

Autoevaluación

Descripción breve de aquellos requerimientos funcionales o de diseño que faltan o no funcionan correctamente y/o pueden ser mejorados y el motivo.

En el tablero, en la tabla de recargas, no pudimos hacer que el monto de bonificación se muestre correctamente en el caso de los tipos de bonificaciones frecuentes. Así que para poder cumplir con el requerimiento debimos agregar un atributo int en el transito que indica el descuento aplicado en el tránsito.

Creemos que faltan algunas excepciones y cursos alternativos bastante esenciales a la hora de validar datos. Aunque entendemos que esto no sea parte del programa de la materia, de todas maneras, agregamos algunos.

Ejemplos:

- Ingreso de datos en las vistas sean del tipo requerido.
- Ingreso de números negativos y cálculos que resulten en números ilógicos.

Creemos que de la misma manera que realizamos herencia en algunas clases de dominio porque así lo sería mejor en un caso de proyecto real, también se debería contemplar que la lógica de dichas clases acompañara la creación de un futuro método donde se puedan ingresar objetos de los tipos de esa clase.

Teniendo en cuenta las reglas de calificación (ver más abajo) auto califique su trabajo. La calificación debe incluir un detalle por áreas y una calificación general.

Concepto	Pts.
Funcionalidad: Implementación de la funcionalidad solicitada en Java	(-39 a 0)
Diagrama conceptual de domino	2
Diagrama de diseño	2
División lógica	5
Experto	6
Fachada y arquitectura	2.5
MVC	6
Observador y manejo de eventos	6
Manejo de excepciones	2.5
Polimorfismo	5
Autoevaluación	1
Total:	38

El punto de división lógica descontado corresponde al requerimiento funcional que no pudimos cumplir al 100% realizando la división lógica correcta.

Descontamos medio punto del manejo de excepciones que, aunque pensamos que es muy correcto podrían agregarse excepciones adicionales, pero en realidad tuvimos en cuentas solamente las de los requerimientos.

En el punto de fachada y arquitectura nos descontamos medio punto porque inicialmente habíamos hecho que la fachada en su totalidad sea observable pero luego terminamos haciendo solamente observable algunas clases. El descuento proviene de no estar seguros de si la manera implementada finalmente sea la correcta. Nos basamos explícitamente en la devolución de una consulta realizada por teams, pero no pudimos corroborarlo con el material de clase.

Para finalizar, creemos que el resto de los puntos están cumplidos casi en su totalidad, realizamos exhaustivas revisiones del código siempre intentando mejorar la división lógica y la arquitectura.