Ejercicio 5

Mal olor: **Feature Envy**, el `agregarNumeroTelefono()` de Empresa, es una lógica que tiene que estar en **GestorNumerosDisponibles**. Refactoring: **Move Method**, de Empresa a GestorNumerosDisponibles. Y en Empresa se realiza una llamada al método de GestorNumerosDisponibles. Otros refactorings:

- Se hace un **Replace Temp With Query**, para la variable temporal boolean encontre.

Mal olor: **Feature Envy**, el `calcularMontoTotalLlamadas(Cliente)`, es una lógica que tiene que estar en **GestorNumerosDisponibles**. Refactoring: **Move Method**, de Empresa a GestorNumerosDisponibles. Otros refactorings:

- Se hace un **Move Field**, de las v.i descuentoJur y descuentoFis usadas por el metodo, y se hace también un **Encapsulate Field.**

Mal olor: Romper encapsulamiento en llamadas de Cliente. Refactoring: **Encapsulate Field**. Se agrega metodo `agregarLLamada()` .

Mal olor: **Feature Envy**, **Long Method** el método `calcularMontoTotalLlamadas()` Cliente está calculando el costo de la llamada dependiendo de su tipo. Se hace un **Move Method** a GestorNumerosDisponibles, y en Cliente se hace un Method Call.

- Para hacerlo también hay que llevar la variable temporal auxc.

Mal olor: **Long Method**, de `calcularMontoTotalLlamadas()` se hace un Extract Method, para calcular el tipo de descuento dependiendo del tipo de Cliente.

- Como tiene que usar la variable auxc, es pasada por parametro.

Mal olor: Reinventando la rueda en `calcularMontoTotalLlamadas()` al hacer un for elto. Refactoring a aplicar: **Replace Loop With Pipeline**.

Mal olor: Switch Statemet? en `calcularMontoTotalLlamadas()` , ya que tenía if para los distintos tipos de clientes. Se aplicó el refactoring **Replace Type Code with Subclasses**, que también incluyó **Pull Down Method** (o Extract Method? el método que retorna el descuento) y **Pull Down Field** (los descuentos). Para hacer esto se crearon las clases ClienteFisico, ClienteJuridico.

Mal olor: **Swtich Statement**, en `obtenerNumeroLibre()` para el tipo de generador. Se aplicó **Replace Conditional Logic With Strategy**. Se creó la jerarquía de clases para el Strategy, se hizo Move Method del método de generador, se agregó el setter de la estrategia y una vez testeado y funcionando se eliminó el switch del método.

- NOTA: Se hizo un switch en la clase cambiarTipoGenerador() , todo para mantener la funcionalidad tal cual (no modificar el TEST). Es otro bad smell pero con Refactoring no se puede cambiar el funcionamiento del código.

Mal olor: Switch Statement en `calcularPrecio()` de Llamada, se aplicó **Replace Type Code with Subclasses**, que inlcuyó **Pull Down Method**. Se crearon las clases LlamadaNacional y LlamadaInternacional.