**Taller #5 DPOO**

Juan Andrés Reyes 202210295

**Async Method Invocation**

Este tipo de diseño se usa con “beans de sesión”, que comunican un servidor con un cliente. Un cliente sincrónico es aquel en el que se debe esperar a que termine la ejecución para que el control regrese al cliente, es decir, se debe esperar al retorno para continuar la ejecución. Por otro lado, en un cliente asincrónico, el control regresa inmediatamente al cliente al momento de ejecutar un método. Cuando el control regresa al cliente, es posible invocar más métodos mientras se espera por la respuesta, del método anterior y así tener distintos métodos simultáneamente en ejecución. Con el control también es posible conectarse mediante “Java SE concurrency API” para obtener el resultado, cancelar la invocación o revisar excepciones.

El diseño asincrónico es útil para operaciones muy largas, tareas intensivas del procesador, tareas en segundo plano o posiblemente disminuir el tiempo de respuesta de una aplicación.

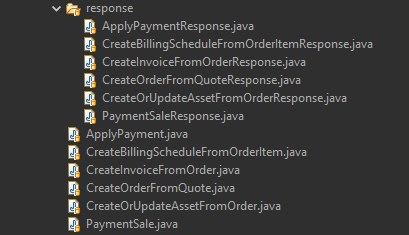
Un caso específico para esto es un pago. Mediante el cliente se realiza la invocación asincrónica de un método de pago. Tras ingresar la información necesaria, se le retorna el control al cliente, dado que el procesamiento de este pago puede llegar a tardar en retornar o incluso podría lanzar una excepción.

**RevCloudJavaSDK**

URI: <https://github.com/SalesforceLabs/RevCloudJavaSDK.git>

Datos generales: Este repositorio simplifica la invocación de los APIs de la “nube de ingresos” de Salesforce. Incluye autenticación, manejo de eventos asincrónicos y normalización de errores.

Para este proyecto, es necesario conectarse con varios clientes, dada la funcionalidad de este. En el código fuente, es posible observar que dividen las acciones de llamado y respuesta en diferentes carpetas. Las carpetas de respuesta llevan el nombre de “response”, es posible observar que por clases de llamado, corresponde una de respuesta. Un ejemplo de esto es la clase “Apply Payment” con su respectiva “ApplyPaymentResponse”.



La implementación permite precisamente realizar movimientos de ingresos y estadísticas, sin necesariamente detener el cliente.

<https://docs.oracle.com/javaee/6/tutorial/doc/gkkqg.html>

<https://www.codeproject.com/Articles/14931/Asynchronous-Method-Invocation>