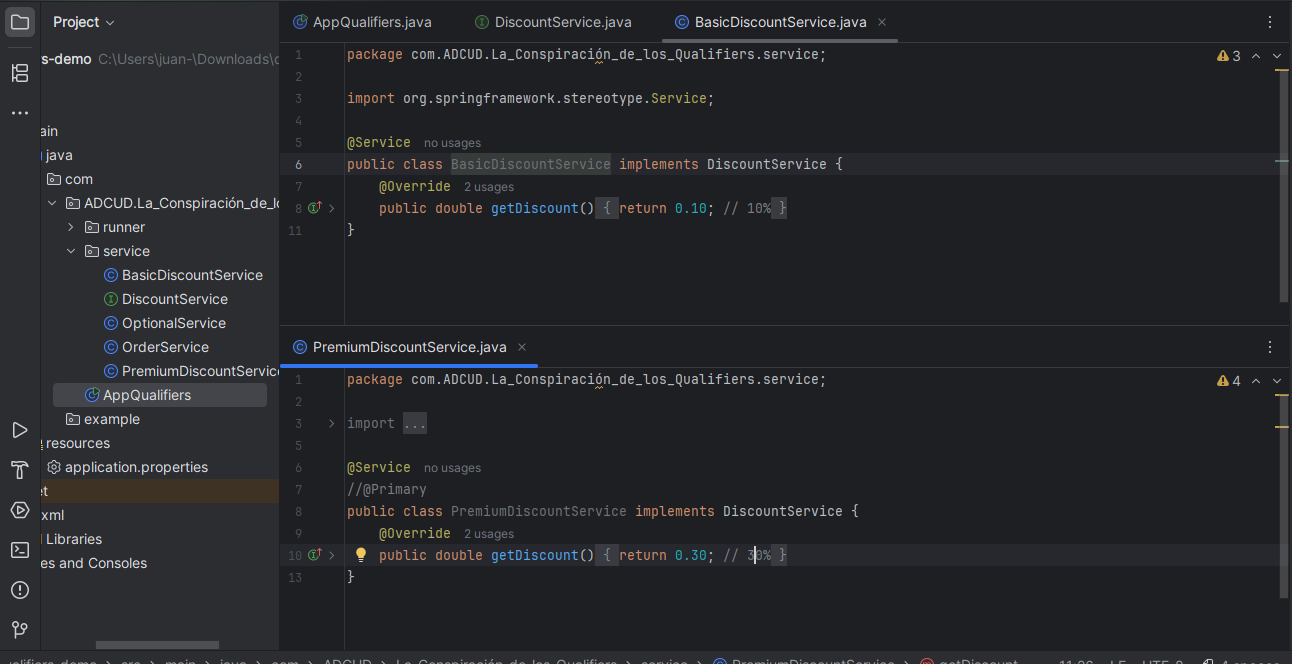
# Punto 3 del taller de Java Spring ACMUD

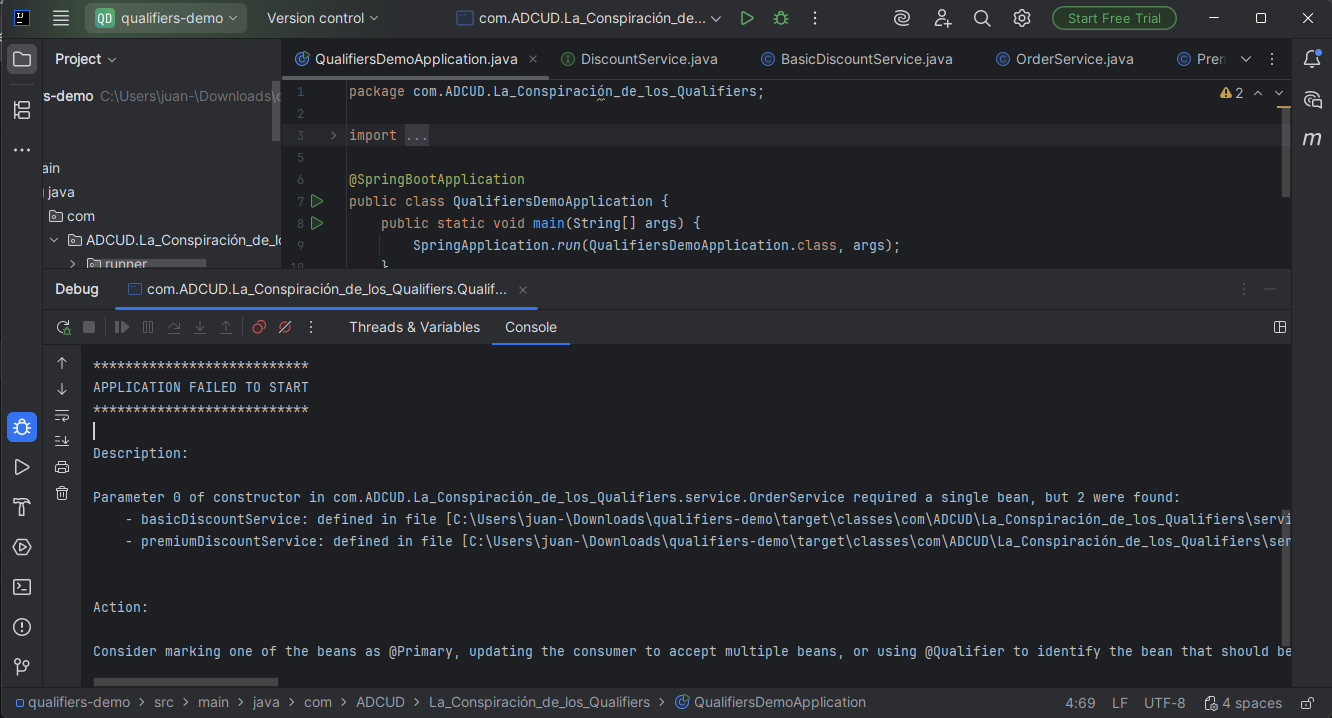
Retos:

1. Se crean dos implementaciones de la interface “DiscountService”,

BasicDiscountService y PremiumDiscountService.

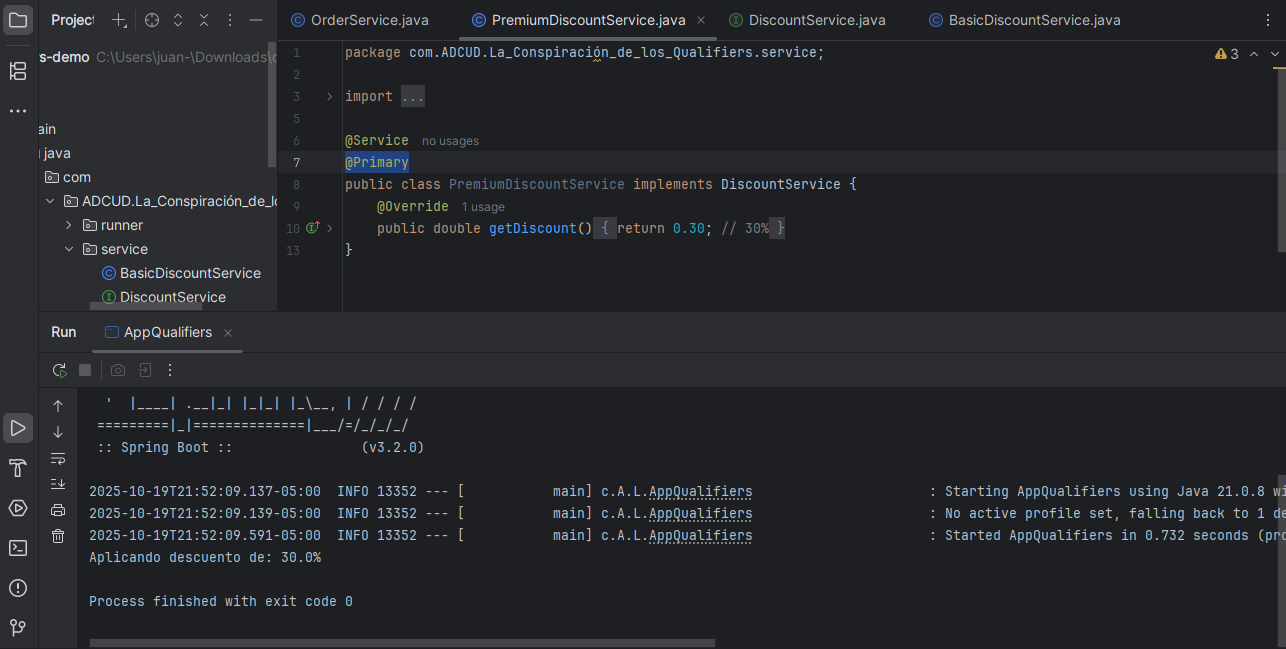


1. Al momento de inyectarlas en OrderService sin especificar cuál usar y ejecutar, se aprecia el siguiente error por consola.



Spring no puede decidir entre dos beans del mismo tipo y necesita una indicación clara.

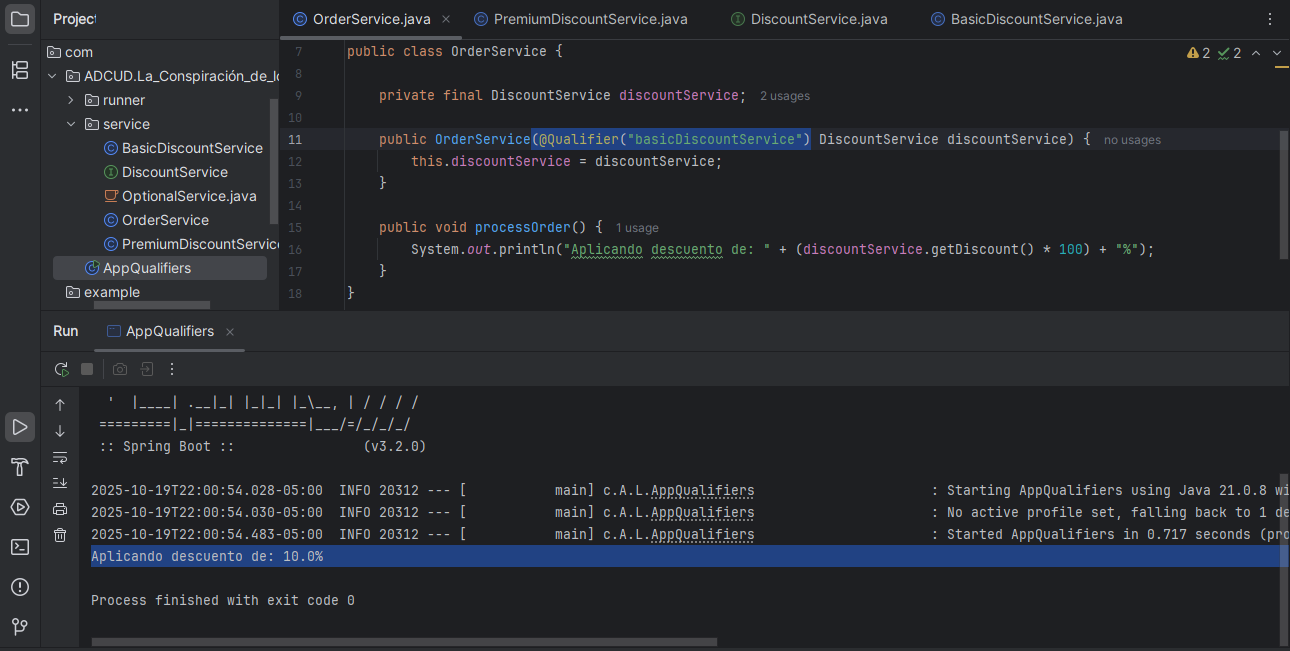
1. Se uso @Primary en una de ellas (PremiumDiscountService) y se verifico el resultado.



La aplicación ahora se ejecutó correctamente y OrderService recibió la implementación premium.

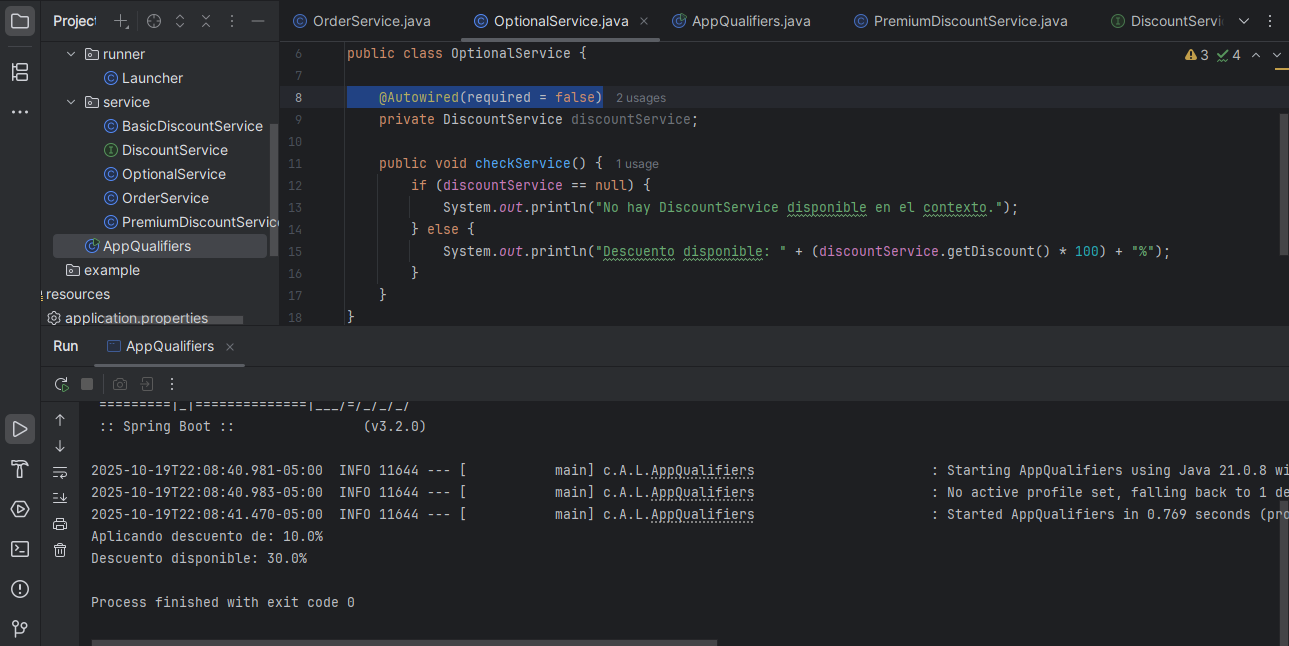
1. Se sustituyo con @Qualifier y se cambió el nombre del bean para provocar

confusión.



El servicio usó correctamente el descuento básico del 10%, debido a que @Qualifier permite especificar el bean exacto que se desea inyectar, lo cual es útil cuando se necesitan diferentes implementaciones en distintos lugares

1. Se uso @Autowired(required=false) en un bean opcional (OptionalService).



Si no hay un bean disponible, el campo discountService queda como null y no se lanza excepción.

En conclusión:

@Primary, se usa principalmente para definir un bean como el predeterminado entre varios del mismo tipo.

@Qualifier, se usa para especificar exactamente qué bean se debe inyectar.

@Autowired(required=false), se usa para hacer opcional la inyección, evitando errores si el bean no existiese.