



# DCA PRÁCTICA 1

Simulación del Versionado Semántico

Grado en Ingeniería Informática

Universidad de Alicante  
Antonio Miguel Corbi Bellot

Cristian Andrés Córdoba Silvestre (05988721G)

Email: cacs2@alu.ua.es

# Explicación del Programa Realizado

El programa elegido para simular el avance en las ramas **estable** y **master** ha sido una **Calculadora**, implementada en **JAVA**. A continuación, se explica en detalle cuales han sido los pasos, implementaciones y consideraciones que se han efectuado en cada una de las versiones desarrolladas:

1-. Se establece la versión inicial de la calculadora dentro de la rama **estable** como la versión **1.0.0**. Esta versión contiene una calculadora que permite realizar las siguientes operaciones básicas entre dos **números enteros**: suma, resta, multiplicación y división.

2-. Se cambia a la rama **master** para empezar a desarrollar las funcionalidades correspondientes para que la calculadora pueda operar con **números reales**. Se crea la versión **1.1.0** en la rama **master**. Dicha versión es una copia **exacta** de la versión **1.0.0** de la rama **estable**

3-. Se añade la versión **1.1.1** a la rama **master**. Esta versión añade las funcionalidades de **suma** y **resta** entre **dos números reales**.

4-. Se añade la versión **1.1.2** a la rama **master**. Esta versión añade las funcionalidades de **multiplicación** y **división** entre **dos números reales**.

5-. Durante el desarrollo de la versión **1.1.2**, un usuario detecta un error que provenía de la versión **1.0.0**. Este error se trata de que el programa no manejaba de forma correcta la división entre **cero**. Como la última versión **estable** era la **1.0.0**, para corregir dicho bug se crea la versión **1.0.1**. De esta forma, los usuarios tienen **solución inmediata** del bug, sin tener que esperar a que salga la nueva versión **estable**.

6-. Por otro lado, se corrige también el error de **división por cero** en la rama máster. Para ello, se crea la versión **1.1.3**, cuyo código es idéntico al de la versión **1.1.2**, con la diferencia de que tiene el error de **división por cero** corregido.

7-. Como ya se ha finalizado la implementación, aparentemente sin errores, de las **cuatro** operaciones básicas para **números reales**; publicamos para los usuarios la versión **2.0.0** en la rama **estable**. El código de dicha versión será **idéntico** al de la versión **1.1.3** de la rama **master**. Hemos incrementado el número mayor puesto que la implementación de las operaciones con números reales ha supuesto un cambio en la **API** del código que existía en la anterior versión estable. Esto es así puesto que, al tratarse JAVA de un lenguaje **fuertemente tipado**, al cambiar la signatura y el tipo devuelto de los métodos **Suma, Resta, Multiplicación y División**, cualquier usuario que haya escrito un código que llamase a cualquiera de estas funciones de la calculadora, dicho usuario va a tener que modificar su código **ya existente** teniendo en cuenta los cambios realizados en la nueva versión para evitar así **errores de compilación (incompatibilidad hacia atrás)**.

8-. A continuación, se procede a añadir las funcionalidades pertinentes para poder realizar las operaciones de **exponenciación** (las raíces cuadradas, raíces cúbicas, ... se entenderán como potencias con **exponente decimal**). Se crea la versión **2.1.0** en la rama **master**, cuyo código es **una copia** de la versión **2.0.0**.

9-. Se añade la versión **2.1.1** en la rama **master**. Dicha versión añade la funcionalidad de realizar la **exponenciación** entre dos números reales.

10-. Como ya hemos terminado la funcionalidad de la **exponenciación**, la publicamos en la rama **estable** en la versión **2.2.0**. El código de esta versión es **idéntico** al de la versión **2.1.1**. En este caso, al no haber **incompatibilidad hacia atrás**, no es necesario aumentar el número mayor de la versión.

11-. La siguiente funcionalidad a añadir sería la de introducir un parámetro para controlar la **precisión** del resultado obtenido en las operaciones de **multiplicación, división y exponenciación**. Añadimos la versión **2.3.0** en la rama **master**. Aumentamos el **número menor** y no el **número mayor** puesto que la funcionalidad de la **exponenciación** no ha supuesto ningún cambio ni en el **API** ni en el **ABI**.

12-. Se crea la versión **2.3.1** en la que se incluye la precisión para la **multiplicación** de dos números reales.

13-. Se crea la versión **2.3.2** en la que se incluye la precisión para la **división** de dos números reales.

14-. Se crea la versión **2.3.3** en la que se incluye la precisión para la **exponenciación** de dos números reales.

15-. Un usuario detecta y comunica un error en la lógica de la exponenciación: no se pueden hacer operaciones donde la base es **negativa** y el exponente es un **número no entero**. Se corrige el error en la rama **estable** en la versión **2.2.1**.

16-. Se corrige el mismo error en la rama **master** en la versión **2.3.4**

17-. Con la nueva funcionalidad del parámetro de la **precisión** implementada y funcional, se publica para los usuarios en la versión **3.0.0** en la rama **estable (ha habido cambios en la API)**. Dicha versión contiene idéntico código con respecto a la versión **2.3.4** de la rama **master**.

18-. A continuación, se pretende añadir que la calculadora pueda manejar las razones trigonométricas básicas, así como el logaritmo en base 10. Se crea la versión **3.1.0** de la rama **master**, cuyo código es idéntico al de la versión **3.0.0** de la rama **estable**.

19-. Se añade la funcionalidad de calcular el **Seno** en la versión **3.1.1** de la rama **master** (con el parámetro de precisión)

20-. Se añade la funcionalidad de calcular el **Coseno** en la versión **3.1.2** de la rama **master** (con el parámetro de precisión)

21-. Se añade la funcionalidad de calcular la **Tangente** en la versión **3.1.3** de la rama **master** (con el parámetro de precisión)

22-. Se añade la funcionalidad de calcular el **Logaritmo Decimal** en la versión **3.1.4** de la rama **master** (con el parámetro de precisión)

23-. Una vez completada la nueva funcionalidad, se publica para los usuarios en la rama **estable** en la versión **3.2.0**. Dicha versión contiene el mismo código que la versión **3.1.4**. No hay **incompatibilidad hacia atrás**, por lo que el número mayor no se aumenta.

24-. A continuación, se pretende añadir la funcionalidad de que la calculadora pueda realizar **sumas** y **restas** entre dos **números complejos**. Se crea la versión **3.3.0**, con código idéntico al de la versión **3.2.0**.

25-. Se crea la versión **3.3.1** en la rama **master**, donde se ha implementado la **suma** y **resta** entre **dos números** complejos.

26-. Como ya se han implementado las nuevas funcionalidades que se requerían, se publica el nuevo código a los usuarios en la versión **4.0.0** en la rama **estable**. Se ha producido un incremento en el número mayor puesto que se ha cambiado la **API** (las funciones **sumar** y **restar** reciben ahora parámetros diferentes, lo que las hace **incompatibles con las versiones anteriores**)

27-. Un usuario encuentra y comunica un bug con respecto a las razones trigonométricas: **no existe la tangente de un número múltiplo de 90° y no múltiplo de 180°**. Como ya se ha subido una nueva versión en la rama **estable**, este bug se soluciona en la versión **4.0.1** de la rama **estable**.

28-. El mismo usuario (**quien es muy listo**) vuelve a encontrar un bug con respecto al logaritmo decimal: **no existe el logaritmo decimal de un número real no positivo**. Como ya se ha subido una nueva versión en la rama **estable**, este bug se soluciona en la versión **4.0.2** de la rama **estable**.