

UNIVERSIDAD GERARDO BARRIOS.
FACULTAD DE CIENCIA Y TECNOLOGÍA.
INGENIERÍA EN SISTEMAS Y REDES INFORMÁTICAS.



CÁTEDRA
PROGRAMACIÓN COMPUTACIONAL III

CATEDRÁTICO:
ING. WILLIAM ALEXIS MONTES GIRÓN.

ACTIVIDAD:
PARCIAL2_C2

INTEGRANTES:
FRANCISCA DEL CARMEN BONILLA ARGUETA. SMSS020724
EDGAR ARNOLDO CAMPILLO ORELLANA. SMTR016224
MADELINE BRUNELLA MEJÍA MEJÍA. SMSS063924
JOSUE ALEXANDER TURCIOS QUINTANILLA. SMIS948820
FREDDY SAMUEL VÁSQUEZ ÁLVAREZ. SMTR056221

FECHA:
19/10/2025

INTRODUCCIÓN

En el presente Trabajo de investigación presentamos una librería de Python llamada Faker, la cual se utiliza para generar datos falsos pero realistas de forma automatizada. Esta herramienta es ampliamente empleada por desarrolladores y analistas para realizar pruebas, llenar bases de datos o crear simulaciones sin necesidad de usar información real.

Faker es una librería externa que permite crear información simulada como nombres, direcciones, correos electrónicos, números de teléfono, fechas de nacimiento o datos de empresas. Cada dato generado parece auténtico, pero no pertenece a ninguna persona u organización real, lo que garantiza la seguridad y confidencialidad en los entornos de prueba.

Su objetivo principal es facilitar el desarrollo y la verificación de sistemas informáticos que necesitan manejar grandes volúmenes de información. Al generar datos ficticios, los programadores pueden comprobar el funcionamiento de sus aplicaciones, validar formularios, probar consultas en bases de datos y realizar análisis sin poner en riesgo información sensible.

En muchos proyectos, trabajar con datos reales puede violar políticas de privacidad o ser difícil de obtener. Faker resuelve este problema proporcionando datos verosímiles de manera instantánea, contribuyendo así a un proceso de desarrollo más ágil, seguro y ético.

La librería opera mediante funciones sencillas que crean distintos tipos de datos. Estas funciones pueden combinarse para construir registros completos que simulen bases de datos reales, permitiendo así realizar pruebas o demostraciones con resultados coherentes. En cuanto a su aplicación en la vida real, Faker se ha utilizado en empresas de tecnología, instituciones académicas y proyectos de investigación para simular datos de usuarios, clientes o transacciones. Además, es una herramienta clave en pruebas de software, desarrollo web, inteligencia artificial y ciencia de datos, donde la disponibilidad de información sintética y segura es fundamental para el éxito de los proyectos.

INDICE

Introducción	2
Faker	4
¿En qué consiste la librería y para qué se usa?	4
Funciones más relevantes y utilizadas.	5
Casos de uso en los que la librería se ha utilizado en la vida real	6
Bibliografía	9

FAKER

¿En qué consiste la librería y para qué se usa?

La librería Faker (también reconocida como “fake-data generator” para Python) es un paquete de código abierto cuyo propósito es generar datos “falsos” o de prueba de forma automática y realista para distintos escenarios de desarrollo de software, análisis de datos o pruebas.

En concreto, permite crear nombres, direcciones, correos electrónicos, textos, coordenadas geográficas, entre otros tipos de datos, de modo que los desarrolladores o científicos de datos puedan “rellenar” bases de datos, probar sistemas o anonimizar datos reales sin depender de datos reales sensibles.

Entre sus usos típicos figuran:

- Generar datos sintéticos para poblar bases de datos en entornos de desarrollo.
- Crear “fixtures” de prueba o datos ficticios para testing de aplicaciones.
- Simular entornos de producción o prototipos sin necesidad de utilizar datos reales personales, ayudando así a proteger la privacidad.
- Facilitar el análisis, prototipado o la demostración de productos (por ejemplo, para mostrar un dashboard completo sin tener aún todos los datos reales).

En resumen, Faker es una herramienta versátil que permite automatizar la creación de datos ficticios con mínima configuración, lo que acelera tareas de desarrollo, prueba y datos.

Funciones más relevantes y utilizadas.

A continuación presentamos algunas funciones (“métodos” o “providers”) de Faker que se consideran de uso frecuente, con su descripción y cómo pueden emplearse:

1. `fake.name()` — Genera un nombre completo aleatorio (por ejemplo: “Lucy Cechtelar”).
2. `fake.address()` — Genera una dirección ficticia, incluyendo posiblemente calle, ciudad, código postal.
3. `fake.email()` — Genera una dirección de correo electrónico aleatoria, útil para poblar campos de email o login ficticio.
4. `fake.text()` — Genera un bloque de texto aleatorio (por ejemplo para la descripción de un usuario, “bio”, comentarios, etc.).
5. `fake.latitude()` / `fake.longitude()` — Generan coordenadas geográficas aleatorias; útiles para poblar campos de ubicación geoespacial.
6. `fake.unique.<provider>()` — Esta propiedad permite garantizar que los valores generados sean únicos (por ejemplo para nombres de usuario únicos).
7. Métodos especializados en `faker.providers.python.Provider`, tales como `pybool()`, `pydecimal()`, `pydict()`, `pystr()` — Ayudan a generar valores booleanos aleatorios, decimales, diccionarios y cadenas aleatorias, respectivamente.

Estas funciones muestran la amplitud de Faker: desde datos personales (nombres, direcciones) hasta tipos más genéricos (diccionarios, textos, números) que permiten un alto grado de personalización y flexibilidad en la generación de datos.

Casos de uso en los que la librería se ha utilizado en la vida real

Aquí exponemos varios escenarios reales (o típicos) donde Faker puede y ha sido empleada:

Poblar bases de datos de desarrollo: Cuando un equipo de desarrollo necesita probar la interfaz, consultas, filtros, rendimiento o comportamiento de la aplicación sin disponer aún de datos reales, Faker puede generar cientos o miles de usuarios ficticios, con nombre, correo, ubicación, bio, etc., y permitir validar funcionalidades antes del despliegue.

Pruebas de rendimiento, estrés y análisis de datos: En entornos donde se requiere simular carga o volumen de datos (por ejemplo, para testing de consultas complejas, paginación, indexación), Faker facilita la generación rápida de datos volumétricos.

Anonimización de datos: Si una organización dispone de datos reales sensibles, pero desea trabajar con ellos en entornos menos protegidos, Faker puede generar datos sustitutos que respeten la estructura del original, evitando exposición de información real del usuario.

Prototipado y demostración de productos: Para mostrar un producto, dashboard, app o API que requiere datos consistentes (clientes, ubicaciones, transacciones), Faker permite generar datos coherentes con poco esfuerzo para crear una experiencia “realista” al usuario o cliente de la demo.

Generación de datasets sintéticos para Machine Learning o Data Science: Como parte del pre procesamiento, aprendizaje o pruebas de modelos, se pueden usar datos generados con Faker para simular datos de entrenamiento o para cubrir casos poco frecuentes (aunque con la salvedad de que los datos generados no reemplazan datos reales de calidad completa).

4. Comentarios adicionales (ventajas, limitaciones y consideraciones)

Ventajas:

- Es muy fácil de instalar y comenzar a utilizar (pip install Faker).
- Amplia variedad de “providers” (generadores de datos) y soporte de localización (múltiples idiomas y formatos regionales).
- Permite reproducibilidad a través del “seed” del generador, lo que resulta útil en pruebas donde se desea consistencia.

Limitaciones:

- Si bien genera datos realistas en formato, no garantiza que sean semánticamente muy coherentes (por ejemplo, un nombre puede no coincidir con un dominio de correo generado, etc.).
- No está pensado para generar datos sensibles reales ni para producción de datos con calidad de “datos del mundo real” más allá de la prueba o prototipado.

Consideraciones:

- Es recomendable usar Faker sólo en entornos de desarrollo o pruebas, y separar dependencias para que no quede en producción si no es necesario.
- Al generar datos “únicos” (via fake.unique) hay que tener cuidado con duplicados eventuales o agotamiento del espacio de valores.
- Cuando se trabaja con localizaciones (idiomas/regiones), conviene verificar si los “providers” relevantes existen o están actualizados para ese locale.

Ejemplos:

#Codigo

#Ejemplo sencillo

Ejercicio: Generador de datos falsos de Japón

Este programa usa la librería Faker para crear nombres, correos y direcciones japonesas

#Importacion de la libreria from faker

import Faker fake = Faker('ja_JP')

for i in range(4):

print("Nombre:", fake.name())

print("Correo:", fake.email())

print("Dirección:",fake.address())

print("-----")

BIBLIOGRAFÍA

Coding nomads // Generación de datos falsos con Python Faker.

Datacamp Tutorial de creación de datos sintéticos con Python Faker // Python faker. - 10 de Agosto. 2022.

Farsante // Genere cantidades masivas de datos falsos (pero realistas) para pruebas y desarrollo.. - 2022.

Geeksforgeeks Biblioteca de falsificadores de Python [Journal]. - 24 Nov, 2024.

Jeremy Yu Medium // Cree conjuntos de datos sintéticos fácilmente con Python Faker y Google Colab. - 29 de 2 de 2024.