Sesión 2: Manejo y uso de bases de datos

Carlos Ugarte

29.08.2025

1 Introducción: ¿por qué bases de datos?

Por favor, entren al software PyCharm CE y repitan el proceso visto en la anterior clase hasta que su directorio de trabajo sea la carpeta clonada de este taller.

Las bases de datos son nada más que un conjunto de datos estructurado y almacenado sistemáticamente con el objetivo de acceder y manipularlos posteriormente. En el ambiente informático, las bases de datos son el elemento base desde el cual se parte hoy en día y la noción tiene una aplicación en la práctica de diversas disciplinas, especialmente en las que se necesite gestionar datos, tanto más cuanto más voluminosos sean. Una base de datos, y he ahí su naturaleza general, puede ser formada con cualquier tipo de datos: datos puramente espaciales (geometrías, etc.), datos numéricos, alfanuméricos, lingüísticos, etc.

Los dos principios vitales para constituir una base de datos son: estructura y sistematicidad. Con tales principios, es que se facilita la utilización de dichos datos. Esto se vuelve de especial importancia cuando las bases de datos alcanzan un ámbito más allá de lo personal, y las prácticas más habituales basadas en una gestión "manual" de un conjunto de ficheros no son una opción adecuada. La solución para lograr esa necesaria gestión centralizada de los datos es el uso de bases de datos y también, como veremos más adelante, los sistemas gestores de bases de datos, que representan la interfaz entre las bases de datos y los distintos usuarios.

Las bases de datos pueden tener distintas estructuras, es decir, métodos para organizar y almacenar la data en la memoria de la computadora. Las estructuras definen la manera en la cual los elementos de los datos se encuentran relacionados entre sí y cómo se pueden realizar operaciones sobre ellos. Las estructuras más comunes son las siguientes:

- Matrices: Bloques contiguos de memoria que almacenan elementos del mismo tipo de datos, a los que se accede mediante índice.
- Listas enlazadas: Colecciones de nodos, donde cada nodo contiene datos y un puntero al siguiente nodo.
- Pilas: Estructuras de datos lineales que siguen el principio de "último en entrar, primero en salir" (LIFO).
- Colas: Estructuras de datos lineales que siguen el principio de "primero en entrar, primero en salir" (FIFO).
- Árboles: Estructuras de datos jerárquicas donde los datos se organizan en nodos conectados por aristas, como árboles binarios o árboles balanceados.
- Gráficos: Estructuras de datos no lineales que constan de nodos (vértices) y conexiones (aristas) entre ellos.

- Tablas hash: Estructuras de datos que asignan claves a valores para una recuperación de datos eficiente.

Muchas veces, las bases de datos no son una estructura de datos simple (como una pila o lista enlazada), sino más bien un conjunto de datos estructurado que combina varias estructuras y varios tipos de estructuras.

Este escalonado armado de una base de datos usualmente involucra algunos sencillos scripts de Python que permiten relacionar toda la data que tenemos de una manera sistemática, estructurada y transparente. En la siguiente sección veremos dónde almacenar nuestros datos de forma que sea sencillo colaborar con colegas y que otras personas puedan reproducir nuestros experimentos para testear si llegan a los mismos resultados.

2 Colaborando con Github

Ahora, escriban desde la terminal el siguiente comando git:
git pull origin main

GitHub es una plataforma en la nube donde uno puede almacenar, compartir y colaborar con otros para escribir código. Las ventajas de almacenar nuestro código ahí son:

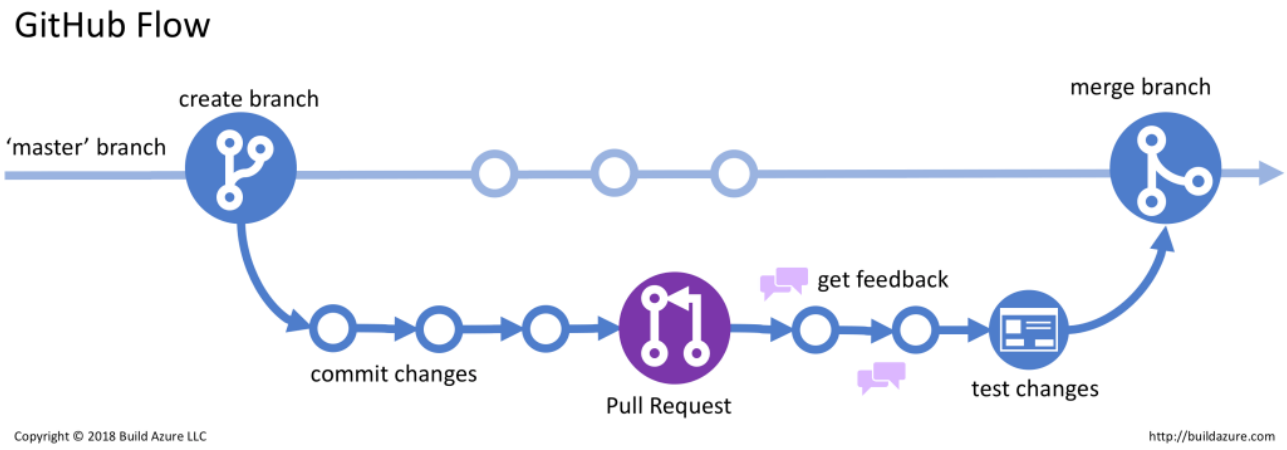
- Exhibir o compartir tu trabajo.
- Seguir y gestionar los cambios en tu código a lo largo del tiempo.
- Dejar que otros revisen tu código y hagan sugerencias para mejorarlo.
- Colaborar en un proyecto compartido sin preocuparte de que tus cambios afecten el trabajo de tus colaboradores antes de que estés listo para integrarlos.
- El trabajo colaborativo, una de las características fundamentales de GitHub, es posible gracias al software de código abierto Git, sobre el que se basa.

2.1 ¿Qué es entonces Git?

Git es un sistema de control de versiones que rastrea inteligentemente los cambios en los archivos. Git es particularmente útil cuando tú y un grupo de personas realizan cambios en los mismos archivos simultáneamente.

Para hacer esto en un flujo de trabajo basado en Git, normalmente:

- Crea una rama a partir de la copia principal de los archivos en los que tú (y tus colaboradores) están trabajando.
- Edita los archivos de forma independiente y segura en tu propia rama personal.
- Deja que Git integre inteligentemente tus cambios específicos en la copia principal de los archivos, para que no afecten las actualizaciones de otros.
- Deja que Git registre tus cambios y los de los demás, para que todos sigan trabajando en la versión más actualizada del proyecto.



2.2 ¿Cómo interactúan Git y Github juntos?

Visiten https://github.com y creen una cuenta.

Al subir archivos a GitHub, los almacenas en un repositorio de Git. Esto significa que, al realizar cambios (o confirmar cambios) en tus archivos en GitHub, Git comenzará a rastrear y gestionar automáticamente tus cambios.

Hay muchas acciones relacionadas con Git que puedes realizar en GitHub directamente en tu navegador, como crear un repositorio de Git, crear ramas y subir y editar archivos.

Sin embargo, la mayoría de las personas trabajan con sus archivos localmente (en su propio ordenador) y luego sincronizan continuamente estos cambios locales (y todos los datos de Git relacionados) con el repositorio remoto central de GitHub. Existen muchas herramientas que puedes usar para ello, como GitHub Desktop.

Una vez que empieces a colaborar con otros y todos necesiten trabajar en el mismo repositorio simultáneamente, continuamente:

Obtendrás los últimos cambios realizados por tus colaboradores desde el repositorio remoto de GitHub. Reenvía tus propios cambios al mismo repositorio remoto en GitHub. Git descubre cómo fusionar inteligentemente este flujo de cambios y te ayuda a gestionarlo mediante funciones como las solicitudes de incorporación de cambios.

3 Bases de datos lingüísticas disponibles

Para distintos aspectos de la gramática de las lenguas del mundo:

Grambank

Para listas de conceptos lingüísticos (vincula conjuntos de conceptos estandarizados con glosas de listas de palabras (etiquetas) de varias fuentes para ayudar a los investigadores a comparar idiomas):

Concepticon

Base de datos bibliográfica y de clasificación de las lenguas del mundo, con acceso abierto a materiales lingüísticos y afiliaciones lingüísticas. Ofrece una amplia colección de recursos, incluyendo gramáticas, artículos y diccionarios, junto con información detallada sobre las familias lingüísticas y sus relaciones:

Glottolog

Base de datos de colexificaciones a través de las lenguas del mundo. Este recurso proporciona estructura y datos para que los investigadores estudien las asociaciones semánticas, los significados de las palabras, la evolución del lenguaje y la conceptualización humana en diversas lenguas del mundo:

CLICS

Database of Cross-Linguistic Norms, Ratings, and Relations, recurso que selecciona múltiples conjuntos de datos que contienen información sobre diversas propiedades de palabras y concepts. Las propiedades que ofrece la base de datos abarcan desde variables recopiladas automáticamente, como frecuencias de palabras (normas), hasta estudios psicolingüísticos con participantes humanos (calificaciones) y datos comparativos dentro de cada lengua o entre ellas (relaciones). Esto permite a los investigadores comparar la misma variable (p. ej., calificaciones de la edad de adquisición) entre lenguas o utilizar diferentes variables para investigar su pregunta de investigación específica:

NoRaRe

South American Indigenous Language Structures (SAILS) es una gran base de datos de propiedades gramaticales de idiomas recopilada a partir de materiales descriptivos (como gramáticas de referencia):

SAILS

World Atlas of Language Structures (WALS) es una base de datos de propiedades estructurales de muchas lenguas del mundo. Contiene datos sobre características fonológicas, gramaticales y léxicas, recopilados por expertos lingüísticos a partir de materiales descriptivos existentes:

WALS

Dataset de muestras genéticas poblacionales cotejadas con bases de datos de diversidad lingüística. Cada población genética está asociada a la lengua principal hablada por su población:

GeLaTo

Automated Similarity Judgment Program (ASJP) contiene listas de 40 palabras de todos los idiomas del mundo. Se puede obtener una distancia léxica comparando las listas, lo cual resulta útil, por ejemplo, para clasificar un grupo lingüístico e inferir su edad de divergencia.

ASJP

4 Extrayendo data de bases de datos

Ahora seguimos con extracción de datos de una base de datos. Para esto incorporamos pasos realizados similar a cuando clonamos el repositorio del taller en nuestras computadoras. Los pasos son:

1. Crear un directorio dentro de nuestro folder de proyectos.
2. Movernos a la nueva carpeta.
3. Clonar la carpeta con la data del repositorio de Github respectivo a la base de datos que queramos usar.

5 Actividades para la casa

Actividad para la casa correspondiente a la segunda sesión: Sube tu proyecto a Github

1. Crea un repositorio a Github llamado 'mi_proyecto'.
2. Linkea ese repositorio remoto a tu repositorio local creado para la actividad para casa anterior.
3. Crea una branch titulada 'colexificaciones' y otra titulada 'filogenia'.

4. Una vez terminado, crea un issue en el repositorio del taller del curso en mi perfil colocando tu nombre como nombre del issue y el link de tu proyecto en la descripción.