FUNDAMENTOS DE PROGRAMACIÓN

Grupo 8

Scrum Master

Diego Fabián Sánchez Villamil

Development Team

Giezi A. Diaz. Morales

Roymel Torres

Michael Cortes

Jhon Walter Riveros

Gustavo Jaime González Cortezano

Contenido

[**Sprint Planning** 3](#_Toc83063874)

[**Descripción de proyecto.** 4](#_Toc83063875)

[**Sprint 1** 5](#_Toc83063876)

[**Diagrama de casos de uso** 6](#_Toc83063877)

[**Diagrama Navegacional** 10](#_Toc83063878)

# **Sprint Planning**

**Fecha: 13/09/2021**

**Horario: 7:00 pm a 9:00 pm**

**Recursos:** Enlace

**Participantes:**

1. Scrum Master

* Diego Fabián Sánchez Villamil

1. Development Team

* Giezi A. Diaz. Morales
* Roymel Torres
* Michael Cortes
* Jhon Walter Riveros
* Gustavo Jaime González Cortezano

# **Descripción de proyecto.**

El gimnasio La arena del sudor Solicita la creación de un aplicativo el cual atienda los siguientes requerimientos.

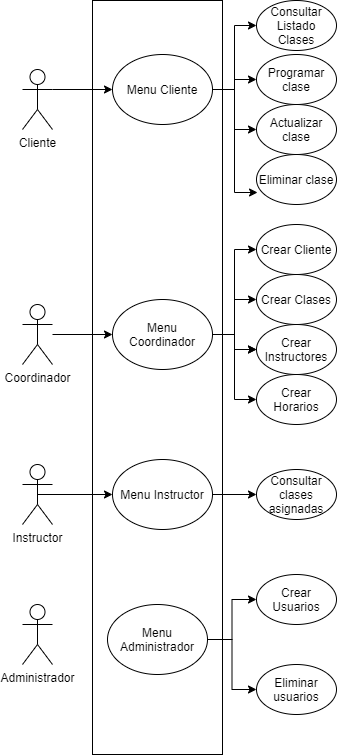
* El usuario solicita que el cliente pueda ingresar con un usuario y contraseña para programar clase, actualizar clase, eliminar clase y consultar listado de clases.
* El usuario solicita una cuenta coordinador donde se puedan crear clases, cliente, instructores y horarios.
* El usuario solicita que el instructor pueda ingresar y visualizar las clases asignadas.
* El usuario solicita la creación de una cuenta administradora para crear y eliminar usuarios.

# **Sprint 1**

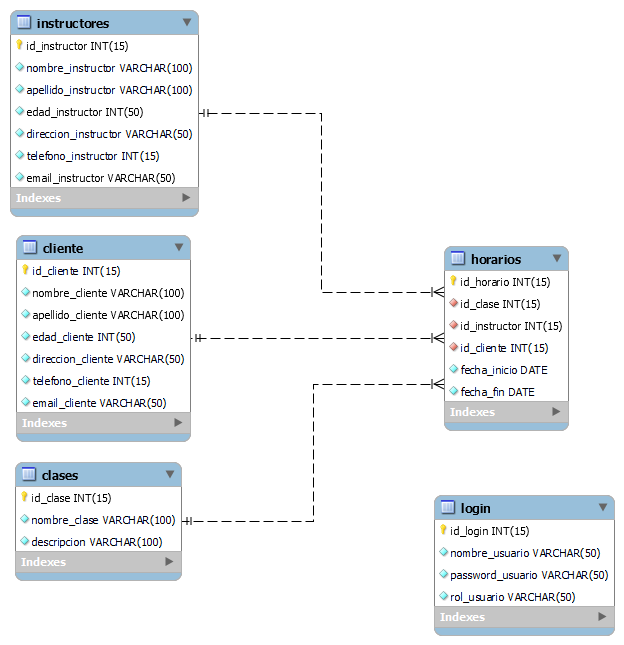
Para el Desarrollo del proyecto se realizará en 5 ***Sprint***. El primer ***Sprint*** que tiene una duración de 1 semana se desarrollará de la siguiente manera:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | ***No. Historias de Usuario*** | ***Nombre de la Historia de Usuario*** | ***Como <tipo de usuario>*** | ***Quiero <realizar alguna tarea>*** | ***Para <completar mis tareas a tiempo>*** | ***Criterios de Aceptación*** |
| Sprint 1 | GAS-01 | ***Diagrama de casos de uso*** | Giezi Diaz | Desarrollar las relaciones entre los componentes del sistema | Debe tener representada la lógica interna del sistema | * Documentación de los componentes * Se debe especificar las relaciones entre componentes * Un componente puede tener más de una relación |
| GAS -02 | ***Modelo de datos*** | Roymel Torres | Modelar la estructura de la base de datos del sistema | Debe tener organizada la información, haciendo optima la asignación y la petición de datos | * Modelo Entidad relación y documentación * Cada tabla debe tener su respectivo identificador, el cual debe ser único. |
| GAS -03 | ***Implementación de Base de Datos*** | Michael Cortes  Jhon Riveros | Realizar la sincronización del modelo de la base de datos con el sistema | Debe complementar el correcto funcionamiento del sistema | * Documentación de la BD * La base de datos tiene que estar montada en algún gestor de bases de datos |
| GAS -04 | ***Diagrama Navegacional*** | Gustavo Gonzales | Diseñar la estructura de navegación entre interfaces del sistema | Debe tener representada la lógica navegacional del sistema | * Se debe especificar qué acciones se pueden ejecutar en cada interfaz * Se mostrarán atributos de las interfaces (cual será el home, y cual necesita autentificación) * Se debe presentar de manera jerárquica. |

# **Diagrama de casos de uso**



**Diagrama entidad relación**

****

**Implementación de Base de Datos**

**El código creado al exportar la BD de MySQL utilizando HeidiSQL de Laragón es**

**DROP** **DATABASE** **if** **EXISTS** GYM;

**CREATE** **DATABASE** **IF** **NOT** **EXISTS** GYM;

**USE** GYM;

**CREATE** **TABLE** **if** **NOT** **EXISTS** clases(

id\_clase **INT** (15) **NOT** **NULL** **AUTO\_INCREMENT**,

nombre\_clase **VARCHAR** (100) **NOT** **NULL**,

descripcion **VARCHAR** (100) **NOT** **NULL**,

**CONSTRAINT** ic\_pk **PRIMARY** **KEY** (id\_clase));

**CREATE** **TABLE** **IF** **NOT** **EXISTS** instructores(

id\_instructor **INT** (15) **NOT** **NULL** **AUTO\_INCREMENT**,

nombre\_instructor **VARCHAR** (100) **NOT** **NULL**,

apellido\_instructor **VARCHAR** (100) **NOT** **NULL**,

edad\_instructor **INT** (50) **NOT** **NULL**,

direccion\_instructor **VARCHAR** (50) **NOT** **NULL**,

telefono\_instructor **INT** (15) **NOT** **NULL**,

email\_instructor **VARCHAR** (50) **NOT** **NULL**,

**CONSTRAINT** ii\_pk **PRIMARY** **KEY** (id\_instructor));

**CREATE** **TABLE** **if** **NOT** **EXISTS** cliente(

id\_cliente **INT** (15) **NOT** **NULL** **AUTO\_INCREMENT**,

nombre\_cliente **VARCHAR** (100) **NOT** **NULL**,

apellido\_cliente **VARCHAR** (100) **NOT** **NULL**,

edad\_cliente **INT** (50) **NOT** **NULL**,

direccion\_cliente **VARCHAR** (50) **NOT** **NULL**,

telefono\_cliente **INT** (15) **NOT** **NULL**,

email\_cliente **VARCHAR** (50) **NOT** **NULL**,

**CONSTRAINT** ia\_pk **PRIMARY** **KEY** (id\_cliente));

**CREATE** **TABLE** **if** **NOT** **EXISTS** horarios(

id\_horario **INT** (15) **NOT** **NULL** **AUTO\_INCREMENT**,

id\_clase **INT** (15) **NOT** **NULL**,

id\_instructor **INT** (15) **NOT** **NULL**,

id\_cliente **INT** (15) **NOT** **NULL**,

fecha\_inicio **DATE** **NOT** **NULL**,

fecha\_fin **DATE** **NOT** **NULL**,

**CONSTRAINT** ih\_pk **PRIMARY** **KEY** (id\_horario),

**CONSTRAINT** ho\_icl\_fk **FOREIGN** **KEY** (id\_clase) **REFERENCES** clases (id\_clase),

**CONSTRAINT** ho\_ii\_fk **FOREIGN** **KEY** (id\_instructor) **REFERENCES** instructores (id\_instructor),

**CONSTRAINT** ho\_ic\_fk **FOREIGN** **KEY** (id\_cliente) **REFERENCES** cliente (id\_cliente));

**CREATE** **TABLE** **IF** **NOT** **EXISTS** login(

id\_login **INT** (15) **NOT** **NULL** **AUTO\_INCREMENT**,

nombre\_usuario **VARCHAR** (50) **NOT** **NULL**,

password\_usuario **VARCHAR** (50) **NOT** **NULL**,

rol\_usuario **VARCHAR** (50) **NOT** **NULL**,

**CONSTRAINT** lo\_id\_pk **PRIMARY** **KEY** (id\_login));

# **Diagrama Navegacional**

