

Análisis comparativo de Streamlit y Bokeh para el desarrollo de dashboards interactivos en Python: características, personalización y facilidad de implementación.

Comparative Analysis of Streamlit and Bokeh for Interactive Dashboard Development in Python: Features, Customization, and Ease of Implementation

-Fecha de recepción: 08-04-2026-Fecha de aceptación: 23-06-2026-Fecha de publicación: 06-07-2026

Stefany Antonella Campaña Lozano¹
Universidad Técnica Particular de Loja (UTPL), Loja, Ecuador
antonellacampa6@gmail.com

Resumen

El presente artículo tiene como objetivo comparar Streamlit y Bokeh para el desarrollo de dashboards interactivos en Python, considerando criterios de apariencia, actualización durante el desarrollo y facilidad de implementación. La investigación se fundamenta en un análisis documental y comparativo, complementado con la implementación de dos prototipos. Los resultados evidencian que Streamlit facilita el desarrollo mediante componentes predefinidos, mientras que Bokeh ofrece mayores posibilidades de personalización, aunque requiere una implementación más elaborada en el caso estudiado. Se concluye que ambas herramientas presentan ventajas y limitaciones, por lo que su elección depende de los requerimientos y características de cada proyecto.

Palabras clave: dashboards interactivos, Python, Streamlit, Bokeh, visualización de datos.

Abstract

This article aims to compare Streamlit and Bokeh for the development of interactive dashboards in Python, considering criteria related to appearance, updates during development, and ease of

implementation. The research is based on a documentary and comparative analysis, complemented by the implementation of two prototypes. The results show that Streamlit facilitates development through predefined components, while Bokeh offers greater customization possibilities, although it required a more elaborate implementation in the case studied. It is concluded that both tools present advantages and limitations; therefore, their selection depends on the requirements and characteristics of each project.

Keywords: interactive dashboards, Python, Streamlit, Bokeh, data visualization.

Introducción

El incremento en la generación y disponibilidad de datos ha impulsado la necesidad de utilizar herramientas que faciliten su procesamiento, análisis y representación visual. En este contexto, los dashboards interactivos constituyen una alternativa para organizar y presentar información mediante gráficos, indicadores y componentes visuales que favorecen la interpretación de los datos y apoyan los procesos de toma de decisiones. Guerrero González (2021) destaca la relación entre el crecimiento de la información y el desarrollo de herramientas orientadas a su visualización.

El lenguaje de programación Python ofrece diversas herramientas para el procesamiento y la visualización de datos, entre las que se encuentran Streamlit y Bokeh. Estas permiten desarrollar soluciones interactivas mediante enfoques diferentes: Streamlit facilita la construcción de aplicaciones de datos mediante componentes predefinidos, mientras que Bokeh se orienta a la creación de visualizaciones interactivas con posibilidades de personalización e integración en aplicaciones web.

Sin embargo, la disponibilidad de distintas herramientas plantea la necesidad de identificar cuáles son sus características, ventajas y limitaciones durante el desarrollo de dashboards interactivos. La elección de una herramienta no depende únicamente de sus funcionalidades, sino también de aspectos relacionados con la personalización de la interfaz, el comportamiento de actualización durante el desarrollo y las facilidades de implementación. Por ello, resulta pertinente establecer criterios comparativos que permitan reconocer las particularidades de cada alternativa.

En este contexto, el presente artículo tiene como objetivo analizar comparativamente Streamlit y Bokeh para el desarrollo de dashboards interactivos en Python, a partir de sus características técnicas y de los resultados obtenidos mediante la implementación de dos prototipos. El análisis considera criterios relacionados con la apariencia, el comportamiento de actualización durante el desarrollo y las facilidades de implementación, con la finalidad de identificar las ventajas y limitaciones observadas en cada herramienta.

La investigación retoma los resultados del estudio desarrollado por Campaña Lozano (2022) y los presenta desde una perspectiva comparativa, con el propósito de aportar elementos que orienten la selección de herramientas para la construcción de dashboards interactivos en Python.

Materiales y Métodos

2.1. Marco teórico

2.1.1. ¿Qué es un Dashboard?

Los dashboards son herramientas de visualización que permiten presentar información relevante mediante gráficos, tablas e indicadores organizados en una interfaz. Según Few (2006), su finalidad es facilitar el seguimiento y la interpretación de información relacionada con determinados objetivos.

Los dashboards interactivos incorporan funcionalidades como filtros, selección de variables y actualización de visualizaciones, permitiendo explorar los datos de acuerdo con las necesidades del usuario. Estas características favorecen el análisis de información y requieren herramientas que faciliten tanto la representación gráfica como la interacción con los datos.

2.1.2. Bokeh

“Bokeh es una biblioteca de Python orientada a la creación de visualizaciones interactivas para navegadores web (Jurić, 2019). Permite desarrollar gráficos personalizables e incorporar herramientas de interacción, como selección de datos, desplazamiento y modificación de parámetros.

Entre sus principales características se encuentran la personalización de elementos gráficos, la integración con bibliotecas de procesamiento de datos y la posibilidad de desarrollar aplicaciones

interactivas mediante Bokeh Server. Estas funcionalidades permiten adaptar las visualizaciones a diferentes necesidades, aunque el nivel de complejidad de la implementación depende de las características del proyecto.

2.1.3. Streamlit

Streamlit es un framework de código abierto basado en Python que facilita el desarrollo de aplicaciones web interactivas orientadas al análisis y la visualización de datos (Streamlit Inc., 2022). Su funcionamiento se basa en la utilización de componentes predefinidos que permiten construir interfaces sin requerir necesariamente conocimientos especializados de desarrollo web.

Entre sus principales características destacan la integración con bibliotecas de Python, la incorporación de componentes interactivos y la actualización de la aplicación durante el proceso de desarrollo. Estas funcionalidades favorecen la creación de prototipos y aplicaciones de visualización de datos mediante una estructura de programación accesible.

2.2. Metodología

El presente artículo está fundamentado en la investigación desarrollada por Campaña Lozano (2022), titulada Implementación de dashboards interactivos para la visualización de datos usando lenguaje de programación Python, realizada en la Universidad Técnica Particular de Loja. A partir de dicho estudio, se sintetizan y analizan los resultados obtenidos mediante la implementación de dos prototipos desarrollados con Streamlit y Bokeh, considerando criterios relacionados con la apariencia, el comportamiento de actualización durante el desarrollo y las facilidades de implementación. Esta adaptación tiene como finalidad presentar los principales hallazgos de la investigación original en un formato de artículo científico.

2.2.1. Diseño de la investigación

La investigación se desarrolló mediante un enfoque analítico-documental y comparativo, basado en la revisión de información técnica sobre Streamlit y Bokeh y en el análisis de dos prototipos implementados con estas herramientas. La comparación se realizó considerando sus características generales y la experiencia obtenida durante el desarrollo de los prototipos, a partir

de criterios relacionados con la apariencia, la actualización durante el desarrollo y las facilidades de implementación.

2.2.2. Desarrollo de los prototipos

Como parte de la investigación original, se implementaron dos dashboards interactivos utilizando Python. El prototipo desarrollado con Streamlit empleó un conjunto de datos en formato CSV relacionado con publicaciones en Twitter y el análisis de sentimientos sobre aerolíneas estadounidenses. Por otra parte, el prototipo elaborado con Bokeh utilizó información de instituciones educativas de Ecuador almacenada en formato XLS.

La implementación permitió examinar las funcionalidades de ambas herramientas, los componentes utilizados para la representación de datos y las particularidades de sus procesos de desarrollo. Debido a que los prototipos utilizaron conjuntos de datos y funcionalidades diferentes, la comparación se orientó a identificar características y experiencias de implementación, sin establecer equivalencias directas de rendimiento computacional.

2.2.3. Criterios de comparación

Los criterios de evaluación se establecieron a partir de la revisión documental y de la experiencia obtenida durante la implementación de los prototipos. Para organizar el análisis se consideraron como referencia los atributos de calidad del software abordados por Bass et al. (2012), adaptándolos a las características observadas durante el desarrollo de dashboards interactivos.

La evaluación se estructuró en tres categorías:

- **Apariencia:** comprende las posibilidades de personalización de la interfaz y de las visualizaciones, así como la disponibilidad de componentes predefinidos. Esta categoría se relaciona con determinados aspectos de la usabilidad, particularmente con la presentación de información y la configuración de elementos visuales.
- **Actualización durante el desarrollo:** examina el comportamiento de los prototipos al modificar el código en un entorno local y los mecanismos necesarios para visualizar los cambios. Este criterio no constituye una medición del rendimiento computacional.

- **Facilidades de implementación:** considera la experiencia del desarrollador, los recursos de aprendizaje y la posibilidad de integrar otros lenguajes de programación durante la construcción de los dashboards.

Los criterios establecidos se presentan en la Tabla 1.

Tabla 1.

Categorías de comparación

Categoría	Criterio
Apariencia	Personalización de la interfaz y los gráficos; componentes predefinidos.
Rendimiento	Actualización automática o manual de los cambios realizados en el código.
Facilidad para el desarrollo	Experiencia del desarrollador y curva de aprendizaje; integración con otros lenguajes de programación.

Nota. Adaptado de los criterios establecidos en Campaña Lozano (2022).

2.2.4. Procedimiento de análisis

El análisis se desarrolló en dos etapas. En la primera se revisaron las características generales de Streamlit y Bokeh, considerando aspectos como compatibilidad, instalación, disponibilidad de repositorios, acceso al código fuente y actividad de sus comunidades de desarrollo.

En la segunda etapa se examinaron los prototipos implementados, aplicando los criterios de apariencia, actualización durante el desarrollo y facilidades de implementación. Los resultados se organizaron mediante tablas comparativas y se complementaron con la identificación de ventajas y limitaciones observadas en cada herramienta.

La interpretación se realizó de manera descriptiva, considerando las condiciones particulares de los prototipos y evitando generalizaciones sobre el comportamiento de ambas herramientas en otros contextos.

Resultados y/o Discusión

3.1. Comparación de características generales

La revisión documental permitió identificar las características generales de Streamlit y Bokeh, considerando aspectos relacionados con su compatibilidad, instalación, disponibilidad de recursos y actividad en los repositorios de desarrollo. Los resultados se presentan en la Tabla 2.

Tabla 2.

Comparación entre las características de librerías.

Parámetros	Streamlit	Bokeh
Funciona con la última versión de Python	Cumple	Cumple
Tiene información en repositorios	Cumple	Cumple
Fácil de instalar	Cumple	Cumple
Es de código abierto	Cumple	Cumple
Issues	763	913
Fork	4190	3119
Tengan una comunidad abierta para realizar una discusión	Cumple	Cumple
Las librerías puedan generarse como aplicaciones	Cumple	Cumple

Nota. Datos correspondientes a la revisión de repositorios realizada en la investigación original de Campaña Lozano (2022). Los valores de issues y forks son históricos y requieren indicar la fecha exacta de consulta, si está disponible.

Los resultados evidencian que ambas herramientas cumplieron los criterios generales establecidos en la investigación. En cuanto a la actividad de los repositorios, Bokeh registró un mayor número de issues, mientras que Streamlit presentó una mayor cantidad de forks.

Sin embargo, estos indicadores reflejan aspectos de la actividad de los repositorios y no permiten determinar directamente la calidad del software, la facilidad de implementación ni el rendimiento de las herramientas.

3.2. Análisis de apariencia

La evaluación de apariencia permitió identificar diferencias en la personalización de las interfaces, la configuración de los gráficos y la disponibilidad de componentes durante la implementación de los prototipos. Los resultados registrados se presentan en la Tabla 3.

Tabla 3.

Apariencia

Aspecto evaluado	Argumentos	Bokeh	Streamlit
Apariencia general del dashboard	Existencia de temas predeterminados.	NO	SI
Apariencia general del dashboard	Personalización total	SI	NO
Apariencia general del dashboard	Personalización parcial	SI	SI
Apariencia de las gráficas	Gráficos integrados	NO	SI
Apariencia de las gráficas	Gráficos incorporados	NO	SI
Apariencia de las gráficas	Personalización total	SI	NO
Apariencia de las gráficas	Personalización Parcial	SI	SI
Componentes preexistentes	Componentes preexistentes	NO	SI

Aspecto evaluado	Argumentos	Bokeh	Streamlit
Títulos y subtítulos	Uso de archivos HTML y CSS	SI	NO
Títulos y subtítulos	Personalización total	SI	NO
Títulos y subtítulos	Personalización parcial	SI	SI

Nota. Se conservan los resultados registrados en los prototipos de Campaña Lozano (2022). Las respuestas «Sí» y «No» describen la evaluación original y no deben interpretarse como ausencia absoluta de funcionalidades en las herramientas.

Ventajas y limitaciones de Streamlit

En el prototipo desarrollado con Streamlit se identificaron ventajas relacionadas con la disponibilidad de componentes predefinidos, la incorporación de gráficos y la facilidad para estructurar la interfaz. Estas características simplificaron determinadas tareas de implementación y redujeron la necesidad de configurar individualmente algunos elementos visuales.

No obstante, durante el desarrollo se observaron limitaciones para realizar ciertas modificaciones de apariencia. Asimismo, la personalización de gráficos mediante bibliotecas externas requirió consultar documentación adicional y familiarizarse con sus funcionalidades.

Ventajas y limitaciones de Bokeh

En el prototipo desarrollado con Bokeh se identificaron posibilidades de personalización de gráficos y elementos visuales, particularmente en aspectos relacionados con colores, dimensiones y presentación de componentes. Estas características permitieron adaptar las visualizaciones a los requerimientos del caso estudiado.

Sin embargo, algunas tareas de personalización y configuración de componentes resultaron más elaboradas durante la implementación. En determinados casos fue necesario recurrir a conocimientos complementarios y consultar documentación técnica para completar las modificaciones previstas.

En términos generales, Streamlit facilitó la construcción de la interfaz mediante componentes predefinidos, mientras que Bokeh permitió una configuración visual más detallada en el prototipo evaluado, aunque con un proceso de implementación más elaborado.

3.3. Actualización durante el desarrollo

Esta categoría permitió examinar el comportamiento de los prototipos al realizar modificaciones en el código durante la fase de desarrollo en un entorno local. La Tabla 4 presenta los resultados observados respecto a los mecanismos de actualización.

Tabla 4. Rendimiento

Categoría	Parámetro	Argumentos	Bokeh	Streamlit
RENDIMIENTO	Capacidad para refrescar automáticamente el reporte (front-end) luego de un cambio de código en la fase de desarrollo en un ambiente local.	Actualización de servicio automático	NO	SI
RENDIMIENTO	Capacidad para refrescar automáticamente el reporte (front-end) luego de un cambio de código en la fase de desarrollo en un ambiente local.	Actualización del servicio manual.	SI	SI

Ventajas y limitaciones de Streamlit

Durante el desarrollo del prototipo con Streamlit se observó que las modificaciones realizadas en el código podían reflejarse mediante el mecanismo de actualización de la aplicación. Esta característica facilitó la revisión de cambios durante las actividades de implementación y prueba.

En el criterio específico evaluado no se identificaron limitaciones relevantes relacionadas con la actualización automática, aunque este resultado corresponde únicamente a las condiciones del prototipo desarrollado.

Ventajas y limitaciones de Bokeh

El prototipo elaborado con Bokeh permitió ejecutar y visualizar la aplicación en un entorno local. Sin embargo, durante la implementación evaluada, determinadas modificaciones requirieron reiniciar manualmente el servidor para reflejar los cambios realizados.

Esta diferencia corresponde al procedimiento utilizado en la investigación y no implica que todas las aplicaciones desarrolladas con Bokeh requieran necesariamente el mismo mecanismo de actualización.

En consecuencia, Streamlit presentó un proceso de actualización más directo en el caso estudiado, mientras que Bokeh requirió intervenciones adicionales durante determinadas modificaciones del código. Estos resultados no permiten establecer conclusiones sobre la velocidad de procesamiento o el rendimiento computacional de ambas herramientas.

3.4. Facilidades de implementación

La evaluación de las facilidades de implementación consideró la experiencia obtenida durante el desarrollo de los prototipos, la disponibilidad de recursos de aprendizaje y la integración con otros lenguajes de programación. Los resultados se presentan en la Tabla 5.

Tabla 5.

Facilidades para el Desarrollo

Categoría	Parámetro	Argumentos	Bokeh	Streamlit
Experiencia del desarrollador, curva de aprendizaje	Prototipo	Actualización de servicio automático	NO	SI
Experiencia del desarrollador, curva de aprendizaje	Prototipo	Actualización del servicio manual	SI	SI
Posibilidad de trabajo con de otros lenguajes	Integración	Trabajo con más de 2 lenguajes de programación	SI	NO
Posibilidad de trabajo con de otros lenguajes	Integración	Usa el lenguaje adicional para la personalización de las gráficas	SI	SI
Posibilidad de trabajo con de otros lenguajes	Integración	Usa el lenguaje adicional para el prototipo en general	SI	NO

Ventajas y limitaciones de Streamlit

Durante la implementación del prototipo con Streamlit, la disponibilidad de documentación, ejemplos y recursos de consulta facilitó la comprensión de sus funcionalidades. Asimismo, la posibilidad de construir la aplicación principalmente mediante Python contribuyó a simplificar el proceso de desarrollo.

No obstante, la incorporación de determinadas funcionalidades o visualizaciones externas requirió consultar documentación específica y comprender los mecanismos de integración correspondientes.

Ventajas y limitaciones de Bokeh

En el caso de Bokeh, se identificaron ventajas relacionadas con las posibilidades de integración y personalización de elementos interactivos, así como con la disponibilidad de ejemplos y recursos técnicos para el desarrollo de visualizaciones.

Sin embargo, durante la implementación del prototipo fue necesario dedicar mayor esfuerzo a comprender la estructura de algunos componentes y sus mecanismos de configuración. Determinadas personalizaciones también requirieron conocimientos complementarios de tecnologías web.

En síntesis, Streamlit presentó facilidades relacionadas con la construcción de aplicaciones mediante componentes predefinidos y una estructura de programación accesible, mientras que Bokeh ofreció posibilidades de configuración más detalladas, acompañadas de una mayor complejidad de implementación en el caso analizado.

Conclusiones

El análisis comparativo permitió identificar que Streamlit y Bokeh presentan características y funcionalidades que favorecen el desarrollo de dashboards interactivos en Python. Ambas herramientas cuentan con recursos de documentación, repositorios de código abierto y comunidades de desarrollo que facilitan su utilización y aprendizaje.

En relación con la apariencia y las facilidades de implementación, Streamlit destacó por la disponibilidad de componentes predefinidos y la simplicidad de su estructura

de programación, lo que facilitó la construcción del prototipo. Por su parte, Bokeh ofreció mayores posibilidades de personalización de gráficos y elementos visuales en el caso estudiado, aunque requirió un proceso de configuración más elaborado.

Respecto a la actualización durante el desarrollo, Streamlit permitió visualizar los cambios realizados en el código de manera más directa, mientras que la implementación evaluada con Bokeh requirió intervenciones manuales para reflejar determinadas modificaciones. Estas diferencias corresponden a las condiciones de los prototipos analizados y no constituyen una evaluación general del rendimiento de ambas herramientas.

Finalmente, se concluye que la elección entre Streamlit y Bokeh debe considerar los objetivos del proyecto, el nivel de personalización requerido y la experiencia del desarrollador. Streamlit constituye una alternativa para facilitar la construcción de aplicaciones interactivas, mientras que Bokeh ofrece opciones de configuración más detalladas. En consecuencia, ambas herramientas presentan ventajas y limitaciones que deben valorarse de acuerdo con las necesidades específicas de implementación.

Contribución de autoría (Taxonomía CRediT)

Stefany Antonella Campaña Lozano: Conceptualización, metodología, investigación, análisis formal, desarrollo de software, visualización de datos, redacción del borrador original, revisión y edición del manuscrito.

Referencias

Enciclopedia Concepto. (2022). ¿Qué es el método analítico? <https://concepto.de/metodo-analitico/>

Campaña Lozano, S. A. (2022). Implementación de dashboards interactivos para la visualización de datos usando lenguaje de programación Python [Tesis de grado, Universidad Técnica Particular de Loja].

Colaboradores de Bokeh. (2022). bokeh.documento . Bokeh.Documento. <https://docs.bokeh.org/en/latest/docs/reference/document.html>

Data Science. (2024). Bokeh: la biblioteca de visualización Python de nueva generación.

DataScientest <https://datascientest.com/es/bokeh-la-biblioteca-de-visualizacion-python#:~:text=En%20resumen%2C%20Bokeh%20es%20una,pueden%20visualizarse%20en%20diferentes%20plataformas.>

Data Scientist en Python. (2023). Streamlit. DataScientest. <https://entrenamiento-data-scientist-python.readthedocs.io/leccion5/streamlit.html>

- Dask. (2018). Anaconda, In. <https://docs.dask.org/en/stable/>
- Few, Stephen. (2006). Information dashboard design : the effective visual communication of data. O'Reilly.
- González Guerrero, A. (2021). EVOLUCIÓN HISTÓRICA DE LOS DASHBOARDS.
- Jurić, S. (2019). Usporedba Python biblioteka za vizualizaciju podataka. <https://urn.nsk.hr/urn:nbn:hr:137:510308>
- Len Bass, P. C. R. K. (2012). Software Architecture in Practice, Third Edition.
- McKinney, W. (n.d.). Python for Data Analysis. www.it-ebooks.info
- Mhadhbi, Nadia.(2024). Tutorial de Python: Streamlit. Datacamp. <https://www.datacamp.com/es/tutorial/streamlit>
- Streamlit Inc. (2022). Streamlit documentation. <https://docs.streamlit.io/>
- Murillo, W. (2024). Diseño de dashboards interactivos: mejorando la interpretación de datos en la elasticidad de precios y demanda de productos ferreteros en la empresa “Importador Ferretero Trujillo” (Tesis de maestría). Universidad de las Américas, Quito.
- Vanrell Jordi. (2024). Librerías de Python para visualización interactiva de mapas. Data Machine Learning Visualization. <https://blog.damavis.com/librerias-de-python-para-visualizacion-interactiva-de-mapas/>