Diagrama con Patrones

Justificacion

# Observer

Utilizamos el patrón observer ya que queremos resolver el problema de cómo notificar al usuario cuando finaliza un evento o sorteo. Mediante el patrón observer un usuario se puede suscribir a un sorteo o evento y cuando el sorteo o evento cambia su estado la clase observador, que contiene a los clientes que están suscriptos, le enviará una notificación del cambio de estado diciendo como salió en ese evento o sorteo

# Builder

Utilizamos el patrón builder ya que queremos resolver el problema de cómo crear diferentes combos con descuentos los cuales tienen una cantidad de pases, tipo y souvenires especificados. Mediante el patrón builder se puede crear varias clases concretas las cuales modifican a una instancia de la clase combo, por ejemplo, una clase concreta ComboFamiliar puede setear la instancia que se crea del combo con una cantidad de pases específicos, ejemplo de familia tipo 4 personas = 4 pases, con un listado de souvenires propios para cada individuo y un descuento por comprar el combo de un x porcentaje. Esto le da flexibilidad a la hora de crear promociones o vender pases para fechas poco concurridas, como por ejemplo, si existen fechas que segun la estadistica baja la cantidad de gente que asiste al parque se crea un combo para dicho periodo de tiempo con el nombre “ComboPromoXFechas”.



