***Informe***

**Revisión obtenida por el profesor:**

1) No cierra que el tp esté en inglés, el enunciado está en castellano, nosotros hablamos en castellano, lo hizo más difícil de entender. También había algunos errores de ortografía (with sin h), que probablemente serían menos frecuentes en español ;-). Finalmente, tomando la idea de lenguaje común (ubiquitous language) de Evans en Domain Driven Design, el lenguage tiene que ser uno único compartido por todos, por lo tanto, arranca con usar el lenguaje hablado.

2) Las pruebas de PromotionTest son correctas.

3) El método:

public double getCost() {

double cost = 0;

cost = calculateCost(this.getAttractions());

cost = cost- ((cost \* percentage)/100);

return cost;

}

podría escribirse más claramente así:

public double getCost() {

double cost = calculateCost(this.getAttractions());

return cost- ((cost \* percentage)/100);

}

4) No cierra que PercentagePromotion y AbsolutePromotionn redefinan applyPromotion y que el código sea el mismo. Debería estar en Promotion. Tampoco me termina de cerrar el código de ese método, no lo entiendo, tal vez falten algunas aclaraciones sobre la fórmula. El de AXB sí lo entiendo.

5) El método calculateCost está repetido con la misma implementación en varios lugares, debería estar en uno sólo, la redundancia innecesaria es una de las peores males del software

6) ¿No sería calculateTravelTime un método de itinerario? No me imagino a la secretaría de turismo haciendo ese cálculo. Idem con el tiempo total.

7) Disponibility no es la palabra, se dice Availability, igual debería estar en castellano.

8) En verifyAttractionIsAppropiateForUser, si la atracción es apropiada, la borrás, no se entiende por qué. En cualquier caso, el nombre no sugiere que haga eso, dice verify, no modify.

**Correcciones aplicadas a la devolución:**

1) Se corrigió todo el código y se lo tradujo al castellano.

3)Se aplicó un refactor al método señalado.

4) Se asignó el método “applyPromotion”, ahora llamado “aplicarCostoDePromocion” a la clase Promocion, de manera que el único que terminaría redefiniendo el mismo sería PromocionPorcentual.

Se aplicó un refactor a ese mismo método tratando de hacerlo más declarativo y entendible.

5) Se asignó el método calcularCosto a la clase Promocion para evitar las repeticiones innecesarias.

6)Respecto a este ítem, consideré que dado que el usuario lo tiene asignado la Secretaría de turismo, me pareció más apropiado que determine él, ell tiempo de viaje total para el recorrido, ya que es el que tiene la información de la velocidad a la que se desplaza el usuario.

Consideré que Una sugerencia debía tener la información necesaria para el usuario, pero no la lógica del negocio.

8)Se aplicó un refactor a la forma en que se generan las sugerencias.

**Cambio importante:**

He realizado un gran cambio a la forma en que se generan las sugerencias, por considerar que la forma en que lo estaba realizando era una complejidad innecesaria.

En el pasado, a medida que se agregaba una atracción a la sugerencia, se aplicaban las promociones de manera que quede un mayor margen de dinero, a causa del descuento, para que entren más atracciones.

Ahora agrego las atracciones hasta que me lo permita el dinero y el tiempo que cuenta el usuario. Y posteriormente aplico los descuentos de las promociones.

**Correcciones aplicadas después de la revisión (Informe de impacto):**

Al depender de una jerarquía de herencia con Promocion y sus implementaciones, se encontró una dificultad de diseño al agregar la promoción PromocionFamiliar, a la hora de poder aplicar el método redefinido aplicarCostoDePromocion(), ya que este mismo, necesitaba pararmetrizar la cantidad de entradas para realizar el cálculo necesario.

Consideré que por la forma en que genero las sugerencias (sin aplicar ninguna promoción a las atracciones hasta que tengo la sugerencia completa), era necesario aplicar primero las 3 promociones que tenía en un principio y por último la nueva Promocion PromocionFamiliar, ya que de ésta manera el descuento iba a ser mayor porque se aplicaría sobre el descuento obtenido por las otras promociones.

Se agregó un atributo más a las sugerencias, cotoXCantidadDeEntradas, de tal manera que además de calcularse el costo total por persona también se cuente con el dato del costo total por la cantidad de entradas solicitadas.