

ANTEPROYECTO SOBRE EL **PROYECTO INTEGRADO CURSO** **2024/2025**

2º Desarrollo de Aplicaciones Web

José García De Lemus

ÍNDICE:

Tecnologías a utilizar	3
Diagrama UML	4
Diagrama E/R	6
Comentarios sobre las elecciones y dificultades encontradas en el desarrollo del proyecto	
Manual de usuario	

1. Tecnologías a utilizar

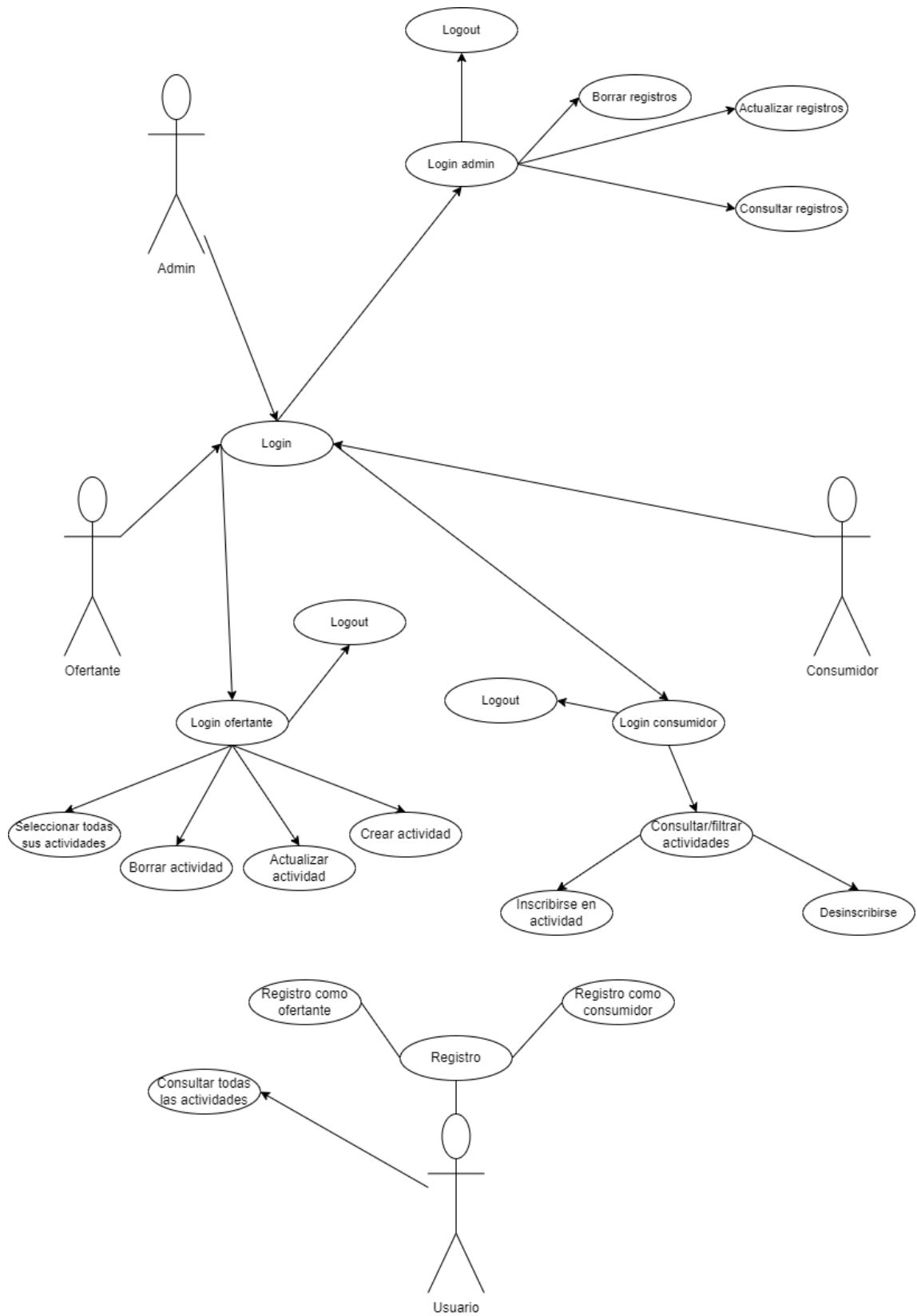
En el desarrollo del proyecto utilizaré las siguientes tecnologías:

- JAVA SPRING para el desarrollo Back End.
- JavaScript (Angular) para el desarrollo Front End.
- SQL(MariaDB) para la creación y gestión de la base de datos.
- HTML, CSS y BootStrap para las labores de diseño.
- GitHub para la gestión de versiones.

Cabe también mencionar que haré uso de Docker para el desarrollo del proyecto en local y de proxmox para la exposición y despliegue del proyecto.

La razón por la que he escogido estas tecnologías es porque son con las que más familiarizado estoy actualmente, y porque las otras de las que podría disponer (las que trabajo actualmente en las FCT) generan código automáticamente, por lo que no están permitidas.

2. Diagrama UML

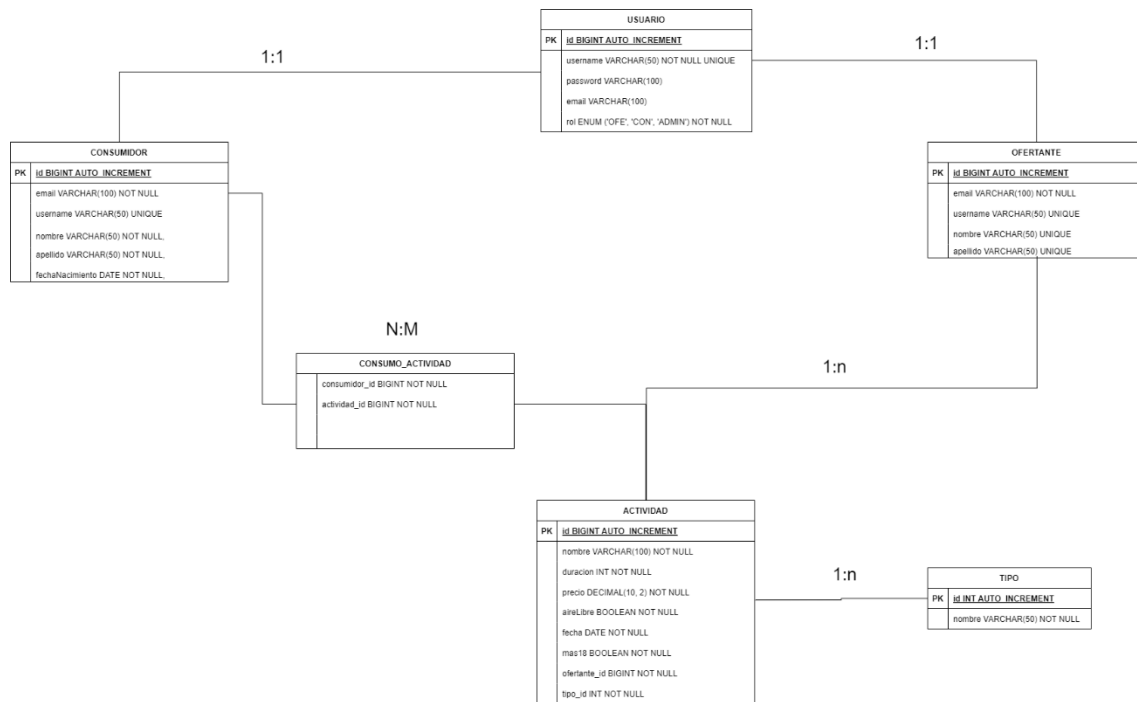


Al registro como tal sólo accederán los usuarios sin registrar (sin especificar consumidores u ofertantes) ya que una vez se registra un usuario, se le asigna un rol obligatoriamente. Los consumidores pueden consultar las actividades ofertadas, filtrarlas, inscribirse en ellas o desinscribirse. Los ofertantes podrán consultar sus actividades, crear más, actualizarlas o borrar alguna (un CRUD de sus actividades, en resumen).

También está la posibilidad de que haga login el administrador, que básicamente puede modificar, borrar o consultar los registros que él quiera.

Por último, también añadiré la posibilidad de acceder al sitio web sin registrarse, en cuyo caso saldrá una lista de todas las actividades, y en caso de seleccionar cualquier opción, se redirigirá al registro.

3. Diagrama E/R



La tabla usuarios se relaciona por 1:1 tanto con ofertantes como con consumidores, la clave foránea es el nombre de usuario, ya que un usuario puede asignarse el mismo email a la cuenta de ofertante y consumidor, me parecía más lógico que llegaran los supuestos correos al mismo email.

A su vez, los consumidores tienen una relación N:M con actividades ya que un consumidor puede estar registrado a muchas actividades, y una actividad puede tener varios consumidores inscritos. En el caso de los ofertantes no es así, un ofertante puede ofrecer varias actividades, pero una actividad solo estará ofrecida por un solo ofertante.

Por último, una actividad puede ser de un tipo, pero un tipo de actividades puede tener varias actividades asociadas, por lo que la relación es 1:N.

```

CREATE SCHEMA IF NOT EXISTS pInt;

USE pInt;

CREATE TABLE
    IF NOT EXISTS USUARIO (
        id BIGINT AUTO_INCREMENT PRIMARY KEY,
        username VARCHAR(50) NOT NULL UNIQUE,
        password VARCHAR(100),
        email VARCHAR(100),
        rol ENUM ('OFE', 'CON', 'ADMIN') NOT NULL
    );

CREATE TABLE
    IF NOT EXISTS TIPO (
        id INT AUTO_INCREMENT PRIMARY KEY,
        nombre VARCHAR(50) NOT NULL
    );

CREATE TABLE
    IF NOT EXISTS CONSUMIDOR (
        id BIGINT AUTO_INCREMENT PRIMARY KEY,
        nombre VARCHAR(50) NOT NULL,
        apellido VARCHAR(50) NOT NULL,
        fecha_nacimiento DATE NOT NULL,
        email VARCHAR(100) NOT NULL,
        username VARCHAR(50) UNIQUE,
        CONSTRAINT fkConsumidorUsuario FOREIGN KEY (username) REFERENCES
USUARIO (username) ON DELETE CASCADE ON UPDATE CASCADE
    );

CREATE TABLE
    IF NOT EXISTS OFERTANTE (
        id BIGINT AUTO_INCREMENT PRIMARY KEY,
        nombre VARCHAR(50) NOT NULL,
        apellido VARCHAR(50) NOT NULL,
        email VARCHAR(100) NOT NULL,
        username VARCHAR(50) UNIQUE,
        CONSTRAINT fkOfertanteUsuario FOREIGN KEY (username) REFERENCES
USUARIO (username) ON DELETE CASCADE ON UPDATE CASCADE
    );

```

```

CREATE TABLE
  IF NOT EXISTS ACTIVIDAD (
    id BIGINT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL,
    duracion INT NOT NULL,
    precio DECIMAL(10, 2) NOT NULL,
    aireLibre BOOLEAN NOT NULL,
    fecha DATE NOT NULL,
    mas18 BOOLEAN NOT NULL,
    ofertante_id BIGINT NOT NULL,
    tipo_id INT NOT NULL,
    FOREIGN KEY (ofertante_id) REFERENCES OFERTANTE (id) ON DELETE
CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (tipo_id) REFERENCES TIPO (id) ON DELETE CASCADE ON
UPDATE CASCADE
  );

CREATE TABLE
  IF NOT EXISTS CONSUMO_ACTIVIDAD (
    consumidor_id BIGINT NOT NULL,
    actividad_id BIGINT NOT NULL,
    PRIMARY KEY (consumidor_id, actividad_id),
    FOREIGN KEY (consumidor_id) REFERENCES CONSUMIDOR (id) ON DELETE
CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (actividad_id) REFERENCES ACTIVIDAD (id) ON DELETE
CASCADE ON UPDATE CASCADE
  );

```


4. Comentarios sobre las elecciones y dificultades
5. encontradas en el desarrollo del proyecto
6. Manual de Usuario