

Autoevaluación Obligatorio Desarrollo y Diseño de Aplicaciones

Tomas Lanterna - Nicolas de Santa Ana

La forma de mostrar las mesas en la vista del mozo no nos pareció la indicada, ya que necesariamente tuvimos que usar un condicional para setearle los colores y un switch para modificar la cantidad de mesas mostradas.

Todo esto nos hizo abarcar 106 líneas de código en la vista, lo cual no nos pareció conveniente.

Para manejar los pedidos nuevos y para mostrar los pedidos de la mesa, tuvimos que incluir en el controlador una validación. Esta validación controla que ni el pedido ni la mesa lleguen vacías, lo que hacía que se detenga la aplicación.

Por otro lado, cuando se mantiene abierta la lista de mozo conectados para las transferencias, si un mozo ingresa la misma no se actualiza.

Por último cuando se cierra una Mesa, la lista de pedidos no se vacía sino que al seleccionarla de nuevo ahí se muestra vacía

Como conclusión, sentimos que contamos con los conocimientos necesarios y el tiempo para realizar la entrega.

Concepto	Pts.
Funcionalidad: Implementación de la funcionalidad solicitada en Java	(-39)
Diagrama conceptual de domino.	2
Diagrama de diseño	2
Claridad y reúso del código	2
División lógica	6
Experto	6
Fachada y arquitectura	3
MVC	6
Observador y manejo de eventos	4
Manejo de excepciones	3
Polimorfismo	5
Autoevaluación	1
Total:	40

Autocalificación del Proyecto

CONCEPTOS	PUNTOS
Funcionalidad: Implementación de la funcionalidad solicitada en Java	-----
Diagrama conceptual de domino	2
Diagrama de diseño	2
Claridad y reúso del código	2
División lógica	5
Experto	5
Fachada y arquitectura	3
MVC	5
Observador y manejo de eventos	4
Manejo de excepciones	3
Polimorfismo	5
Autoevaluación	1
Total	37