Development Team: Jhendy Johana Cordero Padilla – Christian Camilo Campos Castro – Omar Alejandro Coronado Tolosa Scrum master: Cindy Nathalia Medina Suarez –

GRUPO 7 DEVELOPMENT TEAM | LA CALIFORNIA

G7\_Sprint 1

c3-g43-desarrollo de software t.martha yolanda diaz

# **Sprint Planning**

# **Fecha:** 15 de septiembre de 2021

# **Horario**: 8 pm

# **Recursos:** Se utiliza el link [https://teams.microsoft.com/l/meetup-join/19%3ameeting\_NzJjOGUxMDQtYjQzZC00ZjRiLTlmZDUtZjBjMTU5ODZiMGQw%40thread.v2/0?context=%7b%22Tid%22%3a%2232c67a4e-23f3-41ad-b463-a807bd1f2c9b%22%2c%22Oid%22%3a%223735fa53-f7e0-47e2-a53c-1571631c0c3a%22%7d](https://teams.microsoft.com/l/meetup-join/19%3Ameeting_NzJjOGUxMDQtYjQzZC00ZjRiLTlmZDUtZjBjMTU5ODZiMGQw@thread.v2/0?context=%7b%22Tid%22%3A%2232c67a4e-23f3-41ad-b463-a807bd1f2c9b%22%2C%22Oid%22%3A%223735fa53-f7e0-47e2-a53c-1571631c0c3a%22%7d) vía teams.

# **Participantes:**

# **Development Team**: Jhendy Johana Cordero Padilla – Christian Camilo Campos Castro

# **Scrum máster**: Cindy Nathalia Medina Suarez

**Descripción de proyecto Gimnasio Hard Body .**

El gimnasio Hard body necesita una página web que tiene como finalidad apoyar los procesos de entrenamiento a los afiliados, para ello necesita implementar un sistema de información que tiene como objetivo fundamental automatizar el proceso de inscripciones de entrenadores y afiliados a un sistema de control de acondicionamiento, donde a cada afiliado se le asigna un único entrenador y el entrenador puede apoyar los procesos de acondicionamiento físico de muchos afiliados. El Recepcionista necesita tener una aplicación informática para gestionar la información de cada afiliado, de igual forma, poder ver los datos de un afiliado en particular.

De cada afiliado requiere la siguiente información:

Nombres, apellidos, dirección, teléfono, barrio donde reside y año en que se matricula en el Hard Body para recibir los servicios.

De cada entrenador requiere la siguiente información:

Nombres, apellidos, especialidad, año de nacimiento, teléfono y barrio donde reside.

Se necesita gestionar los afiliados, donde se pueda: agregar sus datos personales a través de un formulario. También se pueda modificar, cambiar el estado del afiliado, ver la información de todos los afiliados matriculados y ver la información de un afiliado en particular.

Se requiere gestionar los entrenadores, donde se pueda agregar sus datos personales a través de un formulario. También se pueda modificar, cambiar el estado del entrenador, ver la información de todos los entrenadores que laboran en el Hard Body y ver la información de un entrenador en particular.

El administrador requiere ver los reportes de todos los afiliados y entrenadores que tiene el Hard body, y descargar estos reportes para su gestión.

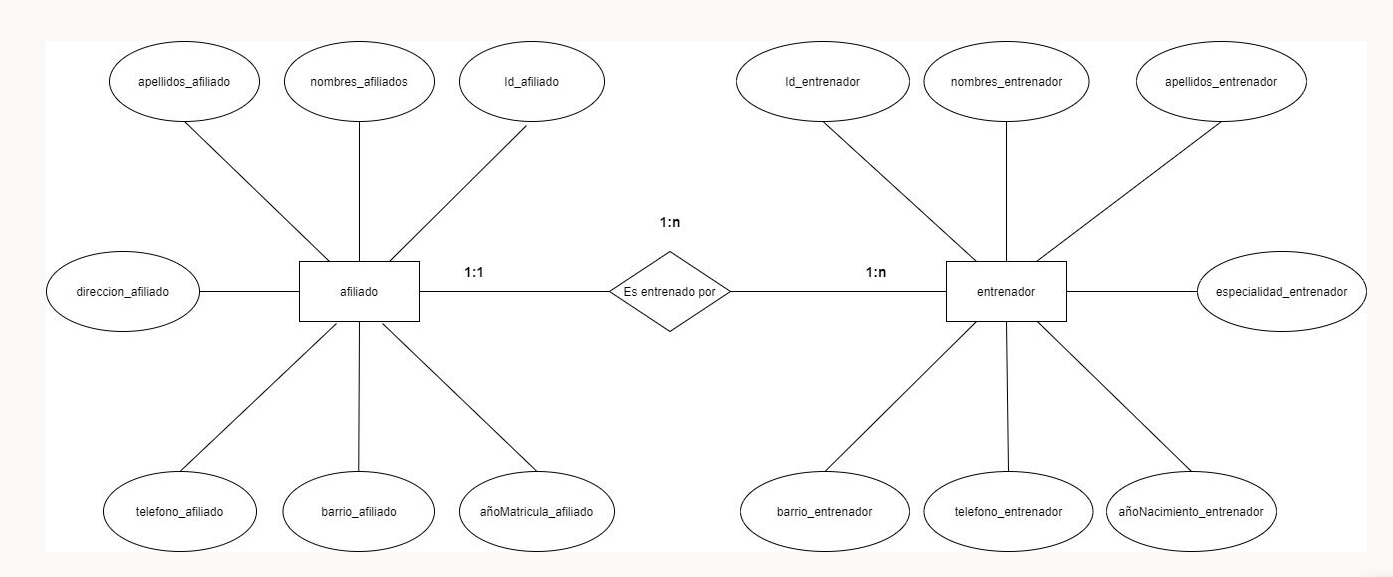
Para entrar al sistema, se requiere que exista una validación de los usuarios (recepcionista y administrador) de tal forma que les permita manipular ciertas funcionalidades del sistema. En el caso del recepcionista debe permitirle agregar, modificar, visualizar el informe general o particular, la información tanto de los afiliados, como de los entrenadores.

Para el caso del administrador, le permita ver e imprimir los reportes de los afiliados y entrenadores.

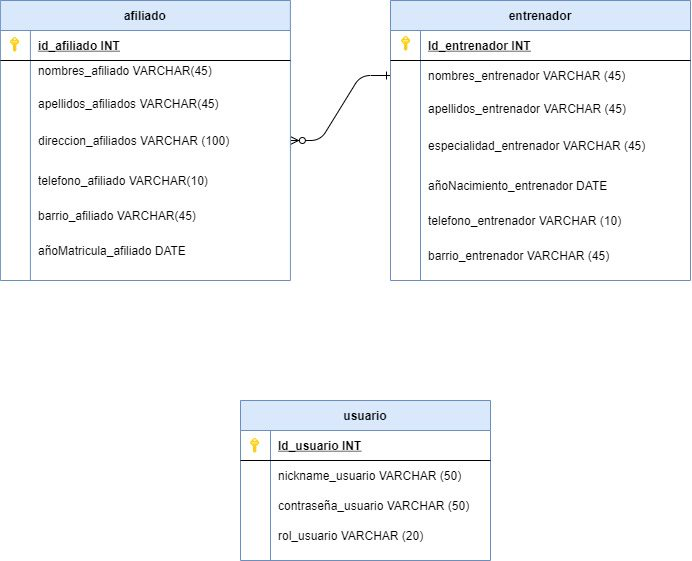
Para el desarrollo del diagrama de Entidad – Relación y diagrama Relacional se usó la herramienta Draw.io. Los archivos .jpg y .drawio se almacenaron en GitHub y el siguiente enlace corresponde a la ruta de acceso:

<https://github.com/Natamed85/ProyectoG43/tree/main/Modelo%20ER%20y%20Relacional>

**Diagrama Entidad – Relación**



**Diagrama Relacional**



--comprueba si existe una base de datos llamada hardbody o en su defecto la crea

--/\*!40100 es la version de Mysql

--latin1 \*/; conjunto de caracteres según especificacion ISO 8859-1 West European

CREATE DATABASE IF NOT EXISTS `hardbody` /\*!40100 DEFAULT CHARACTER SET latin1 \*/;

USE `hardbody`;

-- se crea la tabla afiliado y se da la estructura los datos a solicitar

CREATE TABLE afiliado (

`Numero` int null auto\_increment,

`id\_afiliado` int(11) NOT NULL,

`nombres\_afiliado` varchar(45) DEFAULT NULL,

`apellidos\_afiliado` varchar(45) DEFAULT NULL,

`barrio\_afiliado` varchar(45) DEFAULT NULL,

`direccion\_afiliado` varchar(100) DEFAULT NULL,

`telefono\_afiliado` varchar(10) DEFAULT NULL,

`anomatricula\_afiliado` date DEFAULT NULL,

`entrenador\_id\_entrenador` int(11) NOT NULL,

`nick\_afiliado` varchar(13) NOT NULL,

`password` varchar(15) NOT NULL,

PRIMARY KEY (`id\_afiliado`),

INDEX `fk\_afiliado\_entrenador\_idx` (`entrenador\_id\_entrenador`), -- indice para mejorar la busqueda de datos

CONSTRAINT `fk\_afiliado\_entrenador` FOREIGN KEY (`entrenador\_id\_entrenador`) REFERENCES `entrenador` (`id\_entrenador`); -- crea una clave externa y una clave foranea para mantener la coherencia de los datos relacionados

-- La exportación de datos fue deseleccionada.

-- Volcando estructura para tabla hardbody.entrenador

CREATE TABLE IF NOT EXISTS `entrenador` (

`id\_entrenador` int(11) NOT NULL,

`nombres\_entrenador` varchar(45) DEFAULT NULL,

`apellidos\_entrenador` varchar(45) DEFAULT NULL,

`titulo\_entrenador` varchar(150) DEFAULT NULL,

`anonacimiento\_entrenador` date DEFAULT NULL,

`telefono\_entrenador` varchar(10) DEFAULT NULL,

`barrio\_entrenador` varchar(45) DEFAULT NULL,

PRIMARY KEY (`id\_entrenador`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1; --motor de la base de datos a crear InniDBclaves es especial para claves foráneas y bloqueos

-- crea una tabla usuario si no existe

CREATE TABLE IF NOT EXISTS `usuario` (

`id\_usuario` int(11) NOT NULL,

`nick\_usuario` varchar(50) DEFAULT NULL,

`password\_usuario` varchar(50) DEFAULT NULL,

`rol\_usuario` varchar(20) DEFAULT NULL,

PRIMARY KEY (`id\_usuario`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1; --motor de la base de datos a crear

-- La exportación de datos fue deseleccionada.

/\*!40101 SET SQL\_MODE=IFNULL(@OLD\_SQL\_MODE, ' ') \*/; --

/\*!40014 SET FOREIGN\_KEY\_CHECKS=IFNULL(@OLD\_FOREIGN\_KEY\_CHECKS, 1) \*/;

/\*!40101 SET CHARACTER\_SET\_CLIENT=@OLD\_CHARACTER\_SET\_CLIENT \*/;

/\*!40111 SET SQL\_NOTES=IFNULL(@OLD\_SQL\_NOTES, 1) \*/;