
 */
public function getActivityData(): Collection
{
    return $this->activityData;
}

public function addActivityData(ActivityData $activityData): self
{
    if (!$this->activityData->contains($activityData)) {
        $this->activityData[] = $activityData;
        $activityData->addId($this);
    }

    return $this;
}

```

```

    public function removeActivityData(ActivityData $activityData):
        self
    {
        if ($this->activityData->contains($activityData)) {
            $this->activityData->removeElement($activityData);
            $activityData->removeId($this);
        }

        return $this;
    }
}

```

Mision.php

```

use Doctrine\Common\Collections\ArrayCollection;
use Doctrine\Common\Collections\Collection;
use Doctrine\ORM\Mapping as ORM;

/**
 * @ORM\Entity(repositoryClass="App\Repository\MisionRepository")
 */
class Mision
{
    /**
     * @ORM\Id()
     * @ORM\GeneratedValue()
     * @ORM\Column(type="integer")
     */
    private $id;

    /**
     * @ORM\Column(type="string", length=255)
     */
    private $nameMision;

    /**
     * @ORM\Column(type="string", length=150)
     */
    private $shortNameMision;

    /**
     * @ORM\ManyToMany(targetEntity="App\Entity\Subject",
     *                 inversedBy="misions")
     */
    private $relationSubject;

    /**
     * @ORM\Column(type="text", nullable=true)
     */
    private $descriptionMision;

    public function __construct()
    {
        $this->relationSubject = new ArrayCollection();
    }
}

```

```

public function getId(): ?int
{
    return $this->id;
}

public function getNameMision(): ?string
{
    return $this->nameMision;
}

public function setNameMision(string $nameMision): self
{
    $this->nameMision = $nameMision;

    return $this;
}

public function getShortNameMision(): ?string
{
    return $this->shortNameMision;
}

public function setShortNameMision(string $shortNameMision): self
{
    $this->shortNameMision = $shortNameMision;

    return $this;
}

/**
 * @return Collection|Subject[]
 */
public function getRelationSubject(): Collection
{
    return $this->relationSubject;
}

public function addRelationSubject(Subject $relationSubject): self
{
    if (!$this->relationSubject->contains($relationSubject)) {
        $this->relationSubject[] = $relationSubject;
    }

    return $this;
}

public function removeRelationSubject(Subject $relationSubject):
self
{
    if ($this->relationSubject->contains($relationSubject)) {
        $this->relationSubject->removeElement($relationSubject);
    }

    return $this;
}

```

```

    public function getDescriptionMision(): ?string
    {
        return $this->descriptionMision;
    }

    public function setDescriptionMision(?string $descriptionMision):
        self
    {
        $this->descriptionMision = $descriptionMision;

        return $this;
    }
}

```

ActivityData.php

```

use Doctrine\Common\Collections\Collection;
use Doctrine\ORM\Mapping as ORM;
use Symfony\Component\Security\Core\User\UserInterface;
use Doctrine\Common\Collections\ArrayCollection;
use DateTime;
use DateTimeZone;

/**
 * @ORM\Entity(repositoryClass="App\Repository\ActivityDataRepository")
 */
class ActivityData
{
    /**
     * @ORM\Id()
     * @ORM\GeneratedValue()
     * @ORM\Column(type="integer")
     */
    private $id;

    /**
     * @ORM\Column(type="integer")
     */
    private $generateFor;

    /**
     * @ORM\Column(type="boolean", nullable=true)
     */
    private $isValidate;

    /**
     * @ORM\Column(type="integer")
     */
    private $teacherInfo;

    /**
     * @ORM\Column(type="integer")
     */
    private $misionData;
}

```

```

/**
 * @ORM\ManyToOne(targetEntity="App\Entity\Subject",
 *               inversedBy="activityData")
 */
private $Id;

/**
 * @ORM\Column(type="datetime")
 */
private $createActivity;

public function __construct()
{
    $this->Id = new ArrayCollection();
}

public function getId(): ?int
{
    return $this->id;
}

public function getGenerateFor(): ?int
{
    return $this->generateFor;
}

public function setGenerateFor(int $generateFor): self
{
    $this->generateFor = $generateFor;

    return $this;
}

public function getIsValidate(): ?bool
{
    return $this->isValidate;
}

public function setIsValidate(?bool $isValidate): self
{
    $this->isValidate = $isValidate;

    return $this;
}

public function getTeacherInfo(): ?int
{
    return $this->teacherInfo;
}

public function setTeacherInfo(int $teacherInfo): self
{
    $this->teacherInfo = $teacherInfo;
}

```

```

        return $this;
    }

    public function getMisionData(): ?int
    {
        return $this->misionData;
    }

    public function setMisionData(int $misionData): self
    {
        $this->misionData = $misionData;

        return $this;
    }

    public function addId(Subject $id): self
    {
        if (!$this->Id->contains($id)) {
            $this->Id[] = $id;
        }

        return $this;
    }

    public function removeId(Subject $id): self
    {
        if ($this->Id->contains($id)) {
            $this->Id->removeElement($id);
        }

        return $this;
    }

    public function getCreateActivity(): ?\DateTimeInterface
    {
        return $this->createActivity;
    }

    public function setCreateActivity()
    {
        $this->createActivity = new \DateTime("now", new
            DateTimeZone('America/Santiago'));
    }

    /**
     * @ORM\PrePersist
     * @ORM\PreUpdate
     */
    public function updatedTimestamps()
    {
        $dateTimeNow = new DateTime('dateTime');
        if ($this->getCreateActivity() === null) {
            $this->getCreateActivity($dateTimeNow);
        }
    }
}

```

Anexo B: Manual de usuario.

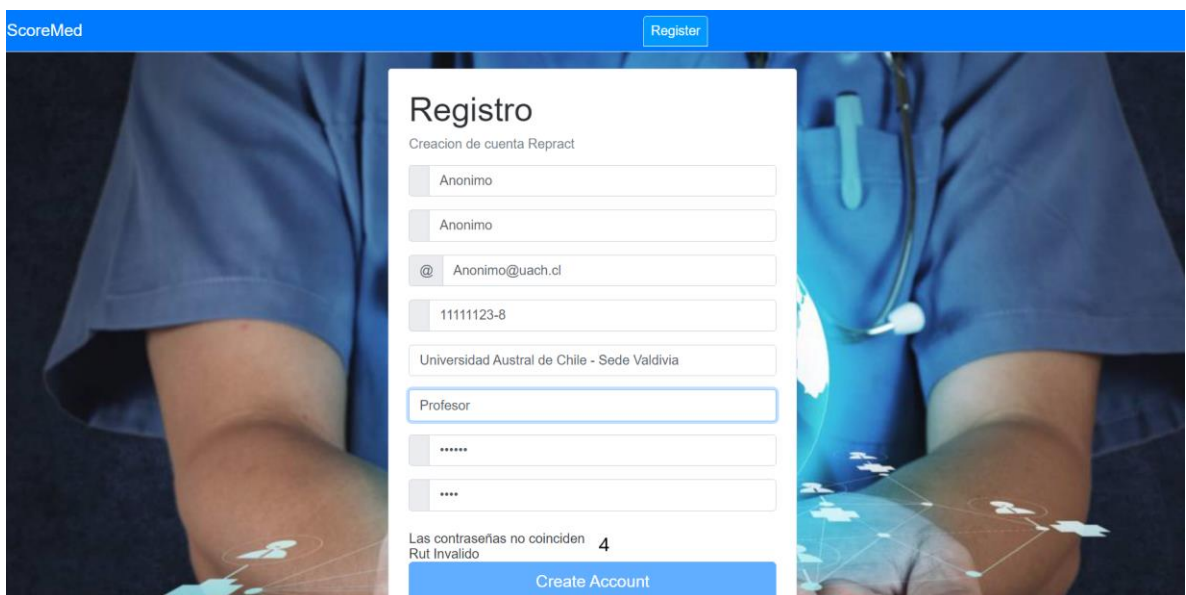
MANUAL DE USUARIO DE SCOREMED

Bienvenido a ScoreMed, este documento es nuestro manual de usuario donde presentaremos nuestra plataforma de manera sencilla para facilitar su uso.

Registro e ingreso



1. En nuestra barra de navegación se observa en donde se puede registrar en nuestra aplicación.
2. Si usted ya está registrado como profesor se le invita a acceder en el punto 2.
3. Al igual que el punto dos, si ya está registrado pero eres estudiante ingresar en el punto 3.



4. En el caso de acceder en la punto 1, se cambiara a la siguiente pestaña donde debe completar con sus datos, y se valida su rut y contraseña.

Uno del Profesor

Si usted es profesor tendrá la siguiente vista luego de iniciar sesión:



5. Si usted ha elegido la sección 2 la barra de navegación sufre una modificación, podrá apreciar que la punto 5 se habilita la sección de “Crear Asignatura” y “Crear Actividad”.
6. Al crear un curso, se explicará posteriormente en el punto 15, podremos ver que se genera una tabla con las asignaturas creadas y vinculadas. En el punto 6 podrá ver los detalles de su curso.
7. En este punto usted podrá editar su curso y así incorporar a profesores clínicos o modificar el curso en caso de ver alguna falla en su creación, como por ejemplo el código o el nombre.
8. En el punto 8 se muestra es la vinculación de una actividad al curso, y en caso de no estar la acción que necesita, se puede crear en el punto 5, y luego volver al punto 8 y vincular esa nueva actividad al curso creado.
9. Si en el punto 9 se presenta un número es porque tiene acciones que validar.
10. Este punto es en caso de querer eliminar la asignatura y borrar todo registro de ella.

ScoreMed Inicio Crear Asignatura Crear actividad Scarlett Letelier Salir

ENFM 111

Proceso de atención de etapas pre y post reproductivas y afecciones médico quirúrgicas de la mujer

Actividades Profesores Alumnos 11

Nombre actividad	Codigo	Cantidad a realizar	Estadísticas
Curacion plana	ACME-0001	4	
Especuloscopia	ACME-0002	12	
Examen Físico de mamas	ACME-0003	8	
Hemoglucotest	ACME-0004	7	
Lavado Clínico de manos	ACME-0005	7	
Puncion IntraMuscular	ACME-0006	80	

11. Al ingresar al punto 6 usted cambiará de vista, donde usted pasará al punto 11 y podrá apreciar las actividades, profesores y los alumnos que están presente en el curso. En la columna “cantidad a realizar” es donde ira el mínimo de la actividad, y en la columna de estadística se podrá apreciar el avance de los estudiantes en dicha actividad. En el botón que está al lado derecho de alumnos se le presentara el resumen del curso en una tabla Estudiantes Vs Actividades.

ScoreMed Inicio Crear Asignatura Crear actividad Scarlett Letelier Salir

Edición de Asignatura

Nombre Asignatura: Proceso de atención de etapas

Abreviación: ENFM 111

☐ Desea editar ? Actualizar Asignatura

☒ Desea Agregar profesores Clínicos ?

Realize búsqueda por nombre o apellido:



















Ingrese informacion...

Nombre	Apellido	Accion

12. En el caso de que acceda en el punto 7 usted podrá editar la asignatura y agregar profesores clínicos.

ScoreMed Inicio Crear Asignatura Crear actividad Scarlett Letelier Salir

Realize busqueda
Se puede buscar por nombre actividad o codigo, si no existe se puede crear
Ingrese informacion...

Actividad	Codigo	Descripcion	Acciones
Curacion plana	ACME-0001	Limpieza de heridas	  
Especuloscopia	ACME-0002	realizar exámenes o procedimientos diagnósticos y terapéuticos de cavidades corporales manteniendo abiertos sus orificios de entrada.	  
Examen Fisico de mamas	ACME-0003	Examen	  
Hemoglucotest	ACME-0004	edir la cantidad de glucosa en la sangre de una persona embarazada	  
Lavado Clínico de manos	ACME-0005	Es la técnica básica utilizada para prevenir la transmisión de infecciones por vía contacto manual, eliminando arrastre los microorganismos que quedan en ellas	  
Puncion IntraMuscular	ACME-0006	es una forma de administración rápida en la que el medicamento es inyectado directamente dentro de un musculo.	  

13. Si usted ingreso al punto 8 desde un curso A, y quiere agregar alguna actividad, solo debe hacer click al botón (+) para realizar la vinculación, además apreciará que podrá editar y eliminar la dicha actividad.

ScoreMed Inicio Crear Asignatura Crear actividad Scarlett Letelier Salir

Realize busqueda
Se puede buscar por nombre del alumno, profesor o actividad
Ingrese informacion...

Nombre	Fecha	Actividad	Profesor Encargado	Estado	Accion
Nicolas Sepulveda	2022-04-13 07:27:25	Curacion plana	Scarlett Letelier	Sin Validar	  
Nicolas Sepulveda	2022-04-13 07:27:34	Curacion plana	Scarlett Letelier	Sin Validar	  
Nicolas Sepulveda	2022-04-13 07:35:40	Curacion plana	Scarlett Letelier	Sin Validar	  
Nicolas Sepulveda	2022-04-13 15:15:01	Curacion plana	Scarlett Letelier	Sin Validar	  

BACK

14. En el caso de que usted acceda en el punto 9 podrá ver las actividades que le faltan validar/rechazar a usted como usuario.

ScoreMed Inicio Crear Asignatura **Crear actividad** Scarlett Letelier Salir

Creacion de nuestra **Actividad**

Nombre Actividad

Codigo Actividad

Descripcion

15. Si usted accedió al punto 5 o algún menú de edición tendrá que llenar o modificar el siguiente formulario no varía mucho si es curso o actividad. En el caso de crear la actividad el campo código actividad es estándar lo genera la plataforma.

Uso del Estudiante

Si usted es estudiante tendrá la siguiente vista luego de iniciar sesión:

ScoreMed Inicio Mis Cursos Record de actividades 16 Nicolas Sepulveda Salir					
Asignatura	Codigo	Descripcion	Profesor	Acciones	
Proceso de atencion de etapas pre y post reproductivas y afecciones medico quirurgicas de la mujer	ENFM 111	Primer Semestre 2020	Scarlett Letelier	17	
Bases Teoricas y Practicas para la atencion de salud de la mujer durante el ciclo reproductivo y neonatal	ENGM 122	Segundo semestre 2020	Scarlett Letelier		

16. Si ingreso como estudiante vera la siguiente página de inicio, en el punto 16 tendrá la barra de navegación “Mis cursos” en esta sección están los cursos a lo que usted se ha inscrito y “Récord de actividades” que es donde encontrara el resumen global de todas las actividades aceptadas y rechazadas.
17. En el punto 17 usted tendrá la opción de inscribirse a una asignatura haciendo click en el curso que fue creado por un profesor.

ScoreMed Inicio Mis Cursos Record de actividades Nicolas Sepulveda Salir					
Asignatura	Codigo	Descripcion	Profesor	Acciones	
Proceso de atencion de etapas pre y post reproductivas y afecciones medico quirurgicas de la mujer	ENFM 111	Primer Semestre 2020	Scarlett Letelier	18 19 20	

18. El punto 18 es similar al punto 6, pero con restricciones.
19. En la siguiente sección usted enviara una actividad a ese curso.
20. Usted podrá apreciar el avance de la asignatura mediante un gráfico de radar.

ScoreMed Inicio Mis Cursos Record de actividades

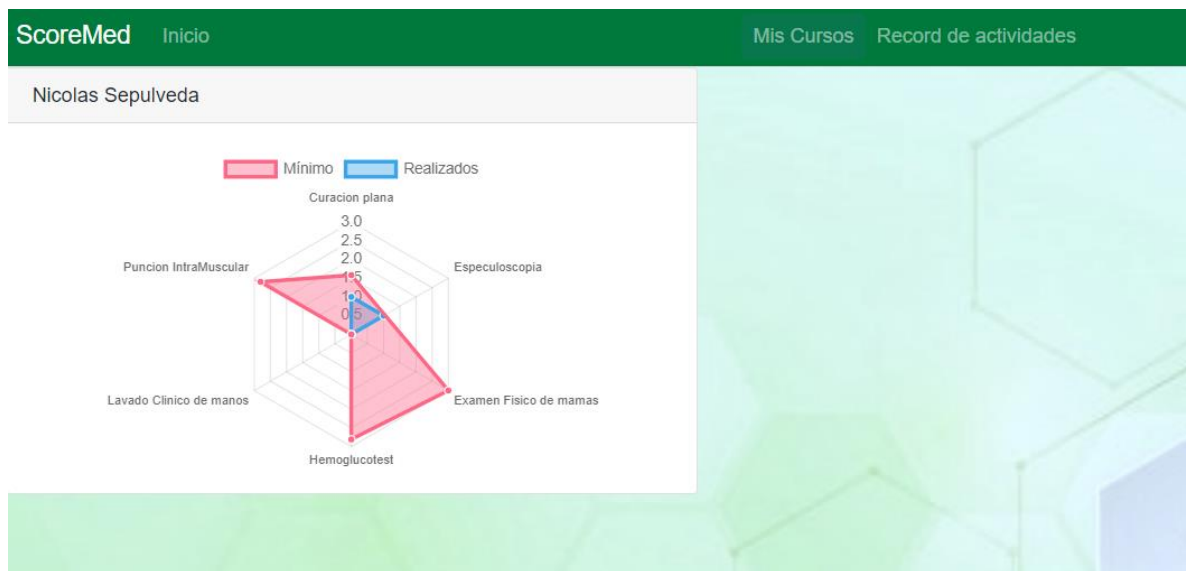
Ingreso de actividad en **La Asignatura**

Seleccione Profesor

Seleccione Actividad que realizo

Enviar Actividad

21. Si usted accede en el punto 19, llegara a la siguiente página donde podrá seleccionar al profesor que le deberá aprobar/rechazar y la actividad que usted como usuario acaba de realizar.



22. Si usted accedió en el punto 20 de la asignatura, podrá apreciar el mínimo por cumplir (rojo) y el otro grafico (azul) se ira generando mientras más actividades realice como estudiante.

ScoreMed Inicio Mis Cursos Record de actividades Nicolas Sepulveda Salir		
Aprobadas Rechazadas		
Aprobadas		
Actividad	Descripcion	Cantidad Realizada
Curacion plana	Limpieza de heridas	2
Especuloscopia	realizar exámenes o procedimientos diagnósticos y terapéuticos de cavidades corporales manteniendo abiertos sus orificios de entrada.	2
Hemoglucotest	edir la cantidad de glucosa en la sangre de una persona embarazada	1
Lavado Clínico de manos	Es la técnica básica utilizada para prevenir la transmisión de infecciones por vía contacto manual, eliminando arrastre los microorganismos que quedan en ellas	1

23. Si en el punto 16 ingreso a “Récord de actividades”, esta será la vista donde podrá apreciar todos sus resultados, en donde saldrán sus actividades Aprobadas y Rechazadas.