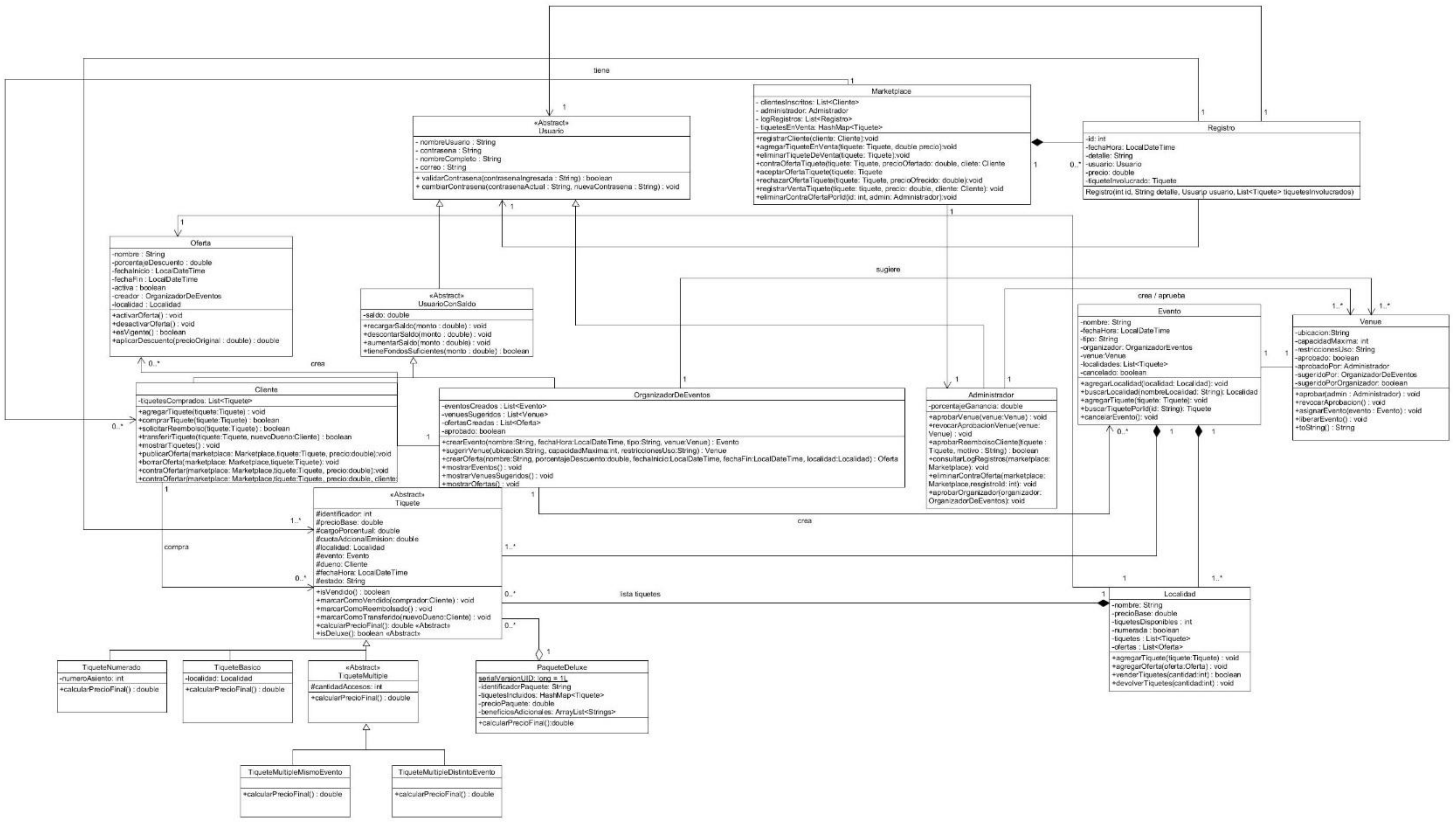
Grupo Industemas

Estudiantes:

- **Andrea Vargas Torres: 202321476**
- **Felipe Correa Ganitsky: 202320667**
- **Sebastian Salazar Sanchez: 202324631**

Figura 1

Diagrama de clases de diseño completo (clases, relaciones, atributos y métodos)



La **Figura 1** muestra el diagrama de clases de diseño correspondiente al sistema *Boletamaster*. En este diagrama se representan las clases principales del modelo, sus atributos, métodos y las relaciones principales existentes entre ellas.

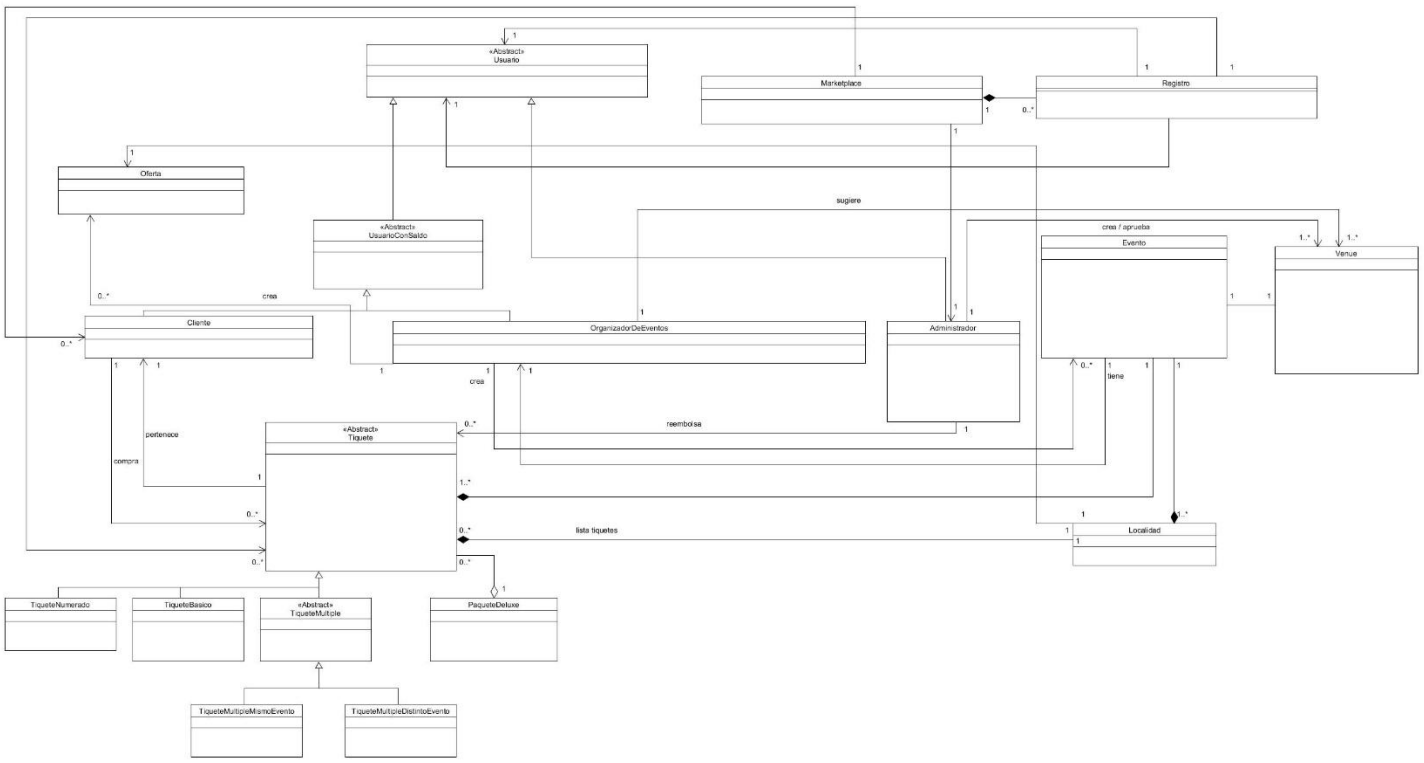
En la implementación del diseño, las clases se agruparon de acuerdo con su responsabilidad dentro del sistema, organizadas en distintos paquetes:

- El paquete **logica.evento** contiene las clases relacionadas con la gestión de los eventos, tales como Evento, Localidad, Oferta y Venue.
- El paquete **logica.tiquete** agrupa la clase abstracta Tiquete, sus subclases y el PaqueteDeluxe (el cual está compuesto por tiquetes).
- El paquete **logica.usuario** contiene las clases Cliente, OrganizadorDeEventos y Administrador, las cuales heredan de la clase Usuario y la clase UsuarioConSaldo (solo Cliente y OrganizadorDeEventos).
- El paquete **logica.reventa** contiene las clases Marketplace y Registro.

Por último, en la **Figura 1** se pueden observar los métodos implementados en cada clase, definidos de acuerdo con las funciones específicas que cada componente desempeña dentro del sistema *Boletamaster*.

Figura 2

Diagrama de clases de alto nivel (relaciones)



La **Figura 2** muestra las relaciones entre las clases que conforman el sistema Boletamaster. En el diagrama se representan las asociaciones, composiciones, agregaciones y relaciones de herencia que estructuran la interacción entre los diferentes componentes del modelo.

Primero, se establecen varias relaciones de composición, que reflejan dependencias fuertes donde la existencia de un objeto depende directamente de otro. La clase Evento compone a la clase Localidad, ya que un evento está formado por una o más localidades. Si se elimina el evento, también se eliminan las localidades asociadas. A su vez, cada Localidad compone a varios objetos de tipo Tiquete, puesto que los tiquetes pertenecen exclusivamente a una localidad específica dentro de un evento. De manera similar, la clase PaqueteDeluxe está compuesta por múltiples objetos Tiquete, representando un conjunto de entradas agrupadas bajo un mismo paquete.

Por otro lado, existen diversas relaciones de asociación, que representan interacciones entre clases sin dependencia estructural:

- Cada Evento se realiza en un Venue.
- Un Evento tiene un OrganizadorDeEventos y un OrganizadorDeEventos puede crear varios Eventos.
- Un OrganizadorDeEventos puede sugerir diversos Venues.
- Un Admistrador puede aprobar varios Venues.

- Una Localidad puede tener una o ninguna Oferta.
- Un OrganizadorDeEventos puede crear varias Ofertas.
- Un Cliente puede comprar o transferir Tiquetes.

Asimismo, el sistema incorpora varias relaciones de herencia que favorecen la reutilización y especialización del comportamiento:

- La clase abstracta Usuario actúa como superclase de Administrador, OrganizadorDeEventos y Cliente.
- Adicionalmente, UsuarioConSaldo extiende la funcionalidad de Usuario y sirve como base para Cliente y OrganizadorDeEventos, quienes requieren manejo de saldo.
- En el dominio de los tiquetes, la clase abstracta Tiquete es la superclase de TiqueteMultiple el cual representa un tipo de entrada con caracteristicas adiconales.

Adicionalemnte, los requerimiento del Proyecto 2 requieren de las siguientes relaciones:

- El Marketplace tiene una lista de Registros asociados (relacion de composicion).
- El Marketplace tiene una lista de Clientes inscritos (Asociación)
- El Marketplace tiene un Administrador (Asociación).
- El Marketplace tiene una serie de tiquetes (Asociación).
- Un registro tiene un usuario asociado, que es quien ejecita la acción (Asociación).
- Un registro involucra a uno o mas tiquetes (Asociación).

En conjunto, estas relaciones estructuran la lógica del sistema *Boletamaster*, garantizando coherencia entre los módulos de eventos, usuarios y tiquetes. El diseño propuesto refleja una arquitectura orientada a objetos bien definida, en la que cada clase cumple una función específica dentro del modelo general.

Nota: Los diagramas presentados en las **Figuras 1** y **2** se pueden visualizar con mayor detalle en los archivos PDF anexos la carpeta Documento_Diagramas