

Grupo 4: CONCIERTOS

Se quiere diseñar un sistema informático para la gestión de una agencia de conciertos, de acuerdo a la descripción siguiente:

La agencia se encarga de la organización de conciertos. Todos los conciertos se identifican y son en una fecha y a una hora concretas. La agencia también organiza giras, que se componen de varios conciertos, aunque no todos los conciertos forman parte de una gira. Todas las giras tienen un nombre único y se almacena su fecha de apertura y fecha de cierre de la gira. Al final de cada gira se tiene un informe, que debe ser revisado, y del que se tiene su identificador y la descripción de la gira. Los informes son generados por el personal de la agencia y revisados por un supervisor. Son también ellos quienes conforman las giras de los conciertos, contactando con los artistas y sus representantes, pero no guardamos en este sistema informático ninguna información sobre el personal de la agencia.

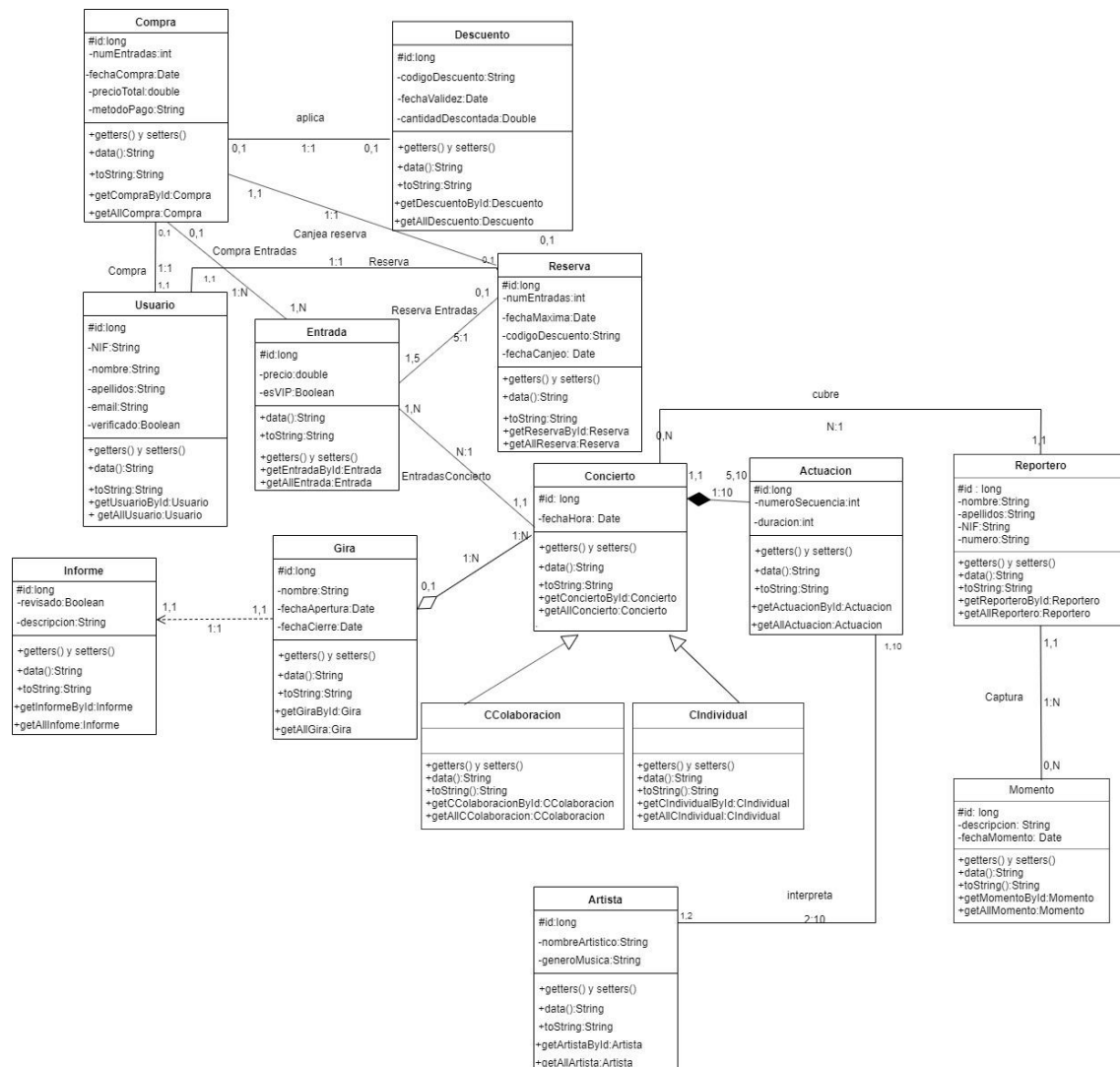
Los conciertos se componen de un número entre 5 y 10 actuaciones diferentes. Todas ellas van identificadas, tienen un número de secuencia (que coincide con el orden de la actuación en el concierto) y se almacena su duración en minutos. En cada actuación intervienen uno o dos artistas, si bien éstos pueden intervenir en varias actuaciones de un mismo concierto. Los conciertos pueden ser o bien individuales o bien colaboraciones. Los conciertos individuales son aquellos en los que actúa un solo artista, mientras que en las colaboraciones al menos hay dos artistas colaborando. De cada artista tenemos su identificador, su nombre artístico, y el género de su música, que se codifica con una letra mayúscula de entre las de un catálogo del que dispone la agencia.

Las entradas de un concierto solo valen para ese concierto. Se identifican unívocamente, tienen un precio en euros y algunas de ellas son V.I.P. Algunas de estas entradas no se adquieren en taquilla, sino que se gestionan a través de la agencia. Para ello, a los usuarios se les registra en el sistema, dotándolos de un identificador único, y se almacenan su nombre y apellidos, su email y su NIF. Estos usuarios, una vez verificados, pueden hacer una única reserva para un máximo de 5 entradas para un concierto concreto. La reserva tiene un identificador y una fecha máxima para ser canjeada por las entradas. De no ser así, se pierde la reserva y las entradas vuelven a estar disponibles. Otra opción es que un usuario realice una compra, la cual se compone de una o más entradas. Las compras siempre van identificadas, tiene un precio total en euros y se almacena el método de pago. Existen descuentos que pueden ser aplicados en una compra de cualquier tipo. Cada descuento se puede aplicar una única vez, siempre que sea antes de su fecha de validez. Los descuentos tienen un código único que el usuario debe introducir al hacer la compra y, si es validado, se descuenta del importe total de la compra la cantidad en euros marcada en el descuento.

Ampliaciones:

- Las reservas de entradas tienen una fecha límite para ser canjeadas por las entradas reales. Indicar qué cambios habría que realizar sobre el modelo anterior para registrar el momento en que se produce ese canje.
- La agencia cuenta con ciertos reporteros que documentan los conciertos. Así, de cada actuación de cada concierto, un reportero captura varios momentos. Éstos van identificados, se guarda una descripción breve y la hora en que ocurrieron. De los reporteros sabemos su nombre y apellidos, NIF y teléfono. Indicar qué cambios habría que realizar en el modelo para incluir la información sobre los momentos que capturan los reporteros durante las giras de conciertos.

DIAGRAMA DE CLASES



Explicación Diagrama de clases

Todas las clases tienen:

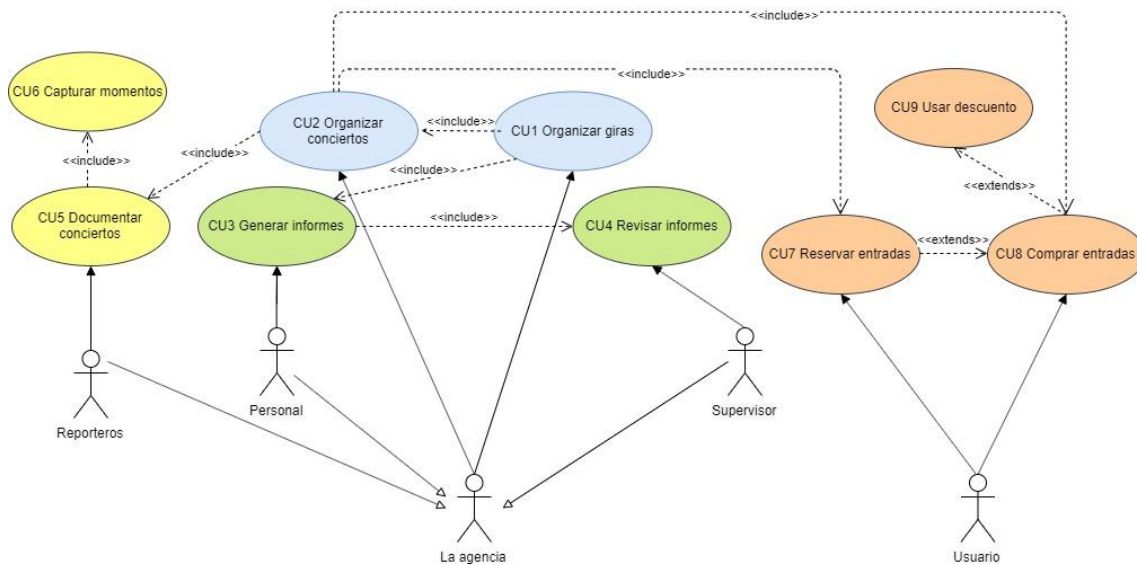
- Un atributo id de tipo long con visibilidad protected, que se generará automáticamente.
- Los métodos getters y setters.
- Los tres constructores básicos (por defecto, por atributos y de copia).
- El método data. Devuelve los atributos de la clase.
- El método toString
- El método nuevoXXX. No recibe atributos, y pide por la entrada de datos estándar que el usuario introduzca un valor para cada atributo de la clase. Al final del método, se muestran al usuario los datos introducidos y se pide su confirmación, y devuelve un objeto de esa clase.
- El método getAllXXX. No recibe atributos y devuelve un ArrayList de todos los objetos de la clase.
- El método getXXXById. Recibe un id de tipo long y, si existe, devuelve el objeto de esa clase con ese id, o devuelve null si no existe.

Cada clase en específico

- Actuación: Tiene los atributos numeroSecuencia (orden de la actuación en el concierto) y duracion (tiempo que dura en minutos) de tipo int, un ArrayList de objetos de tipo Artista con la lista de artistas que actúan en la actuación y un atributo idConcierto (donde se guarda el concierto de la actuación) de tipo long.
- Artista: Tiene los atributos nombreArtistico y generoMusica de tipo String y un ArrayList de objetos de tipo Actuacion con la lista de actuaciones que tiene el artista.
- CColaboracion: Hereda de Concierto. No tiene atributos propios.
- CIndividual: Hereda de Concierto. No tiene atributos propios.
- Compra: Tiene los atributos numEntradas (numero de entradas que se compran) de tipo int, fechaCompra (fecha en la que se compra la/s entrada/s) de tipo Date, precioTotal de tipo double, metodoPago de tipo String, idDescuento (identificador del descuento que se le resta a la compra) de tipo long. También incluye un atributo de tipo usuario que indica qué usuario a realizado la compra.
- Concierto: Tiene los atributos fechaHora de tipo Date que indica la fecha y hora del concierto y un ArrayList de objetos de tipo Actuacion que indica la lista que contiene las actuaciones de un concierto.

- Descuento: Tiene los atributos `codigoDescuento` (código al hacer la compra) de tipo `String`, `fechaValidez` (hasta cuando sirve el descuento) de tipo `Date` y `cantidadDescontada` de tipo `double`.
- Entrada: Tiene los atributos `precio` de tipo `double`, `esVIP` (algunas entradas son V.I.P.) de tipo `boolean` y un atributo de tipo `Concierto` que dice para que concierto sirve la entrada.
- Gira: Tiene los atributos `nombre` (nombre de la gira) de tipo `String`, `fechaApertura` y `fechaCierre` de tipo `Date` y un `ArrayList` de objetos de tipo `Concierto` con la lista de conciertos que hay en la gira.
- Informe: Tiene los atributos `revisado` (los informes se revisan por un supervisor) de tipo `boolean`, `descripcion` (guarda la descripción de la gira) de tipo `String` y un atributo de tipo `Gira` que indica la gira a la que va asociado el informe.
- Momento: Tiene los atributos `descripcion` (descripción breve de cada momento) de tipo `String`, `fechaMomento` (hora en la que ocurrieron) de tipo `Date` y un atributo de tipo `Reportero` que indica el reportero que capturó el momento.
- Reportero: Tiene los atributos `nombre`, `apellidos`, `NIF` y `numero`, de tipo `String` y un atributo de tipo `Concierto` que indica a qué concierto está asignado el reportero.
- Reserva: Tiene los atributos `numEntradas` (cuántas se van a reservar) de tipo `int`, `fechaMaxima` (fecha de cancelación de la reserva) y `fechaCanjeo` de tipo `Date` y `codigoDescuento` (código al hacer la compra) de tipo `String`.
- Usuario: Tiene los atributos `NIF`, `nombre` y `apellidos`, de tipo `String`, `verificado` (necesitan estar verificados para hacer la reserva) de tipo `boolean` y un atributo de tipo `Reserva` que indica la reserva que ha hecho o no el usuario.

DIAGRAMA DE CASOS DE USO



Subsistemas

- Subsistema azul: La agencia organiza giras que se componen de conciertos y también organiza concierto que pueden o no formar parte de una gira.
- Subsistema verde: El Personal de la agencia genera un informe al final de cada gira y el Supervisor de la agencia los revisa.
- Subsistema amarillo: Los reporteros de la agencia se encargan de documentar los conciertos capturando varios momentos de cada actuación.
- Subsistema rojo: El usuario puede reservar y comprar entradas. Si la reserva se hace efectiva proceden a comprar entradas, si caduca no.

Actores

- La agencia: Este actor hereda de sus trabajadores. Organiza giras y conciertos.
- Reporteros: Se encarga de documentar conciertos y capturar momentos.
- Personal: Se encarga de generar informes de cada gira.
- Supervisor: Se encarga de revisar el informe realizado por el personal.
- Usuario: Este actor reserva o compra entradas para un concierto. Puede aplicar descuentos a sus compras en caso de que tenga algún código.

Casos de uso

- CU1: La agencia organiza giras que se componen de varios conciertos.
- CU2: La agencia se encarga de la organización de conciertos.
- CU3: Al final de cada gira (CU2), el personal de la agencia se encarga de generar un informe .
- CU4: Los informes (CU3) deben ser revisados por un supervisor.
- CU5: La agencia cuenta con reporteros que documentan conciertos (CU2).
- CU6: Dentro de la documentación (CU5), de cada actuación de cada concierto, los reporteros capturan momentos.
- CU7: Los usuarios reservan entradas para un concierto (CU2), que pueden ser canjeadas por entradas (Comprar entradas – CU8) antes del vencimiento de la reserva.
- CU8: Los usuarios pueden comprar entradas para acudir a conciertos (CU2).
- CU9: Los usuarios pueden aplicar descuentos a sus compras (CU8) para descontar la cantidad que están en ellos.

REPARTO DE TAREAS

Primer ejercicio:

Diagrama de clases: Luis

Implementación:

- Actuacion: Jairo
- Artista: Luis
- Compra: Luis
- Concierto: David
- Descuento: Jairo
- Entrada: Luis
- Gira: David
- Informe: David
- Reserva: Jairo
- Usuario: Luis

Segundo Ejercicio:

Diagrama de clases: Luis y David

Diagrama de casos de uso: Jairo

Implementación:

- Actuacion: Luis
- Artista: Luis
- CColaboracion: David
- CIndividual: Jairo
- Compra: Luis
- Concierto: Luis
- Descuento: Luis
- Entrada: Jairo

- Gira: Jairo

- Informe: Jairo

- Momento: Luis

- Reportero: David

- Reserva: David

- Usuario: David

Documentación: Luis, David y Jairo