

UNIVERSIDAD NACIONAL AGRARIA LA MOLINA
Departamento de Estadística Informática

INFORME FINAL DE PROYECTO

Lenguaje de Programación 2 (LP2)

TipoCambio.pe

Comparador de Tipo de Cambio en Perú

INTEGRANTES:

Javier Uraco
Fiorella Fuentes
Sebastián Fernández

Semestre 2025-2
Diciembre 2025

Repositorio: github.com/JavierAnthonyUS/tipo-cambio-peru
Demo en vivo: <https://tipo-cambio-peru.onrender.com>

TABLA DE CONTENIDOS

TABLA DE CONTENIDOS	2
1. INTRODUCCIÓN	4
1.1 Contexto y Problemática.....	4
1.2 Relevancia y Aporte Social.....	4
1.3 Objetivos del Proyecto.....	4
Objetivo General	4
Objetivos Específicos.....	4
2. DISEÑO Y PLANEAMIENTO DE LA EXTRACCIÓN	5
2.1 Análisis de Fuentes	5
2.2 Fuentes Seleccionadas	5
2.3 Estrategia de Extracción	5
BCRP (API REST)	5
Kambista y Rextie (Web Scraping)	5
2.4 Arquitectura del Sistema.....	5
3. ESTRUCTURACIÓN DE DATOS.....	6
3.1 Formato de Almacenamiento.....	6
3.2 Diccionario de Datos (13 columnas).....	6
3.3 Relevancia para el Análisis.....	6
4. CÓDIGO DOCUMENTADO	7
4.1 Estándar de Documentación.....	7
4.2 Ejemplo: Scraper BCRP	7
4.3 Ejemplo: Scraper Selenium	7
4.4 Estructura del Proyecto	7
5. APLICACIÓN WEB	8
5.1 Tecnología Utilizada	8
5.2 Páginas de la Aplicación.....	8
5.3 Despliegue en la Nube	8
6. DIFICULTADES ENCONTRADAS Y SOLUCIONES	9
6.1 Páginas Web Dinámicas.....	9
6.2 Identificación de Valores Correctos	9
6.3 Tiempos de Espera en Selenium.....	9
6.4 API del BCRP	9
6.5 Compatibilidad NiceGUI 3.4.1	9
6.6 Coordinación del Equipo.....	9
7. TRABAJO EN EQUIPO	10

7.1 Repositorio GitHub	10
7.2 Distribución de Tareas.....	10
7.3 Historial de Commits.....	10
8. CONCLUSIONES.....	11
8.1 Aprendizajes del Proceso	11
9. REFERENCIAS Y RECURSOS	12
9.1 Fuentes de Datos	12
9.2 Tecnologías y Documentación.....	12
9.3 Enlaces del Proyecto	12

1. INTRODUCCIÓN

1.1 Contexto y Problemática

En Perú, el mercado de cambio de divisas presenta una característica particular: existen múltiples casas de cambio, tanto físicas como digitales, cada una con tasas de compra y venta diferentes para el dólar estadounidense. Esta diversidad, aunque beneficiosa para la competencia, genera un problema práctico para el ciudadano común: ¿dónde obtener la mejor tasa de cambio?

Actualmente, una persona que desea cambiar dólares debe visitar manualmente múltiples sitios web o acudir físicamente a diferentes establecimientos para comparar tasas. Este proceso consume tiempo valioso y, en muchos casos, las personas terminan aceptando tasas subóptimas por desconocimiento o falta de tiempo para investigar.

La diferencia de tasas entre casas de cambio puede parecer pequeña (centavos), pero al cambiar montos significativos, el impacto económico es considerable. Por ejemplo, al cambiar \$1,000 USD, una diferencia de S/ 0.05 en la tasa representa S/ 50 de ahorro o pérdida.

1.2 Relevancia y Aporte Social

Este proyecto aporta valor a la sociedad peruana de las siguientes maneras:

- Ahorro económico para las familias:** Permite identificar la mejor opción de cambio, generando ahorros tangibles en cada transacción.
- Democratización de la información:** Información que antes requería búsqueda manual ahora está disponible en segundos para cualquier persona.
- Optimización del tiempo:** Automatiza un proceso que manualmente tomaría 15-20 minutos en apenas segundos.
- Transparencia del mercado:** Facilita la comparación y ayuda a evitar casas de cambio con tasas abusivas.
- Registro histórico:** Permite analizar tendencias y patrones en el tipo de cambio a lo largo del tiempo.

1.3 Objetivos del Proyecto

Objetivo General

Desarrollar un sistema automatizado de extracción, integración y visualización de tipos de cambio de múltiples fuentes en Perú, utilizando técnicas de web scraping y consumo de APIs.

Objetivos Específicos

- Implementar scrapers para extraer datos de al menos 3 fuentes diferentes.
- Estructurar los datos extraídos en formato CSV para análisis posterior.
- Desarrollar una aplicación web interactiva para visualización en tiempo real.
- Documentar el código y el proceso de desarrollo de forma colaborativa en GitHub.

2. DISEÑO Y PLANEAMIENTO DE LA EXTRACCIÓN

2.1 Análisis de Fuentes

Antes de iniciar el desarrollo, se realizó un análisis exhaustivo de las posibles fuentes de datos. Los criterios de selección fueron:

- **Confiabilidad:** Fuentes oficiales o casas de cambio reconocidas.
- **Actualización:** Datos actualizados en tiempo real o con frecuencia diaria.
- **Accesibilidad técnica:** Posibilidad de extraer datos mediante API o scraping.
- **Relevancia:** Uso significativo por parte de la población peruana.

2.2 Fuentes Seleccionadas

Fuente	Tipo	Método	Justificación
BCRP	API REST	requests + JSON	Fuente oficial del Banco Central
Kambista	Web Scraping	Selenium + regex	Casa de cambio digital líder
Rextie	Web Scraping	Selenium + regex	Popular entre empresas y personas

2.3 Estrategia de Extracción

BCRP (API REST)

El Banco Central de Reserva del Perú ofrece una API pública que retorna datos en formato JSON. La extracción es directa mediante la librería `requests`. Se utilizan los códigos de serie PD04638PD (compra) y PD04639PD (venta) para obtener las tasas oficiales.

Kambista y Rextie (Web Scraping)

Estas páginas utilizan JavaScript para cargar dinámicamente los precios. Esto significa que herramientas como BeautifulSoup, que solo leen HTML estático, no pueden obtener los datos. Por esta razón, se implementó Selenium, que controla un navegador real capaz de ejecutar JavaScript y esperar a que los elementos carguen.

2.4 Arquitectura del Sistema

El sistema se compone de tres capas principales:

1. **Capa de Extracción:** Tres scrapers independientes (BCRP, Kambista, Rextie) que obtienen datos de cada fuente.
2. **Capa de Integración:** Módulo integrador que combina los datos, calcula métricas (spreads) y determina la mejor opción.
3. **Capa de Presentación:** Aplicación web desarrollada con NiceGUI que permite visualización interactiva.

3. ESTRUCTURACIÓN DE DATOS

3.1 Formato de Almacenamiento

Los datos extraídos se almacenan en formato CSV (Comma-Separated Values) en el archivo *data/processed/tipo_cambio_historico.csv*. Este formato fue elegido por su simplicidad, portabilidad y compatibilidad con herramientas de análisis como Excel, Pandas y R.

3.2 Diccionario de Datos (13 columnas)

Columna	Tipo	Descripción
timestamp	datetime	Fecha y hora de extracción
tc_bcrp_compra	float	Tipo de cambio compra BCRP
tc_bcrp_venta	float	Tipo de cambio venta BCRP
tc_kambista_compra	float	Tipo de cambio compra Kambista
tc_kambista_venta	float	Tipo de cambio venta Kambista
tc_rextie_compra	float	Tipo de cambio compra Rextie
tc_rextie_venta	float	Tipo de cambio venta Rextie
spread_bcrp	float	Diferencia venta - compra BCRP
spread_kambista	float	Diferencia venta - compra Kambista
spread_rextie	float	Diferencia venta - compra Rextie
mejor_compra	string	Fuente con mejor tasa para comprar USD
mejor_venta	string	Fuente con mejor tasa para vender USD
cambio_detectado	boolean	Indica si hubo cambio vs registro anterior

3.3 Relevancia para el Análisis

Esta estructura permite realizar múltiples tipos de análisis:

- **Análisis temporal:** Evolución de tasas a lo largo del tiempo.
- **Comparación entre fuentes:** Identificar qué fuente ofrece mejores tasas consistentemente.
- **Análisis de spreads:** Determinar qué fuente tiene menor margen de ganancia.
- **Detección de anomalías:** Identificar cambios bruscos o datos atípicos.

4. CÓDIGO DOCUMENTADO

4.1 Estándar de Documentación

Todo el código del proyecto sigue el estándar de documentación de Python utilizando docstrings. Cada función incluye:

- Descripción breve de la función
- Parámetros de entrada (Args)
- Valores de retorno (Returns)
- Excepciones que puede lanzar (Raises)

4.2 Ejemplo: Scraper BCRP

```
def obtener_tipo_cambio_bcrp():

    """
    Obtiene el tipo de cambio del BCRP mediante su API REST.

    Returns:
        dict: Diccionario con tc_bcrp_compra, tc_bcrp_venta, exito
    """

```

4.3 Ejemplo: Scraper Selenium

```
def obtener_tipo_cambio_kambista():

    """
    Extrae el tipo de cambio de Kambista usando Selenium.

    Utiliza Chrome en modo headless para cargar la página
    y regex para extraer los valores numéricos.

    Returns:
        dict: Diccionario con tc_kambista_compra, tc_kambista_venta, exito
    """

```

4.4 Estructura del Proyecto

```
tipo-cambio-peru/
├── AppTipoCambioPe.py      # Aplicación web principal
└── src/
    ├── scraper_bcrp.py    # Extractor API BCRP
    ├── scraper_kambista.py # Extractor Selenium
    ├── scraper_rextie.py   # Extractor Selenium
    └── integrador.py       # Combina fuentes
├── data/processed/          # Datos CSV
└── docs/                    # Documentación
    └── notebooks/           # Análisis exploratorio
    └── requirements.txt     # Dependencias
```

5. APPLICACIÓN WEB

5.1 Tecnología Utilizada

La aplicación web fue desarrollada utilizando **NiceGUI**, un framework de Python que permite crear interfaces web modernas sin necesidad de escribir HTML, CSS o JavaScript. Esto mantiene el proyecto 100% en Python.

5.2 Páginas de la Aplicación

Página	URL	Funcionalidad
Inicio	/	Presentación del proyecto y características
Demo	/demo	Ejecución de scrapers en tiempo real
Análisis	/analisis	Gráficos Plotly y calculadora de ahorro
Equipo	/equipo	Información de integrantes y tecnologías

5.3 Despliegue en la Nube

La aplicación está desplegada en **Render** (servicio gratuito) y es accesible públicamente en:

<https://tipo-cambio-peru.onrender.com>

6. DIFICULTADES ENCONTRADAS Y SOLUCIONES

6.1 Páginas Web Dinámicas

Problema: Al intentar extraer datos de Kambista y Rextie usando BeautifulSoup, el HTML obtenido no contenía los precios. Las páginas cargan los datos dinámicamente con JavaScript.

Solución: Se implementó Selenium WebDriver en modo headless, que ejecuta un navegador real capaz de procesar JavaScript y esperar a que los elementos carguen.

6.2 Identificación de Valores Correctos

Problema: Las páginas contienen múltiples números (precios, porcentajes, años, etc.). Era difícil identificar cuáles correspondían al tipo de cambio.

Solución: Se utilizaron expresiones regulares para encontrar patrones numéricos y se filtraron por rango válido (3.30 - 3.50 soles por dólar).

6.3 Tiempos de Espera en Selenium

Problema: Los scrapers de Selenium tardaban demasiado o fallaban intermitentemente porque los elementos no cargaban a tiempo.

Solución: Se implementaron esperas explícitas (WebDriverWait) y se optimizaron los tiempos de carga.

6.4 API del BCRP

Problema: El formato de fechas de la API del BCRP requería un rango específico, y los fines de semana no hay datos actualizados.

Solución: Se implementó un rango dinámico de 7 días para asegurar obtener el dato más reciente disponible.

6.5 Compatibilidad NiceGUI 3.4.1

Problema: La documentación de NiceGUI estaba desactualizada. Algunos componentes como ui.header() ya no existían en la versión instalada.

Solución: Se adaptó el código usando componentes alternativos como ui.row() y se consultó directamente el código fuente de la librería.

6.6 Coordinación del Equipo

Problema: Los integrantes tenían horarios diferentes y era difícil sincronizar el trabajo.

Solución: Se utilizó GitHub como plataforma central de colaboración, con commits frecuentes y mensajes descriptivos.

7. TRABAJO EN EQUIPO

7.1 Repositorio GitHub

Todo el desarrollo se realizó de forma colaborativa en GitHub:

github.com/JavierAnthonyUS/tipo-cambio-peru

7.2 Distribución de Tareas

Integrante	Rol	Contribuciones
Javier Uraco	Líder	Scrapers BCRP y Rextie, Integrador, App Web (estructura, Demo, Inicio, Equipo)
Fiorella Fuentes	Desarrolladora	Scraper Kambista, Página Análisis con gráficos Plotly, DIFICULTADES.md
Sebastián Fernández	Documentación	README.md, requirements.txt, documentación técnica

7.3 Historial de Commits

El repositorio muestra la evidencia del trabajo colaborativo con commits de todos los integrantes:

- *feat: agregar aplicación web con NiceGUI* - Javier
- *feat: completar página Análisis con gráficos Plotly* - Fiorella
- *docs: agregar documentación de dificultades y soluciones* - Fiorella
- *docs: actualizar README y requirements* - Sebastián
- *fix: configurar puerto para Render* - Javier
- *fix: usar datos demo para Selenium en Render* - Javier

8. CONCLUSIONES

El desarrollo de este proyecto ha permitido alcanzar todos los objetivos planteados y ha generado aprendizajes significativos en el equipo:

1. **Extracción automatizada exitosa:** Se implementaron scrapers funcionales para 3 fuentes diferentes, combinando técnicas de API REST y web scraping con Selenium.
2. **Estructuración de datos efectiva:** El formato CSV con 13 columnas permite análisis histórico, comparaciones entre fuentes y detección de tendencias.
3. **Aplicación web funcional:** Se desarrolló una interfaz moderna y completa utilizando NiceGUI, manteniendo el proyecto 100% en Python.
4. **Despliegue en la nube:** La aplicación está disponible públicamente en Render, demostrando capacidad de llevar un proyecto a producción.
5. **Trabajo colaborativo:** El uso de GitHub facilitó la coordinación del equipo y dejó evidencia clara de las contribuciones de cada integrante.
6. **Documentación completa:** El código está documentado con docstrings y el proyecto cuenta con documentación técnica en el repositorio.

8.1 Aprendizajes del Proceso

- **Investigar antes de implementar:** Entender cómo funciona una página/API antes de escribir código.
- **Manejar errores proactivamente:** Anticipar qué puede fallar y tener planes de contingencia.
- **Documentar soluciones:** Para referencia futura y para el equipo.
- **Iterar y mejorar:** La primera solución raramente es la mejor.

9. REFERENCIAS Y RECURSOS

9.1 Fuentes de Datos

- Banco Central de Reserva del Perú - <https://estadisticas.bcrp.gob.pe>
- Kambista - <https://kambista.com>
- Rextie - <https://rextie.com>

9.2 Tecnologías y Documentación

- Python - <https://docs.python.org>
- Selenium - <https://www.selenium.dev/documentation>
- NiceGUI - <https://nicegui.io/documentation>
- Plotly - <https://plotly.com/python>
- Pandas - <https://pandas.pydata.org/docs>

9.3 Enlaces del Proyecto

- **Repositorio GitHub:** github.com/JavierAnthonyUS/tipo-cambio-peru
- **Demo en vivo:** <https://tipo-cambio-peru.onrender.com>

— *Fin del Informe* —

Desarrollado con dedicación para LP2 - UNALM 2025