



Instituto Politécnico Nacional  
Escuela Superior de Cómputo  
ESCOM



UA: Análisis y diseño orientado a objetos.

Profesor: Rubén Peredo Valderrama

Grupo: 2CM14

*Equipo 4 - Documentación de la aplicación Web  
Graficadora de funciones cuadráticas utilizando  
React, Babel y Webpack.*

Alumnos:

Calderón González Gil Alfonso  
Montaño Estrada Carolina  
Ruiz González Ian Alexander

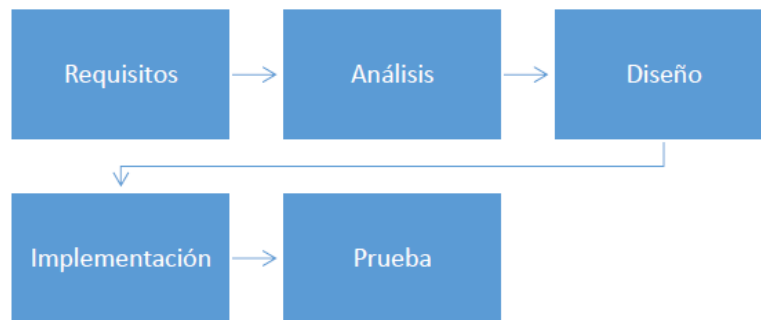
Fecha de entrega: 25 de junio de 2021

## Índice

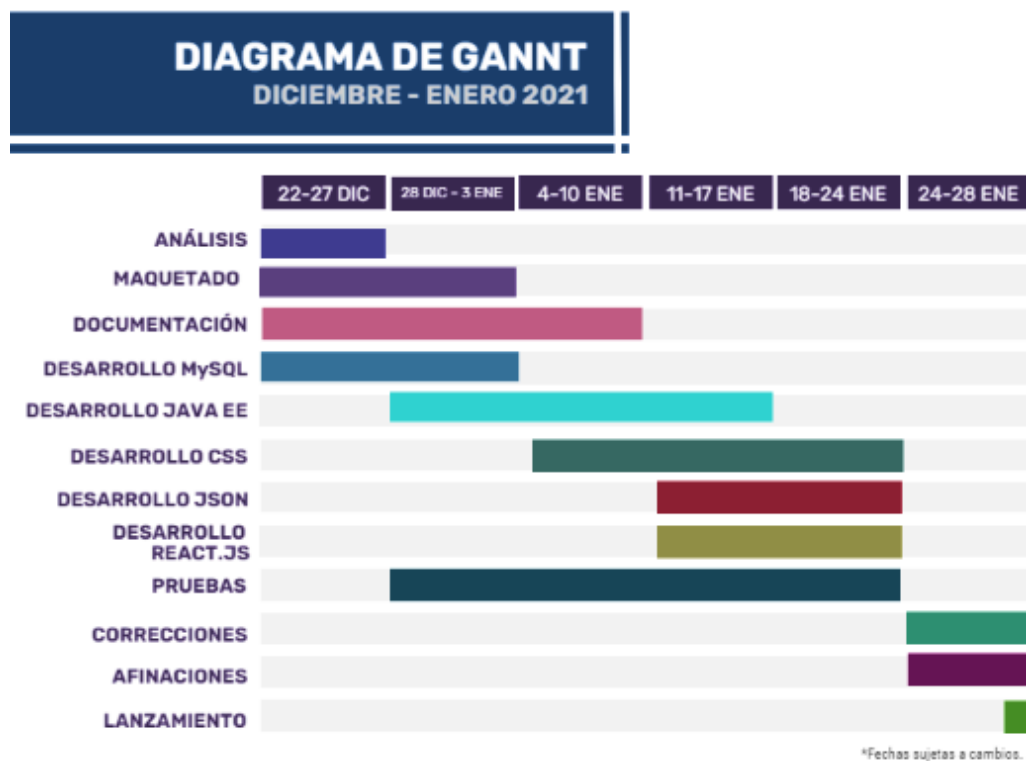
Metodología orientada a objetos seleccionada .....	3
Aplicaciones similares.....	5
Requerimientos .....	6
Requerimientos funcionales.....	6
Requerimientos no funcionales.....	6
Requerimientos de Hardware y Software .....	7
Diagramas .....	8
Diagramas de casos de uso.....	8
Tablas de descripción de casos de uso. ....	9
Diagrama de clases de la base de datos. ....	11
Diagramas de secuencia. ....	11
Diagrama de estados. ....	13
Diagrama de actividades. ....	13
Diagrama de componentes. ....	14
Diagrama de paquetes.....	14
Diagrama de distribución. ....	15
Conclusiones .....	15

## Metodología orientada a objetos seleccionada.

Considerando que nuestro equipo se dio a la tarea de diseñar una aplicación web para ayudar a los estudiantes a visualizar de forma dinámica funciones cuadráticas por medio de una graficadora. Se ha optado por utilizar la metodología **Unified Process (PU)**, dado que dicha metodología trabaja mediante una modelo iterativo e incremental basado en componentes de software interconectados a través de interfaces bien definidas. Dentro de cada iteración capturamos lo que el sistema debía de hacer (Requisitos), estructuramos los requisitos (Análisis) junto a la arquitectura del sistema (Diseño) y construimos el software (Implementación).

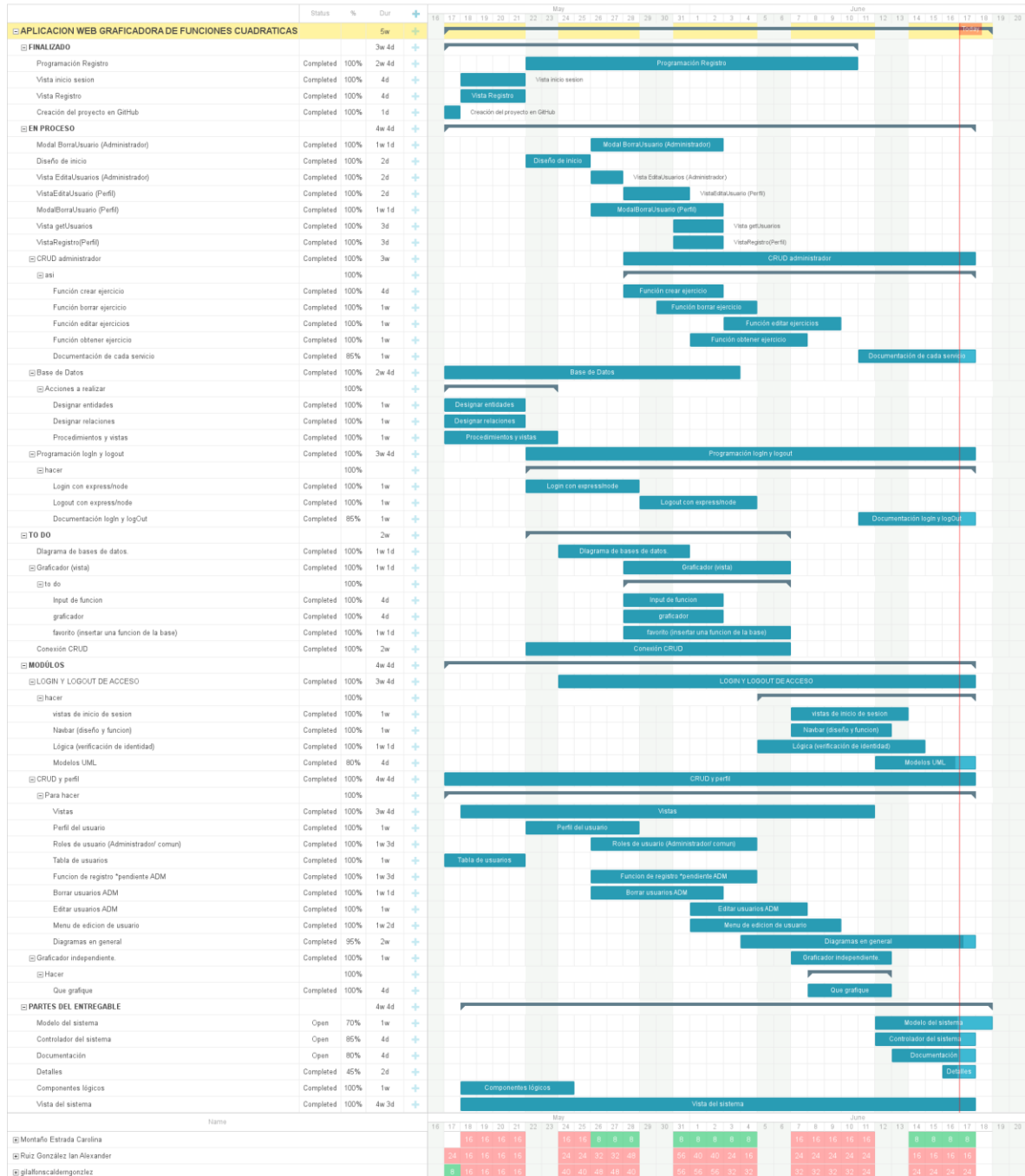


En un caso optimo obtendríamos un diagrama de Gantt de la siguiente forma:



Pero en un caso realista y como fue en nuestra situación mediante la aplicación de **trelló** (que fue la herramienta donde coordinamos nuestro trabajo) obtuvimos un diagrama de Gantt de la siguiente forma.

### Diagrama de Gantt - Equipo 4

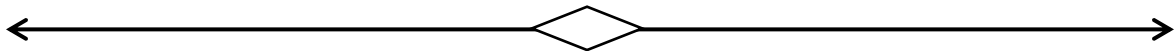


En este diagrama podemos observar como fue nuestro proceso de trabajo en el desarrollo del proyecto en tiempo real y el tiempo empleado en cada una de las funciones y modales desarrollados para la operación de este (Fase de pruebas).

### Aplicaciones similares



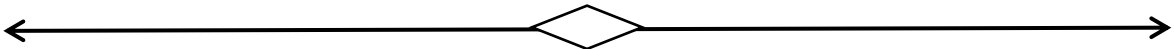
Symbolab es una herramienta avanzada de educación matemática. Permite a los usuarios aprender, practicar y descubrir temas matemáticos utilizando símbolos matemáticos y notaciones científicas, así como texto. Symbolab proporciona soluciones automatizadas paso a paso para los temas algebraicos, trigonométricos y de cálculo que abarcan desde la escuela intermedia hasta la universidad.



### OnlineMSchool

OnlineMSchool es un asistente para los alumnos del colegio, estudiantes y todos que tienen ganas para aprender matemáticas y saber cómo resolver problemas.

*OnlineMSchool*



### Analyzemath



Analyzemath son calculadoras en línea y la geometría para resolver varios temas de geometría, como calcular el área, volumen, distancia, puntos de intersección. Estos pueden ser utilizados para comprobar la tarea respuestas, la práctica o de explorar con distintos valores para la comprensión profunda.

## Requerimientos

Desarrollar una aplicación Web que permita crear ejercicios dinámicos de funciones cuadráticas utilizando react, babel y webpack. La página deberá soportar multimedia como: **textos, imágenes, sonidos y vídeos**. Los ejercicios deberán de contar con un módulo de subida, edición y de borrado de este a la hora de que el usuario inicie su sesión; para ello se requiere ingresar una serie de datos solicitados por el sistema que permitan hacer dichos cálculos. Así mismo, se requiere que el usuario pre-registrado inicie sesión para hacer uso de las funciones de la página. El usuario se encontrará pre-registrado ya que en la rubrica lo solicita de esta forma, contará con un correo y contraseña para poder acceder después de desplazarse por la pagina principal para iniciar sesión.

### Requerimientos funcionales

- El sistema notificará cuando se realicen correctamente las siguientes transacciones:
  - Inicio de sesión del usuario.
  - Registro de ejercicio.
  - Actualización de ejercicio.
  - Eliminación de ejercicio.
- El sistema deberá de validar que los datos que el usuario registra cumplan las siguientes consideraciones:
  - Validar los datos de inicio de sesión.
  - La contraseña del usuario deberá contener mínimo 8 caracteres.
  - El correo tiene un máximo de 30 caracteres.
- El sistema controlará el acceso y lo permitirá solamente a usuarios pre-registrados. Los usuarios deben entrar con su correo y contraseña.
- El sistema validará la ecuación insertada y mostrará el resultado en pantalla.
- Al ingresar ejercicios, todo ejercicio estará asociado a un tema y un subtema. A su vez, todo subtema se encuentra asociado a un tema.
- El administrador podrá agregar, visualizar, editar o eliminar los ejercicios que haya creado.
- El administrador puede consultar los ejercicios que han hecho los usuarios desde la base de datos.
- Al guardar, editar o eliminar un ejercicio, los datos serán únicamente afectados para el usuario en turno sin afectar el de los demás.
- El alumno podrá visualizar los ejercicios guardados con anterioridad mediante el botón de favorito.

### Requerimientos no funcionales

- El sistema contará con una interfaz intuitiva.
- Los contenidos del sitio sólo podrán ser modificados por el administrador.

## Requerimientos de Hardware y Software

Se desglosan diferentes requerimientos que dependerán del uso requerido de la aplicación web, ya sea a nivel desarrollador o a nivel usuario donde:

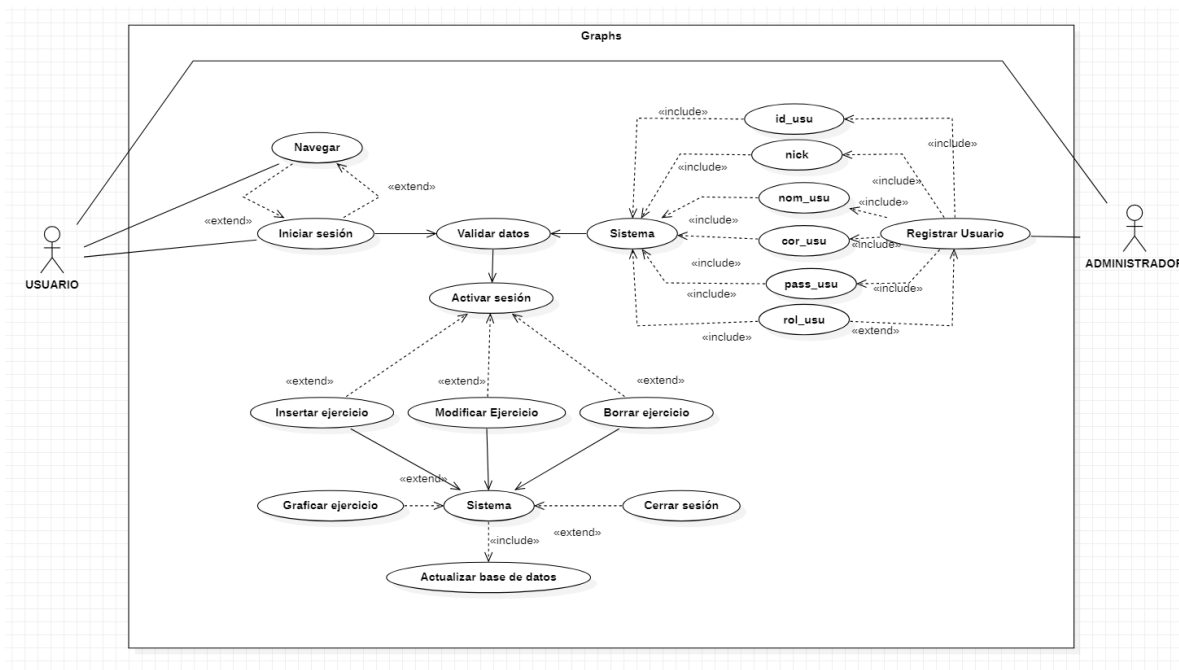
DESARROLLADOR	USUARIO
<ul style="list-style-type: none"><li>● CSS</li><li>● HTML5</li><li>● GitBash</li><li>● Postman</li><li>● Javascript</li><li>● JSON</li><li>● MySQL</li><li>● React.js</li><li>● Webpack</li><li>● Babel</li><li>● Node.js</li><li>● StarUML</li><li>● Discord</li><li>● Contar con un equipo de cómputo con un procesador core i5, ryzen 5 o superior con al menos 8GB de RAM.</li></ul>	<ul style="list-style-type: none"><li>● Conexión a internet</li><li>● Navegador web que soporte HTML5, CSS y Flash</li><li>● 2 GB de RAM mínimo</li></ul>

## Diagramas

En este apartado mostramos los diagramas más significativos (para nosotros) de la aplicación realizada.

### Diagramas de casos de uso.

Este diagrama se puede separar en 2 partes, la parte del administrador que es un actor pasivo ya que tiene que dar de alta al usuario en la base de datos y el usuario que tiene que iniciar sesión para poder trabajar con los ejercicios.



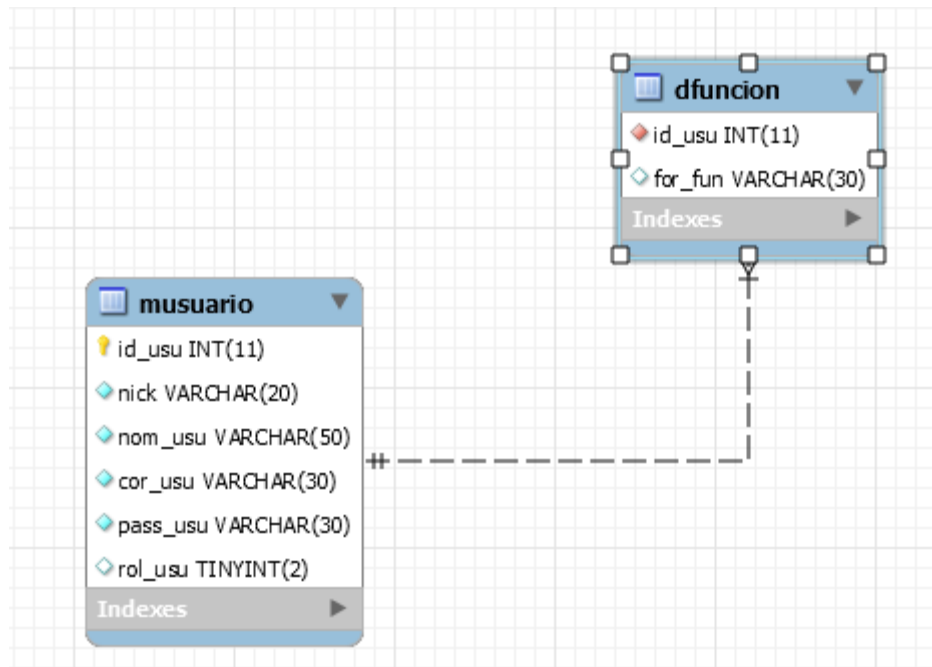


## Tablas de descripción de casos de uso.

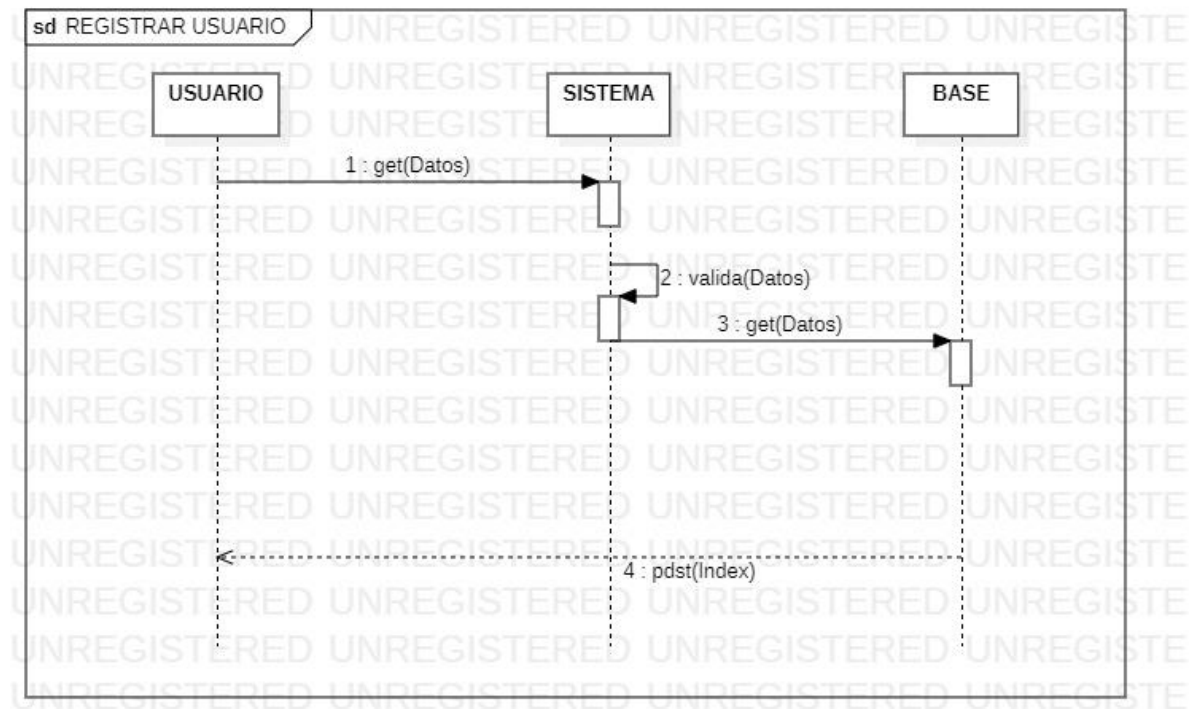
Nombre	Registrar Usuario
Descripción corta:	El usuario debe de ponerse en contacto con el administrador para que lo de alta en el sistema y así pueda entrar a la aplicación de graficas donde podrá crear nuevos ejercicios, editarlos o borrarlos de su cuenta según sea el caso.
Precondición:	Se requiere un usuario interesado.
Postcondición:	El usuario esta dado de alta en el sistema y puede empezar a trabajar en el graficador.
Situaciones de error:	El usuario no recuerda sus datos. El sistema no esta dado de alta y el registro no se completa. La base de datos esta corrupta y se tiene que reponer.
Estado del Sistema en caso de error:	Informa mediante terminal un error en la conexión o actualización del servidor y donde fue dicho error.
Actores	Administrador
Disparadores	Trabajar en la aplicación
Proceso estándar	<ol style="list-style-type: none"> <li>1. El usuario se pone en contacto con el administrador para crear una cuenta.</li> <li>2. El usuario deberá llenar los campos de un Nick, nombre, correo y una contraseña para darlo de alta en el sistema y por default se le asignará un rol cero.</li> <li>3. Se verifica que el usuario se encuentre en la base de datos.</li> <li>4. El usuario puede ingresar en la aplicación.</li> </ol>
Proceso alternativo	<ol style="list-style-type: none"> <li>1. El usuario no contacta al administrador.</li> <li>2. El usuario podría olvidar sus credenciales.</li> <li>3. El usuario captura mal sus datos.</li> <li>4. El sistema no logra registrarlo.</li> </ol>

Nombre	Iniciar sesión
Descripción corta:	El usuario debe ingresar sus credenciales para ingresar al sistema.
Precondición:	El usuario que desea ingresar al sistema debe estar registrado.
Postcondición:	El usuario ingresó con éxito al sistema
Situaciones de error:	El usuario no ingresa el alias o contraseña correctos El usuario no se encuentra registrado
Estado del Sistema en caso de error:	El sistema no le da el acceso al usuario. Mensaje de alerta
Actores	Usuario
Disparadores	Los datos del usuario se deben encontrar registrados en la base de datos
Proceso estándar	<ol style="list-style-type: none"> <li>1. El usuario ingresa a la plataforma.</li> <li>2. El usuario selecciona "Iniciar Sesión".</li> <li>3. El usuario captura sus datos (alias y contraseña).</li> <li>4. El sistema verifica que todos los campos sean completados.</li> <li>5. El sistema consulta con la base de datos si existe un registro con los datos que el usuario ingresó.</li> <li>6. El usuario ingresa al sistema y este le muestra la pantalla principal</li> <li>7. El usuario tiene las opciones de crear, guardar o modificar un ejercicio ya existente.</li> <li>8. El usuario puede visualizar la función o borrarla a la hora de graficarlas.</li> <li>9. El usuario puede visualizar múltiples graficas.</li> </ol>
Proceso alternativo	<ol style="list-style-type: none"> <li>1. El usuario selecciona "Iniciar Sesión".</li> <li>2. El usuario captura sus datos (alias y contraseña).</li> <li>3. El sistema verifica que todos los campos sean completados.</li> <li>4. El sistema consulta con la base de datos si existe un registro con los datos que el usuario ingresó.</li> <li>5. Se notifica al usuario para que este realice las modificaciones correspondientes.</li> <li>6. El usuario captura sus datos (alias y contraseña).</li> <li>7. En caso de que los campos coincidan con algún registro dentro de la base, el sistema le da acceso al usuario y le muestra la página principal; de lo contrario regresa al paso 5.</li> </ol>

Diagrama de clases de la base de datos.



Diagramas de secuencia.



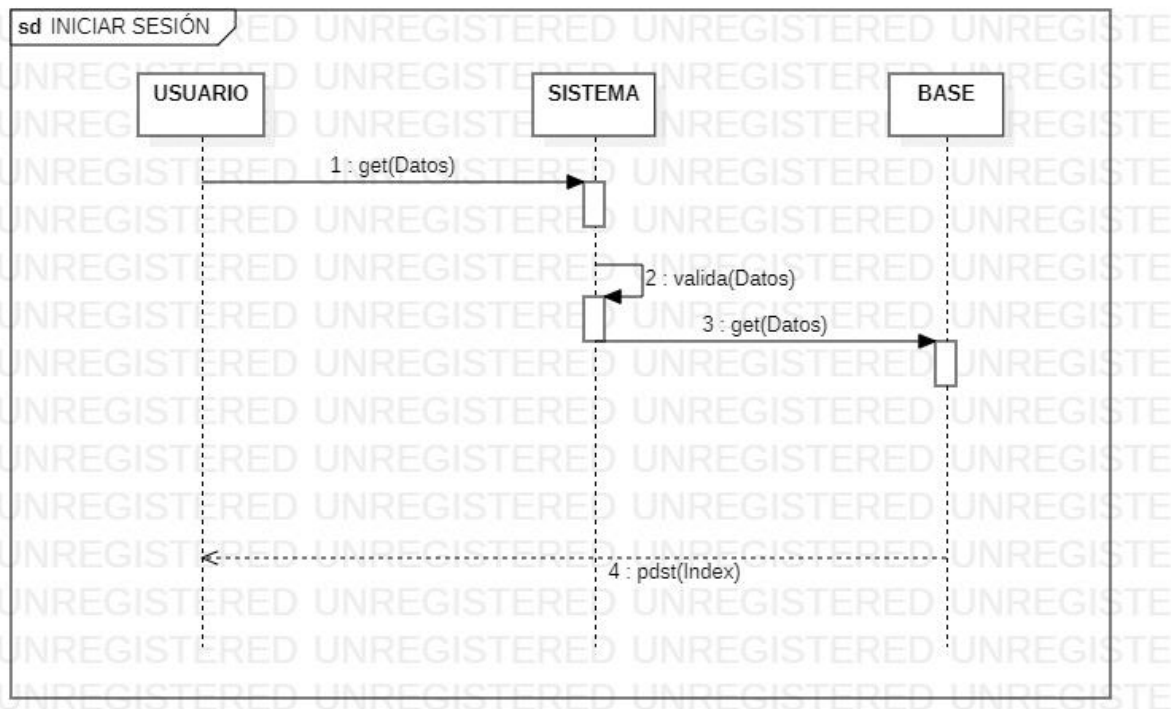
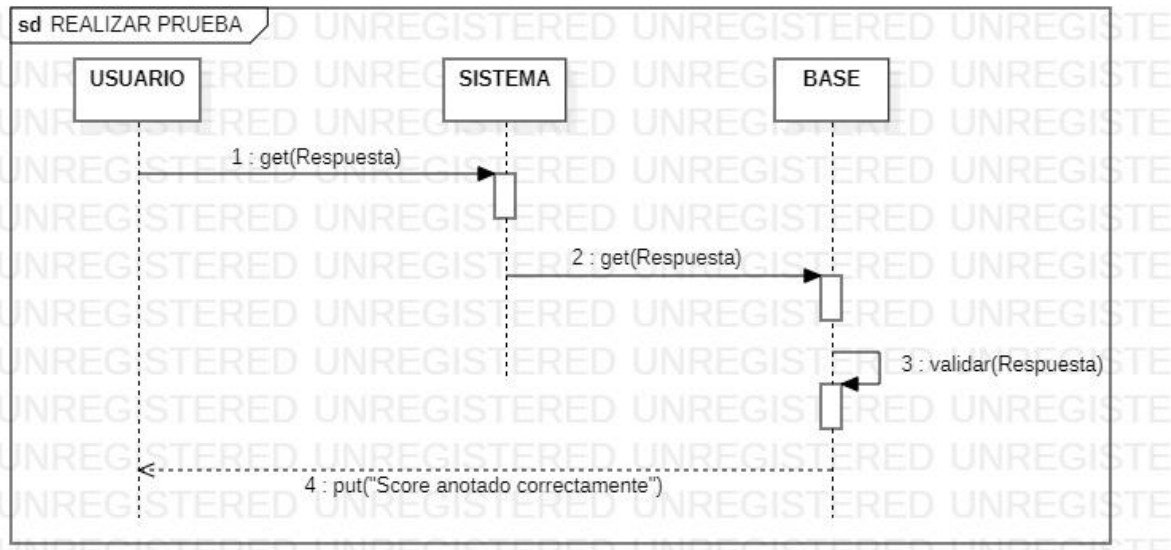


Diagrama de estados.

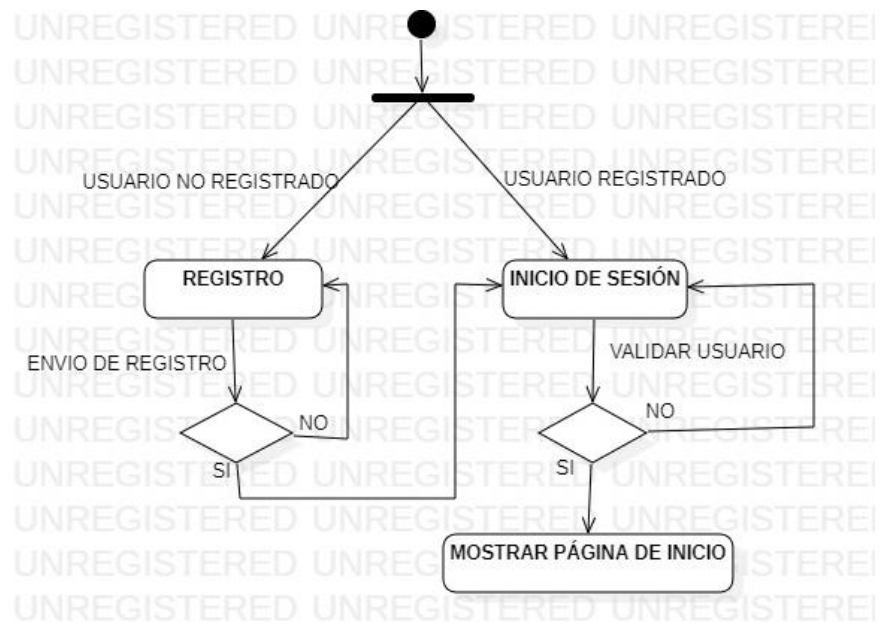


Diagrama de actividades.

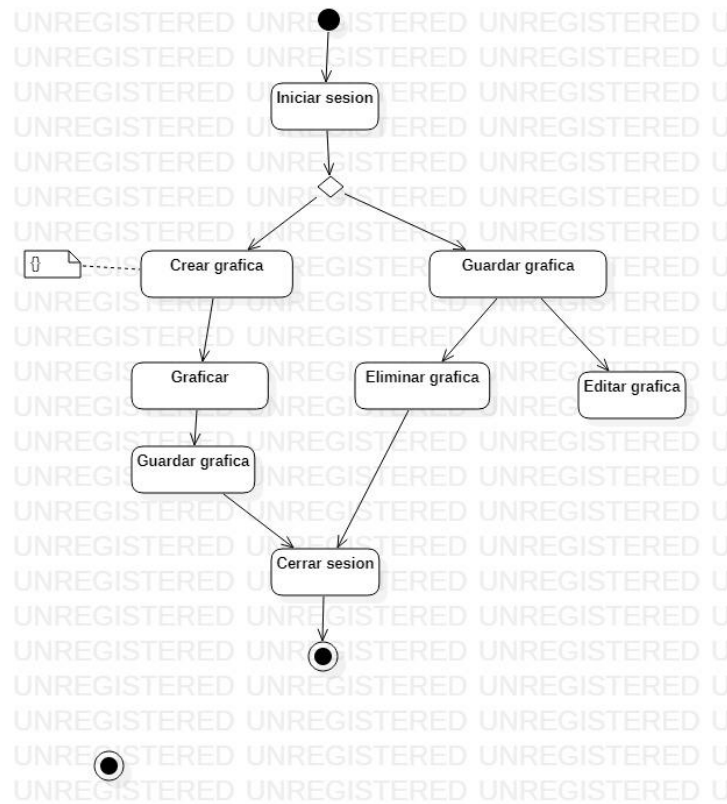


Diagrama de componentes.

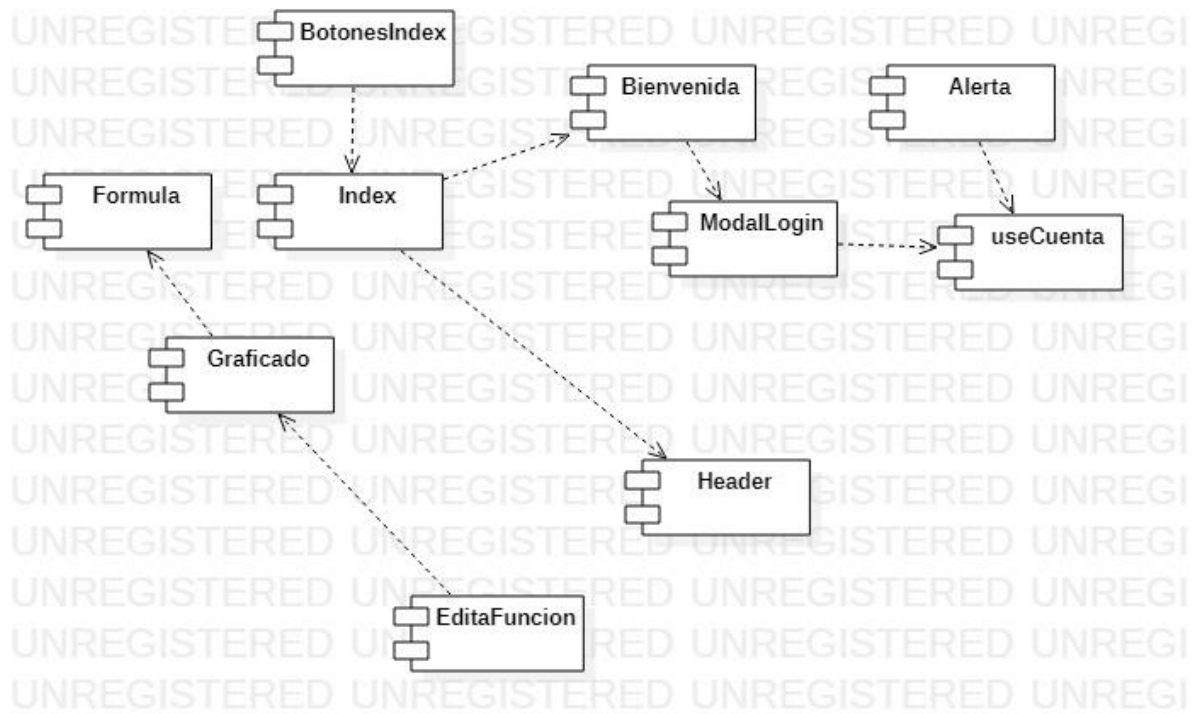
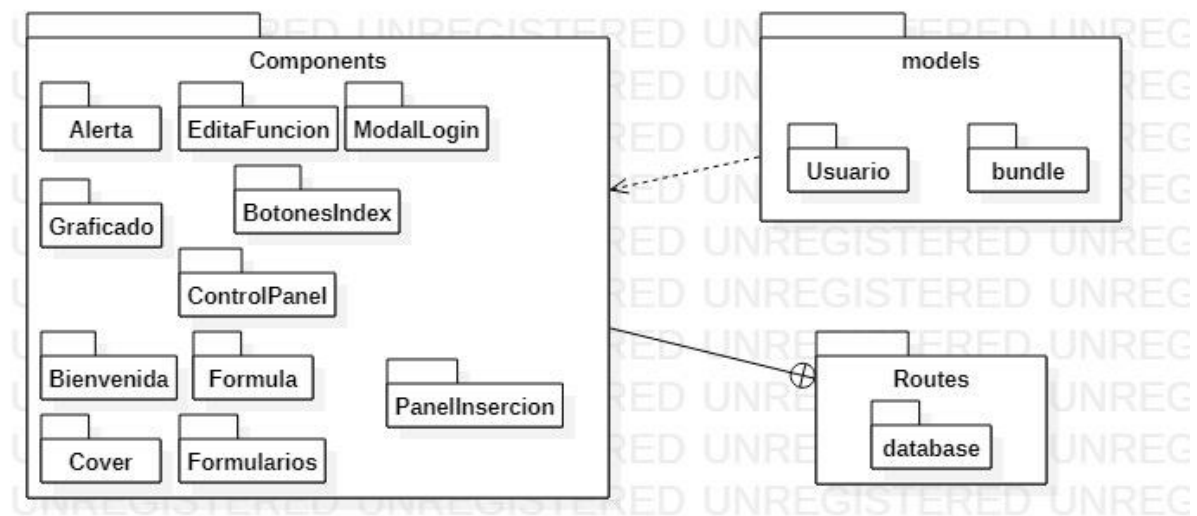
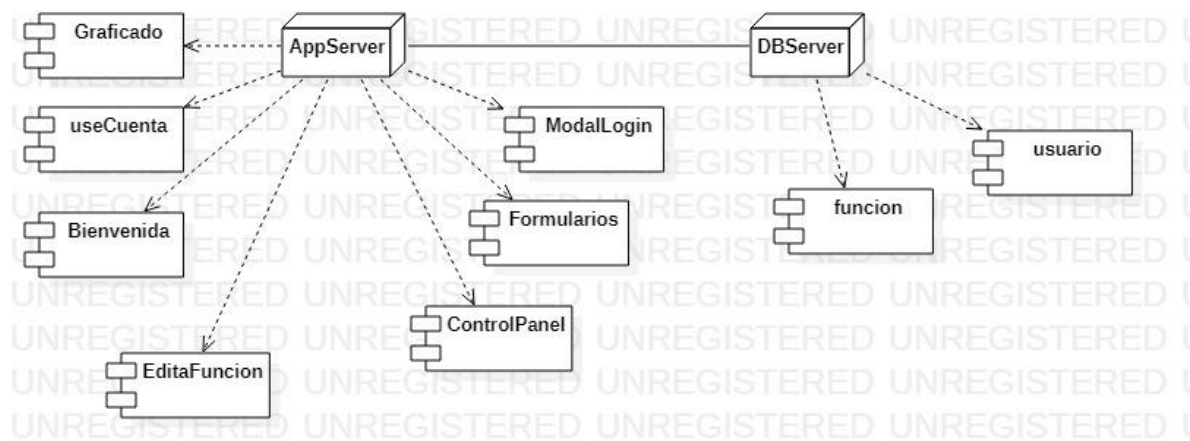


Diagrama de paquetes.



## Diagrama de distribución.



## Conclusiones

A lo largo de este proyecto desde la elección del tema, hasta la implementación y desarrollo de este se aplicaron los conocimientos aprendidos a lo largo de todo el semestre desde las bases hasta los tópicos más complejos como lo son la programación con React.js y el correcto modelaje de los diagramas en el programa StarUML. Tuvimos algunas dificultades con el proyecto y algunas externas lo cual nos impidió presentar el proyecto en tiempo y forma, pero por suerte fuimos capaces de resolver estas dificultades gracias a foros y la documentación de varias de estas dependencias.

Al terminar este proyecto cada integrante del equipo logro aportar sus conocimientos y habilidades a este, de forma que si alguno no lograba entender alguna parte siempre se podría apoyar en el resto de los miembros del equipo y reafirmando los conocimientos para poder resolver futuros problemas similares ya sea en alguna materia de la carrera o en el ámbito laboral