Introducción

Un profesional de la Actuaría, debe estar constantemente actualizado y en conocimiento suficiente de las herramientas de tecnología de la información (TI) disponibles en su campo de trabajo. En nuestro estudio abordaremos primeramente un repaso básico de conceptos esenciales que nos abrirán el panorama y fungirán como la columna vertebral de todo el curso.

Programación Estructurada

Iniciemos con el paradigma de programación más elemental. Para ello nos valdremos del siguiente:

Teorema (de Böhm y Jacopini): *Un programa propio puede ser escrito utilizando únicamente 3 estructuras de control: secuenciales, de selección y de repetición.*

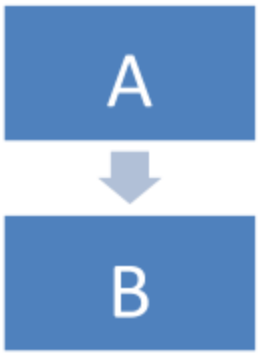
Un programa se dice propio si cumple las siguientes condiciones:

i) Se tiene solo un punto de entrada y un punto de salida
ii) Todas las sentencias del algoritmo son alcanzables (existe al menos un camino que conecta el principio y el fin)
iii) No posee ciclos infinitos

Revisemos a detalle cada una de las estructuras de control básicas:

Estructura secuencial

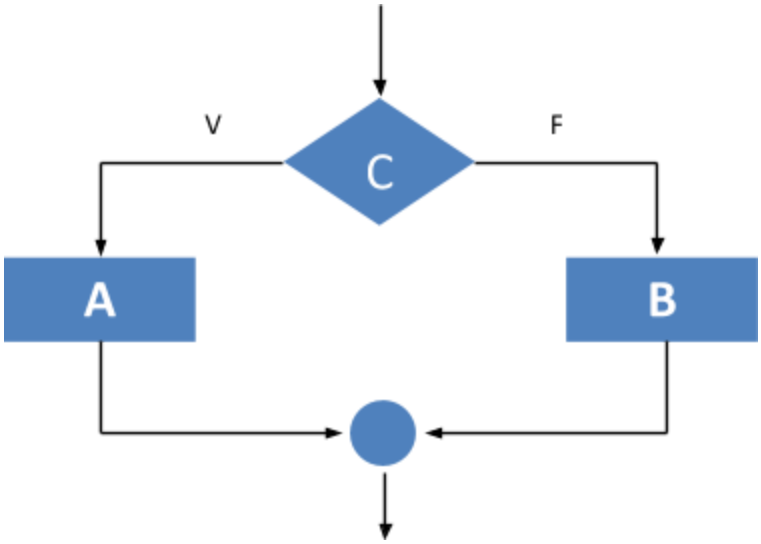
Indica que cada instrucción del programa será ejecutada una detrás de la otra en el orden en que aparecen en el código.



A y B pueden representar desde una instrucción simple hasta un programa completo.

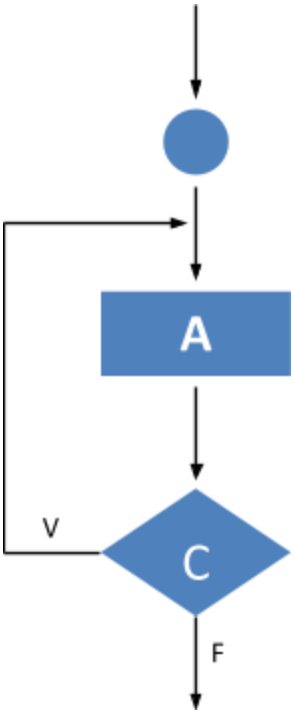
Estructura de selección

Permite elegir entre dos alternativas como resultado de la evaluación de una condición (verdadero-falso). Es la clásica instrucción IF incluida en todos los lenguajes de programación.



Estructura repetitiva

Ejecuta una instrucción repetidamente mientras se cumpla una condición dada.



Con esto en mente, realicemos un sencillo algoritmo para ejemplificar los conceptos con ayuda del lenguaje de programación Python. Python es un poderoso lenguaje de programación multipropósito que se ha abierto paso en el mundo hasta convertirse en el más utilizado de acuerdo al IEEE. Fue creado en 1991 por Guido Van Rossum. El lenguaje es multiplataforma y está disponible para sistemas MacOS, Linux y Windows, pertenece a la comunidad open source y no tiene costo.

Para poder desarrollar código en python, requerimos dos cosas: el intérprete y un editor de código.

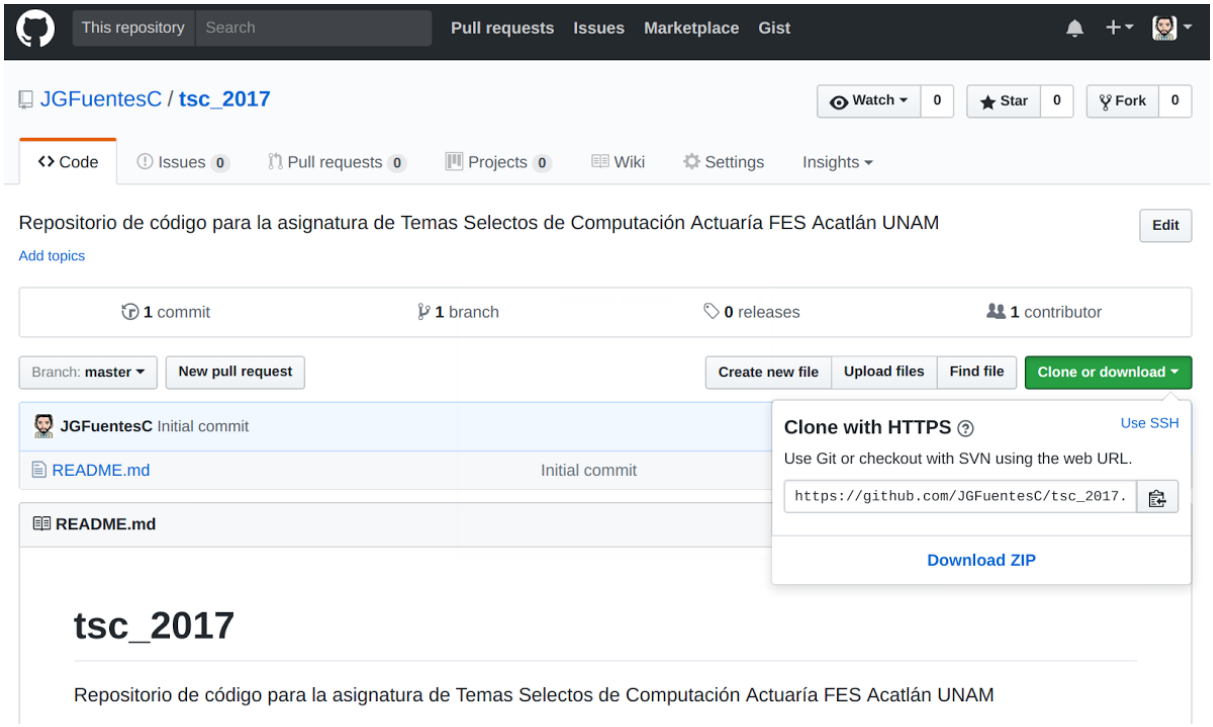
Para principiantes, se recomienda la instalación de la distribución **Anaconda** que contiene una gran variedad de paquetes preinstalados para hacer todo tipo de tareas además del intérprete y editor de código en Notebooks.

Puede descargarse (versión 2.7 por favor) aquí: https://www.continuum.io/downloads

En cuanto a un editor de código tenemos Pycharm https://www.jetbrains.com/pycharm/download/ una excelente opcion que incrementa nuestra productividad al escribir código.

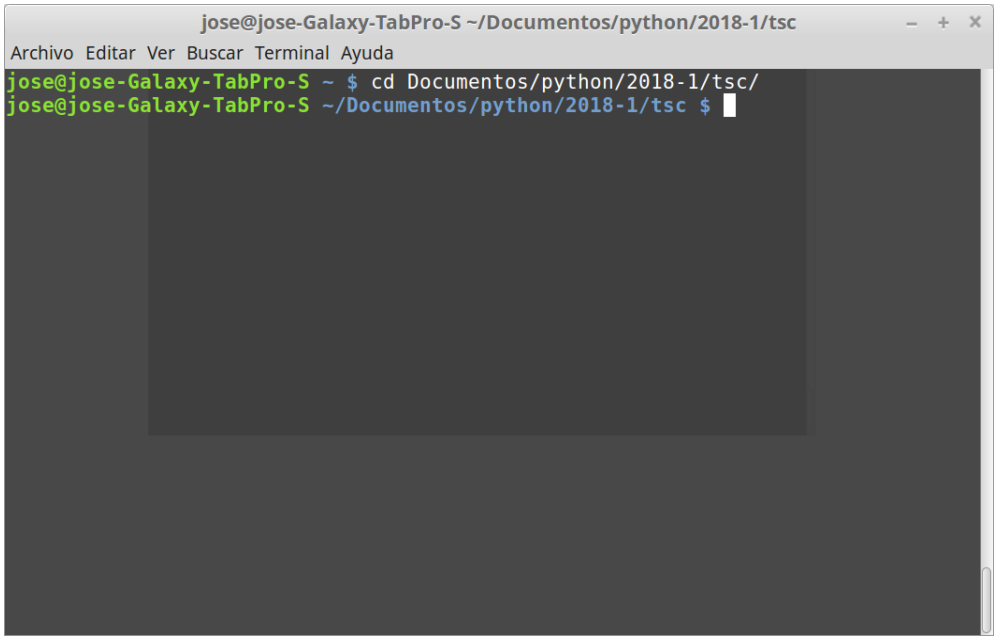
Una herramienta adicional que hará nuestro trabajo más fácil además de ser una tendencia a nivel mundial será el uso de **Git**, el cual es muy popular en el mundo del desarrollo de software. Como actuarios, git no será utilizado a profundidad, sin embargo, será una herramienta imprescindible para compartir código, prácticas, tener todos las mismas versiones y evitar compartir archivos por correo electrónico u otros medios de manera innecesaria. Git es un software de control de versionado escrito por Linus Torvalds (el creador del kernel de linux), para poder instalarlo, podemos seguir las instrucciones del sitio https://git-scm.com/download/. Se recomienda encarecidamente el **no usar Windows** ya que actualmente, prácticamente cualquier herramienta dentro del mismo puede ser sustituida en otro sistema operativo. Una manera más cómoda de gestión de repositorios git es GitHub(https://github.com/), puede ser vista como una “red social de código” hay que crear un perfil ahí para poder acceder a una cantidad inmensa de recursos públicos de código y otras cuestiones científicas. El repositorio que tendrá todo el contenido de nuestro curso está en la dirección https://github.com/JGFuentesC/tsc_2017.

Ahora mostraremos el proceso para clonar el repositorio, conforme avancemos en el curso, dicho repositorio será actualizado periódicamente y será cuestión únicamente de decirle a nuestro repositorio local que se sincronice con el repositorio remoto en la nube. Entramos al repositorio y vamos a la sección *clone or download:*



Copiamos la dirección que se nos presenta, abrimos una terminal en nuestro sistema operativo y cambiamos al directorio donde queramos ubicar nuestro repositorio:

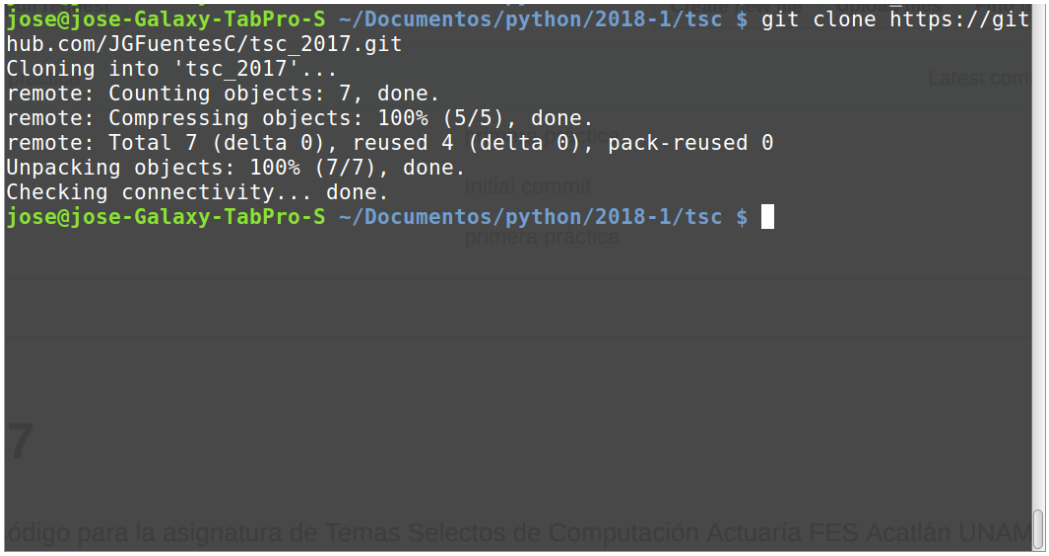
Documentos/python/2018-1/tsc/



Una vez ahí, escribimos en la terminal:

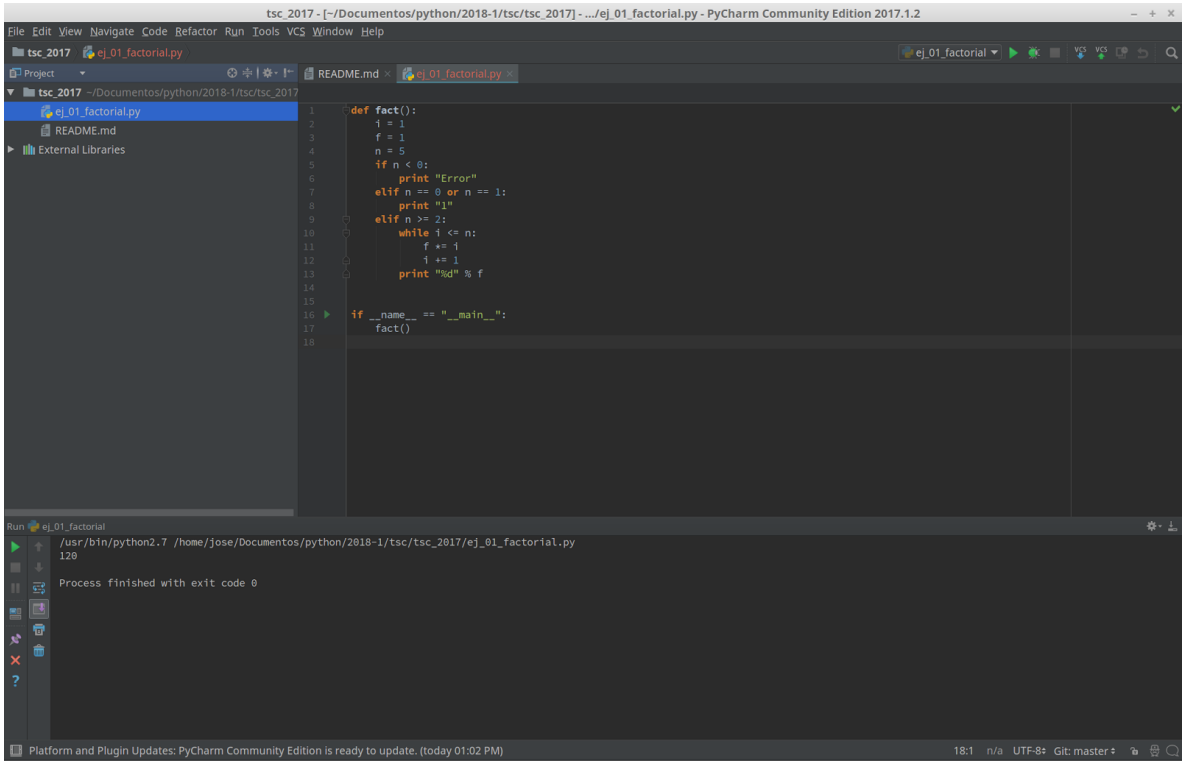
```
git clone https://github.com/JGFuentesC/tsc_2017.git
```

y listo, tendremos clonado el repositorio de la nube en nuestro equipo local:



Ahora, procedemos a abrir pycharm para poder editar y ejecutar los archivos del repositorio.

Para este primer ejemplo, tenemos un pequeño programa que calcula el factorial de un entero.



Se muestra el resultado de la ejecución del programa, el programa ha sido escrito de acuerdo al teorema de estructura, se deja al lector la investigación de la aplicación en código de estos conceptos.

Programación Orientada a Objetos (POO)

Si bien el paradigma estructurado nos permite solucionar la mayoría de los problemas de negocio con los que nos encontraremos en el mercado laboral, también es cierto que la POO nos amplía el espectro de lo que podremos hacer y nos encamina hacia una disciplina mucho más avanzada de desarrollo de software, lo que podría abrirnos oportunidades de trabajo en diversos campos de IT, como el desarrollo web o móvil debido a que en la actualidad el mercado demanda estos conocimientos además de ser un estándar industrial. La POO facilita muchísimo el proceso de desarrollo de software proporcionando modularidad, encapsulamiento y reusabilidad del código.

El concepto fundamental de la POO es la ***CLASE***, una clase es una abstracción de la realidad, su unidad fundamental es el ***OBJETO***. Una clase puede ser vista como una colección abstracta de objetos (una especie de plantilla); un objeto es prácticamente cualquier elemento identificable dentro de los requerimientos de nuestro problema en cuestión. Todo objeto debe cumplir las siguientes características:

- Tiene propiedades definibles
- Posee comportamientos asociados (métodos)
- Pueden comunicarse con otros objetos mediante sus métodos.

Realicemos un ejemplo sencillo, consideremos el objeto vector de R^n e identifiquemos sus propiedades y métodos:

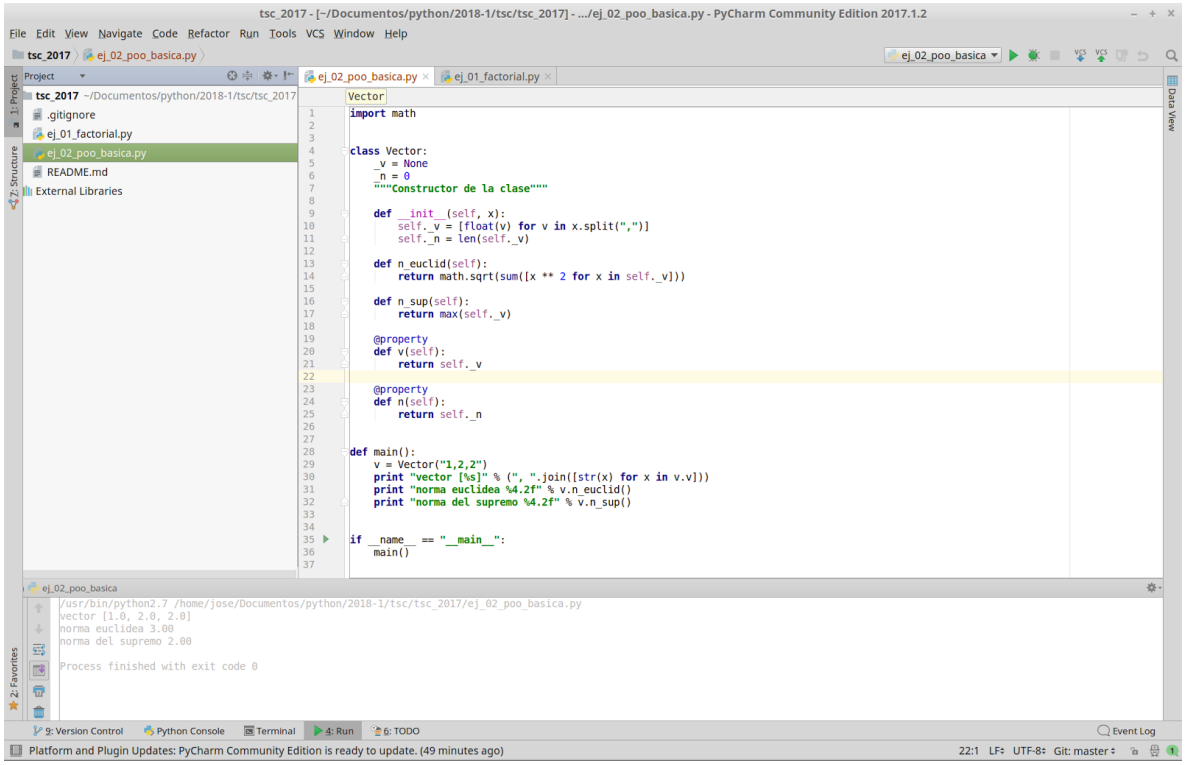
Propiedades

- Dimensión (tamaño) del vector
- Elementos del vector

Métodos

- Norma euclídea
- Norma del supremo

Aquí observamos que los métodos corresponden a cálculos (acciones) a realizar con los vectores mientras que las propiedades nos servirán para caracterizar a cada objeto de la clase vector. Para poder programar OO en Python, utilizaremos la palabra reservada `class`. Pongámoslo en práctica:



Vemos que hemos conseguido con éxito encapsular todo el código referente al vector dejando una cantidad mínima de código en nuestra función principal. Nótese que hemos creado muy fácilmente un objeto de la clase vector y no tuvimos que definirlo por separado ya que al definir la clase hemos agrupado todos los posibles vectores que pudiésemos utilizar en nuestro programa.

Bases de Datos

En el mercado laboral, la programación nos representa una ventaja competitiva en cuanto a la posibilidad de poder automatizar tareas, reducir procesos que manualmente tardarían horas a minutos o generar herramientas de software que faciliten nuestro trabajo. Por otra parte, las bases de datos son definitivamente la materia prima de prácticamente todos los ámbitos laborales del actuario; sean seguros, finanzas, estadística, banca, etc. Las bases de datos jugarán un papel crucial en su vida laboral, es por ello que es conveniente precisar conceptos básicos toda vez que estos determinarán la optimalidad en la explotación de la información.

Comencemos por definir Base de Datos. Una base de datos es una colección de datos informativos relacionados entre sí y organizados en un mismo contexto para su explotación. En términos de cómputo, tendremos un software que fungirá como administrador de dichos datos y sus relaciones permitiéndonos acceder rápidamente a cualquier información que necesitemos. Los sistemas informáticos que cumplen dicha función son llamados RDBMS (acrónimo inglés de Sistema Gestor

de Bases de Datos Relacionales). Existen abundantes fabricantes y productos disponibles en el mercado tanto propietarios como de código abierto, por ejemplo:

- Oracle (Oracle Corp.)
- MySQL(Oracle Corp.)
- SQL Server (Microsoft)
- SQLite (Libre)
- PostgreSQL (libre)

Cada uno cuenta con características estándar y particularidades del fabricante como podrían ser: Motores de almacenamiento, lenguaje de consulta ampliado, herramientas de gestión, herramientas de respaldo, etc.

Los RDBMS más populares en grandes industrias son Oracle y SQL Server ambos cuentan con versiones gratuitas en las que podemos practicar sin restricción y así poder prepararnos para el trabajo.

http://www.oracle.com/technetwork/database/database-technologies/express-edition/overview/index.html

https://www.microsoft.com/en/server-cloud/products/sql-server-editions/sql-server-express.aspx

En el caso de startups, PyMEs y proyectos independientes se utiliza con frecuencia MySQL y PostgreSQL. SQLite es muy popular en desarrollo de aplicaciones móviles. Las ligas de descarga se muestran a continuación, todos son gratuitos:

https://www.sqlite.org/download.html

https://dev.mysql.com/downloads/mysql/

https://www.postgresql.org/download/

Una vez que elegimos el RDBMS donde trabajaremos, revisaremos algunos objetos básicos.

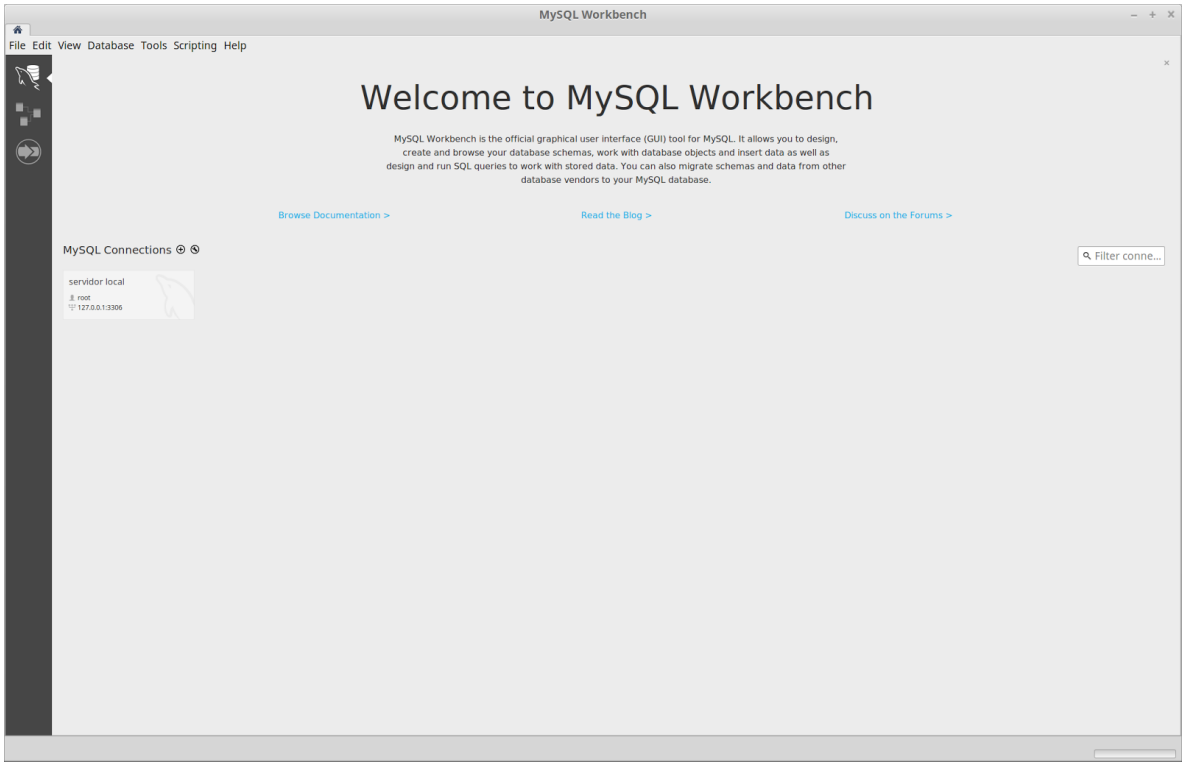
- *Servidor*: Unidad de cómputo donde reside nuestro RDBMS
- *Instancia*: conjunto de capacidades de memoria y procesamientos reservados en el servidor para la ejecución del RDBMS
- *Base de Datos*: Objeto maestro donde se almacenará la información correspondiente a tablas, vistas, procedimientos almacenados, disparadores, funciones, etc.
- *Tabla*: Objeto cuya función es el almacenamiento de datos en forma de gradilla. Cada columna es llamada Campo mientras que cada fila se conoce como registro
- *Vista*: Es una tabla virtual, los datos en ella no se encuentran almacenados en una base de datos, simplemente son una consulta almacenada como objeto

- *Procedimiento almacenado*: Conjunto de instrucciones almacenadas en la base de datos que permite encapsular tareas repetitivas
- *Disparador*: Es un tipo de procedimiento almacenado que se ejecuta al intentar modificar los datos dentro de una tabla
- *Función*: Conjunto de sentencias que retornan un valor de salida
- *SQL*: Acrónimo inglés de lenguaje de consulta estructurado, es un lenguaje de programación de propósito específico para manipular datos almacenados en un RDBMS

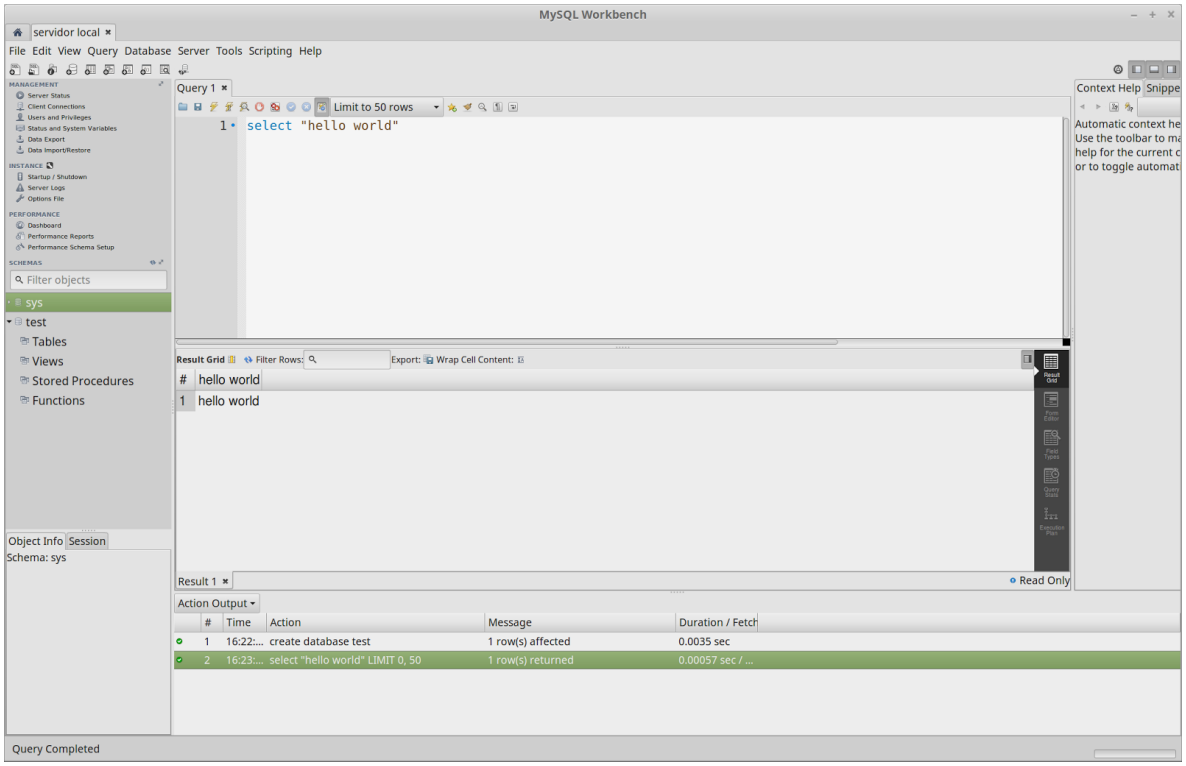
Los disparadores no serán comúnmente utilizados en la práctica actuarial.

Utilicemos como ejemplo la plataforma de MySQL Community Server y mostremos el entorno básico de trabajo.

La herramienta de explotación de datos será **MySQL Workbench**, que podemos descargar desde aquí: https://dev.mysql.com/downloads/workbench/ arrancamos la aplicación:



Seleccionaremos el servidor e instancia a la cual nos conectaremos y proporcionaremos nuestra información para autenticarnos. En el servidor de este ejemplo existe una base de datos llamada test que no contiene todavía objetos, sin embargo, podemos apreciar su contenido en el explorador. Asimismo, hemos abierto una hoja de trabajo SQL al habernos conectado a la instancia. Por último, hemos escrito y ejecutado una sentencia SQL.



En el siguiente tema utilizaremos más a fondo esta herramienta, además de generar abundante código para poner en práctica y refrescar lo visto en la materia de Bases de Datos.