

- **Viaje de Aprendizaje:** Este sprint fue básicamente un safari de APIs. Nos pasamos la semana probando, descartando, frustrándonos y volviendo a empezar. Tuvimos que aprender a leer documentación que parecía escrita en otro idioma, entender formatos de respuesta, límites de uso, si pedían tarjeta de crédito, si daban suficientes requests gratuitos, etc. Lo loco es que, aunque arrancamos medio perdidos, empezamos a darnos cuenta de qué necesitábamos realmente: buena información de vuelos, sin tarjeta de crédito, gratis, sin trabas. Después de comparar de todo incluyendo APIs que parecían buenas hasta otras que te daban solo apenas el origen y destino del vuelo para conseguir una key terminamos quedándonos inicialmente con AviationStack, que era la que mejor balanceaba todo, pero luego llegamos a la conclusión que debido a la gran cantidad de información que ésta nos otorgaba respecto a las otras APIs, no podíamos dejar pasar esta oportunidad, así que Facu puso su tarjeta de crédito y enseguida pudimos obtener las credenciales para ponerla a funcionar. Acá la IA fue nuestro constante copiloto. Chat GPT, Gemini, Copilot... todos ayudándonos a destrabar errores, comparar opciones y entender las cosas que la documentación no explicaba bien. Y así llegaba el sprint 2, en este sprint dejamos de pensar solo en “hacer que funcione” y empezamos a diseñar el proyecto con cabeza. Decidimos armar una estructura de clases inspirada en cómo funciona realmente la aviación: Airline, Aircraft, Airport y Flight, cada una con una responsabilidad clara. La API devolvía demasiados datos, así que tuvimos que elegir qué era esencial para no complicar el código más de lo necesario. También nos tocó pelear con Git, sobre todo con git push, y tuvimos que aprender a trabajar en equipo sin romper todo. Con práctica, discusiones y ayuda de IA logramos organizar bien el repositorio y dejar una base sólida para el resto del proyecto. Este sprint nos enseñó que programar no es solo escribir código, sino también tomar buenas decisiones de diseño desde el principio. Así pasamos del sprint 2 al sprint 3. Acá sentimos el primer gran salto. Dejamos de trabajar con un JSON local y empezamos a consumir datos reales desde la API. Aplicamos herencia y polimorfismo para diferenciar vuelos comerciales y militares, usando como base la clase padre Vuelo, lo que volvió el código mucho más ordenado y flexible. También incorporamos manejo de errores con try/except para evitar que el programa se rompiera cada vez que la API fallaba o devolvía respuestas vacías. El sistema ahora podía normalizar horarios, identificar aerolíneas, filtrar información y hasta determinar cosas como cuál era el vuelo más largo. La IA ayudó a optimizar ideas, pero las decisiones importantes —la arquitectura, la escalabilidad, la forma de procesar los datos— salieron de nuestras propias discusiones y razonamiento. En este sprint el proyecto dejó de ser algo “de juguete” y empezó a sentirse como un sistema real para lo que es nuestras capacidades y realidad. Acercándonos ya al final de este largo camino, nos chocamos con el problema de cómo mostrar toda la información que teníamos y podíamos ver en el backend pero a la vez sin hacer algo muy complejo y que excediera nuestros conocimientos y capacidades, así fue como llegamos al sprint 4. En este sprint nos enfocamos en que todo lo que teníamos en el backend pudiera verse de manera clara y amigable en una interfaz web. Armamos un frontend con HTML y Tailwind que mostraba los vuelos en forma de tabla con un estilo bastante prolijo. Resolver rutas, puertos ocupados y pequeños errores fue parte del proceso, pero lo pudimos ir solucionando con organización y como no, con algo de ayuda de la IA. También limpiamos el backend, sacando funciones que ya no tenían sentido y quedándonos solo con las necesarias para procesar los datos antes de enviarlos al frontend. Al principio pensábamos mostrar todo de manera muy básica, y de repente ya teníamos una interfaz que

permitía ver datos reales de forma cómoda. Fue un sprint donde sentimos un avance enorme y más concreto. Para finalizar, el último sprint fue más reflexivo que técnico. Nos aseguramos de que todo lo que estábamos mostrando respetara los términos de uso de la API, la privacidad de los datos y las buenas prácticas. Decidimos no hacer llamadas en bucle y mantener la actualización manual para no sobrepasar los límites de consultas, no mostrar información sensible o táctica de vuelos militares, no guardar logs de usuarios y dejar claro cuando un dato no podía identificarse. También revisamos sesgos geográficos, ya que nuestro sistema estaba muy optimizado para aerolíneas de Latinoamérica, y agregamos medidas como mostrar “Desconocido” en vez de inventar información. Este sprint nos obligó a pensar como desarrolladores responsables, entendiendo que trabajar con datos reales implica decisiones éticas más allá del código. De cara a futuro dejamos la puerta abierta a de ser necesario o apetecible para el grupo poder implementar nuevas funcionalidades como lo son el botón de exportar todos los datos en forma de csv, y más cambios y funcionalidades del estilo.