



IIC2115 – Programación como Herramienta para la Ingeniería (II/2025)

Laboratorio 7

Aspectos generales

- **Objetivo:** Evaluar la aplicación de web scraping y uso de APIs para extraer información de internet y construir una página web interactiva que permita visualizar datos y presentar información de manera dinámica y ordenada.
- **Entrega:** Parte 1 lunes 24/11 a las 23:59, Parte 2 domingo 07/12 a las 23:59 hrs. Archivos Python Notebook L7_1.ipynb y L7_2.ipynb con las soluciones de las partes 1 y 2 del laboratorio, respectivamente. Los archivos deben estar ubicados en la carpeta L7 del repositorio privado. Entregas que no cumplan con esta especificación no serán corregidas.
- **Formato de entrega:** Utilice múltiples celdas de texto y código para estructurar los archivos. Todas las celdas utilizadas deben estar ejecutadas al momento de entregar el ejercicio, de modo que las salidas generadas sean visibles. **Adjunte además el historial completo de su interacción con asistentes basados en IA, en caso de existir.** Entregas que no cumplan el formato tendrán un descuento de 0,5 puntos.
- **Entregas atrasadas:** el descuento por atraso para la Parte 1 es de 1 punto cada 10 minutos o fracción. El descuento por atraso para la Parte 2 es de 1 punto por cada hora o fracción.
- **Issues:** Las discusiones en las *issues* del Syllabus que sean relevantes para el desarrollo de la evaluación, serán destacadas. Así mismo, el uso de librerías externas que solucionen aspectos fundamentales del problema no podrán ser utilizadas. Solo se podrán utilizar las que han sido aprobadas en las *issues*, previa consulta de los estudiantes.
- **Entregas con errores de sintaxis y/o que generen excepciones en todas las ejecuciones serán calificados con nota 1.0.**

Descripción del problema

El objetivo de este laboratorio es desarrollar habilidades para integrar datos climáticos de distintas fuentes, construir una base de datos estructurada y visualizar la información de forma dinámica y exploratoria.

En ambas partes del laboratorio trabajará con información climática de la API de OpenWeatherMap y de páginas web públicas. **No DEBEN UTILIZARSE** como fuentes de información los conjuntos de datos adicionales utilizados en los ejercicios formativos.

IMPORTANTE: No deje su API KEY en el código, de lo contrario será penalizado con 2,0 ptos. de la nota final de su laboratorio.

Parte 1

- a) Diseñe un sistema que recopile información meteorológica actual y pronosticada a partir de un conjunto de tamaño arbitrario de coordenadas. El sistema debe utilizar múltiples endpoints de OpenWeatherMap. Los resultados deben consolidarse en una estructura coherente que indique, para cada ubicación buscada, el nombre de la ciudad y los datos correspondientes, presentando unidades normalizadas.

Con los datos descargados, realice un análisis que caracterice para cada ciudad su perfil climático proyectado, combinando simultáneamente temperatura, humedad, viento y presión, de modo que se puedan distinguir entre comportamientos estables, moderadamente variables o altamente fluctuantes. El criterio empleado para establecer estas categorías debe estar claramente definido y sustentado en las variables disponibles.

Muestre el funcionamiento de su sistema con un ejemplo simple.

- b) Diseñe un sistema que, para un conjunto arbitrario de ciudades, recopile información actual y pronosticada sobre la calidad del aire utilizando llamadas a APIs. Con los datos obtenidos, realice un análisis que permita identificar y comparar entre ciudades la presencia de episodios relevantes de deterioro de la calidad del aire, combinando simultáneamente distintos contaminantes en uno o más indicadores cuantitativos y midiendo su variación en el horizonte del pronóstico.

Muestre el funcionamiento de su sistema con un ejemplo simple.

- c) Desarrolle una aplicación en Dash que permita ejecutar los sistemas construidos en las misiones anteriores y visualizar sus resultados de forma integrada. La aplicación debe permitir comparar cómo evolucionan en el tiempo las variables meteorológicas y de calidad del aire de un conjunto seleccionable de ciudades,

de modo que sea posible identificar momentos en que los cambios en alguna variable climática se asocian a cambios relevantes en los contaminantes.

La visualización debe hacer posible detectar estas asociaciones temporales al ajustar dinámicamente la selección de ciudades, el período observado y las variables mostradas. Se espera un uso de Dash que vaya más allá de gráficos estáticos, incorporando interacción fluida y coordinación entre componentes para apoyar dicho análisis comparativo.

Para verificar el funcionamiento, incluya en su entrega capturas de pantalla que muestre el funcionamiento del dashboard con interacciones relevantes.

Parte 2

a) Construya un sistema de web scraping que obtenga información pública y estructurada sobre vulnerabilidad, exposición o riesgo climático a nivel país o ciudad, a partir de una fuente confiable y accesible. La información extraída debe ser limpiada, estandarizada y organizada de modo que permita un análisis comparativo entre países.

Cree una base de datos relacional con la información descargada, cuidando de construir un modelo adecuado, y realice un análisis mediante múltiples consultas en SQL, que identifique grupos de países con perfiles de vulnerabilidad similares. Establezca categorías interpretables a partir de los indicadores disponibles. El criterio utilizado para conformar estos grupos y la lógica empleada para distinguir entre niveles altos, medios o bajos de vulnerabilidad deben definirse explícitamente y sustentarse en los datos obtenidos.

b) A partir del índice de vulnerabilidad climática obtenido mediante web scraping en la misión anterior, construya un conjunto de datos que combine dicho índice con condiciones climáticas y ambientales afines, obtenidas desde la API de OpenWeatherMap.

Utilizando este conjunto integrado, entrene modelos de aprendizaje supervisado y/o no supervisado que permitan clasificar a los países en categorías de vulnerabilidad a partir de las variables climáticas recientes, definiendo de manera explícita cómo se construye la variable objetivo, qué atributos se utilizan como predictores y cómo se evalúa cuantitativamente el desempeño del modelo.

Se espera además, un análisis que interprete el rol relativo de las distintas variables en la predicción, de modo de discutir qué aspectos del clima actual parecen más asociados a los niveles de vulnerabilidad reportados externamente.

c) Desarrolle una aplicación web en Dash organizada en múltiples pestañas que permita explorar de forma integrada los conjuntos de datos construidos en las misiones anteriores. La aplicación debe incluir una pestaña dedicada a visualizar la información y análisis de vulnerabilidad obtenida mediante web scraping, otra orientada a examinar las condiciones climáticas actuales obtenidas desde la API y una tercera que permita analizar los resultados del modelo de clasificación o agrupamiento construido con scikit-learn.

Cada pestaña debe permitir la selección interactiva de países y/o ciudades, actualizar dinámicamente su contenido y presentar los resultados mediante gráficos comparativos, vistas geográficas o resúmenes numéricos que permitan interpretar con claridad las diferencias entre los distintos elementos observados.

Se espera también la incorporación de un mapa mundial interactivo que permita mostrar información climática o de vulnerabilidad sobre la ubicación de cada país y/o ciudad y que, al seleccionarlo, permita acceder a un resumen detallado que combine variables recientes, indicadores externos y resultados de la modelación realizada mediante aprendizaje de máquina.

Para verificar el funcionamiento, incluya en su entrega capturas de pantalla que muestre el funcionamiento del dashboard con interacciones relevantes.

Consideraciones sobre la metodología de trabajo

En este laboratorio no solo interesa el resultado final, sino también el proceso seguido para construir la solución. Por ello, se espera que el desarrollo de su entrega se organice a partir de una **división progresiva de cada problema en subproblemas más simples**, siguiendo un enfoque de *divide y vencerás*. La subdivisión debe reflejarse tanto en el diseño de las soluciones, como en la organización del notebook: cada celda debe ser temáticamente coherente, evitando reunir todo el código en una única celda. Todos los supuestos y simplificaciones que se utilicen deben quedar claramente explicitados en el notebook. Además, siempre que sea posible, se recomienda complementar con diagramas.

Otro componente clave es el uso de **prompts estructurados** para interactuar con herramientas de IA. Cada vez que se utilice un asistente para generar código, se debe registrar en una celda de texto el prompt utilizado, incluyendo al menos:

- Propósito y requisitos,
- Entradas y salidas,
- Restricciones y supuestos,
- Hitos intermedios,

- Convenciones.

Este nivel de detalle permite mantener claridad sobre lo que se espera del código y facilita evaluar cómo la IA fue utilizada en el proceso de desarrollo.

El único caso donde está permitido usar código generado por IA sin necesidad de dar estructura al prompt, es cuando este código es luego progresivamente modificado y/o mejorado por el estudiante mismo.

IMPORTANTE: los prompts estructurados no debe haber sido escritos por un asistente basado en IA, deben ser completamente escritos por el estudiante que entrega el laboratorio.

Finalmente, todo el código debe estar acompañado de **tests explícitos**. Un test, en este contexto, es simplemente un fragmento de código que demuestra que otro fragmento funciona como se espera, considerando casos no triviales. No se exige un test unitario por cada método, pero sí que se diseñen pruebas convincentes que permitan validar comportamientos relevantes. Para cada caso probado, debe quedar visible el resultado esperado y el resultado obtenido, de modo que cualquier lector pueda verificar si el comportamiento es correcto.

Es importante enfatizar que **adjuntar el historial de interacción con un asistente de IA no constituye por sí mismo una justificación válida ni suficiente**. Si bien dicho historial debe incluirse como material adicional, no sustituye en ningún caso los requisitos de formato, estructuración y testeo descritos anteriormente, los cuales son obligatorios y forman parte de la evaluación.

Corrección

Para la corrección se revisarán los procedimientos desarrollados para responder las diferentes preguntas y cómo estos cumplen con la materia del capítulo y la metodología de trabajo antes descrita. Dado lo abierto de las preguntas, se espera que todas las respuestas incluyan análisis y visualizaciones que permitan justificar las decisiones tomadas.

Política de Integridad Académica

Los/as estudiantes de la Escuela de Ingeniería de la Pontificia Universidad Católica de Chile deben mantener un comportamiento acorde a la Declaración de Principios de la Universidad. En particular, se espera que mantengan altos estándares de honestidad académica. Cualquier acto deshonesto o fraude académico está prohibido; los/as estudiantes que incurran en este tipo de acciones se exponen a un Procedimiento Sumario. Es responsabilidad de cada estudiante

conocer y respetar el documento sobre Integridad Académica publicado por la Dirección de Docencia de la Escuela de Ingeniería.

Específicamente, para los cursos del Departamento de Ciencia de la Computación, rige obligatoriamente la siguiente política de integridad académica. Todo trabajo presentado por un/a estudiante para los efectos de la evaluación de un curso debe ser hecho **individualmente** por el/la estudiante, **sin apoyo en material de terceros**. Por “trabajo” se entiende en general las interrogaciones escritas, las tareas de programación u otras, los trabajos de laboratorio, los proyectos, el examen, entre otros.

En particular, si un/a estudiante copia un trabajo, o si a un/a estudiante se le prueba que compró o intentó comprar un trabajo, **obtendrá nota final 1.1 en el curso** y se solicitará a la Dirección de Pregrado de la Escuela de Ingeniería que no le permita retirar el curso de la carga académica semestral.

Por “copia” se entiende incluir en el trabajo presentado como propio, partes hechas por otra persona. En caso que corresponda a “copia” a otros estudiantes, la sanción anterior se aplicará a todos los involucrados. En todos los casos, se informará a la Dirección de Pregrado de la Escuela de Ingeniería para que tome sanciones adicionales si lo estima conveniente.

También se entiende por copia extraer contenido sin modificarlo sustancialmente desde fuentes digitales como Wikipedia o mediante el uso de asistentes inteligentes como ChatGPT, Gemini o Copilot. Se entiende que una modificación sustancial involucra el análisis crítico de la información extraída y en consecuencia todas las modificaciones y mejoras que de este análisis se desprendan. Cualquiera sea el caso, el uso de fuentes bibliográficas, digitales o asistentes debe declararse de forma explícita, y debe indicarse cómo el/la estudiante mejoró la información extraída para cumplir con los objetivos de la actividad evaluativa.

Obviamente, está permitido usar material disponible públicamente, por ejemplo, libros o contenidos tomados de Internet, **siempre y cuando se incluya la referencia correspondiente**.

Lo anterior se entiende como complemento al Reglamento del Estudiante de la Pontificia Universidad Católica de Chile (<https://registrosacademicos.uc.cl/reglamentos/estudiantiles/>). Por ello, es posible pedir a la Universidad la aplicación de sanciones adicionales especificadas en dicho reglamento.

Compromiso del Código de Honor

Este curso suscribe el Código de Honor establecido por la Universidad, el que es vinculante. Todo trabajo evaluado en este curso debe ser propio. En caso que exista colaboración permitida con otros/as estudiantes, el trabajo deberá referenciar y atribuir correctamente dicha contribución a quien corresponda. Como estudiante es un debe conocer el Código de Honor (<https://www.uc.cl/codigo-de-honor/>).