

# ASISTENTES COGNITIVOS EN INGENIERÍA DE SISTEMAS

DR. ALEJANDRO SALADO  
UNIVERSIDAD DE ARIZONA

DR. DANIEL SELVA  
UNIVERSIDAD DE TEXAS A&M

Los desarrollos en inteligencia artificial y la potencia de computación disponible hoy en día están permitiendo el desarrollo de asistentes cognitivos para tareas de ingeniería. Un asistente cognitivo es una máquina que actúa como un ingeniero virtual, un compañero **senior** con el que el ingeniero humano trabaja y que ejecuta tareas de ingeniería que los humanos no son capaces de llevar a cabo de manera efectiva o eficiente. Este artículo presenta como ejemplos dos asistentes cognitivos que están actualmente en fase de desarrollo, Daphne y Houston, que tienen los propósitos de ayudar a los ingenieros humanos en la arquitectura de sistemas y la derivación de requisitos, respectivamente. Si bien estos ingenieros virtuales tienen un gran potencial para la mejora de los resultados de las actividades de ingeniería, aún queda camino por recorrer. Son necesarios tanto avances tecnológicos para garantizar buenas bases de conocimiento con las que el asistente cognitivo pueda trabajar, como estudios que permitan conocer efectos negativos que el apoyo cada vez mayor en tecnología pueda tener en las habilidades de los humanos.

---

## PALABRAS CLAVE •

asistentes cognitivos, ingeniero virtual, sistemas expertos, sistemas inteligentes

## CÓMO CITAR ESTE ARTÍCULO •

Salado, Alejandro y Selva, Daniel. 2021. "Asistentes cognitivos en ingeniería" en: UEM STEAM Essentials

---

## INTRODUCCIÓN

Los asistentes cognitivos son máquinas, en el sentido amplio de la palabra, que ayudan a uno o más humanos en tareas cognitivas, de pensar. Actualmente, y cada vez más, los asistentes cognitivos proliferan tanto para tareas domésticas como profesionales. Alexa, Siri o Cortana son ejemplos muy conocidos. Podemos preguntarles acerca de casi cualquier tema, podemos pedirles que nos gestionen la agenda e incluso que nos recuerden hacer un recado si nos acercamos a un sitio. Pero estos asistentes cognitivos no son únicamente agendas pasivas; pueden ir un paso más allá. Por ejemplo, pueden observar el tráfico si saben que vamos a ir a trabajar para proactivamente sugerirnos

cuándo debemos salir de casa. El Editor de Microsoft Word revisa nuestros textos, sugiriendo, no ya correcciones ortográficas o gramaticales, sino también sugerencias para escribir de manera más efectiva. Puede detectar qué tipo de texto queremos (por ejemplo, un currículum) y sugerir plantillas y ordenar el contenido automáticamente conectando con nuestra cuenta de LinkedIn. De hecho, al escribir un correo electrónico uno de los autores de este artículo, le preguntaba el Editor si su estilo era utilizar comas antes de escribir "y" en una lista, para que le pudiera ofrecer sugerencias que se ajustaran a cómo le gusta escribir.

El uso de los asistentes cognitivos nos permite ahorrar esfuerzo mental en tareas que una máquina puede hacer fácilmente, de forma que podamos dedicar nuestra capacidad mental a tareas más placenteras, tareas que una máquina no puede realizar tan bien como nosotros o, simplemente, tareas para las aún no confiamos en la máquina. Al mismo tiempo, podemos confiar a las máquinas tareas para las que pueden proporcionar mejores resultados que los que nosotros podríamos obtener, como aquellas que requieren el procesamiento masivo de datos o que necesitan de una gran memoria de almacenamiento. Y un aspecto clave de los asistentes cognitivos es que hacen todo esto sin necesidad de que el humano tenga que aprender un código de comunicación. El humano puede dirigirse al asistente cognitivo tal y como lo hace con cualquier otra persona: “Cortana, recuérdame que compre paracetamol cuando esté cerca de una farmacia”, “Siri, envíale un correo electrónico a mi jefe diciéndole que llego tarde” o “Alexa, ¿qué película me recomiendas ver esta noche?”

Teniendo en cuenta los beneficios que obtenemos de este tipo de sistemas en nuestro día a día, algunos investigadores han comenzado a explorar cómo podemos utilizar asistentes cognitivos para ayudar al ingeniero en su trabajo. Con un asistente cognitivo que actúa como ingeniero virtual, el ordenador pasa de ser una herramienta en la que introducimos datos, creamos modelos y ejecutamos simulaciones, a un compañero de trabajo con mucha experiencia con el que co-diseñamos, a quien asignamos algunas de nuestras tareas, en quien nos apoyamos para entender mejor qué decisiones de ingeniería debemos tomar y quien se encarga de que el conocimiento que generamos en nuestro proyecto quede almacenado en la organización para que otros compañeros, y nosotros mismos, nos podamos beneficiar del mismo en futuros proyectos. En vez de evaluar opciones de diseño para un sistema, preguntaremos a nuestro compañero virtual “¿qué te parece este diseño?”, “¿cómo podemos mejorar la eficiencia de esta fuente de alimentación?”, “¿qué riesgo crees que asumimos si no ejecutamos este ensayo?” o “¿nos costaría mucho reemplazar esta interfaz por simplemente un único comando?”

Ahora bien, tal y como ocurre con la adopción de casi cualquier desarrollo tecnológico, el cambio de paradigma que supone la utilización de asistentes cognitivos posiblemente conllevará un deterioro de algunas capacidades actuales. Esta situación ya la experimentamos en nuestra vida cotidiana, por ejemplo, cuando nos sentimos perdidos sin nuestro GPS y somos incapaces de proporcionar o recordar indicaciones que nos dé alguien a pie de calle o cuando somos incapaces de recordar o memorizar rápidamente un número de teléfono si no tenemos nuestro móvil a mano. De la misma manera, tenemos que esperar que, conforme un ingeniero humano se apoye cada vez más en su compañero virtual, las capacidades asociadas a las tareas que deja de hacer vayan a disminuir. ¿Será esto problemático?

El tiempo dirá.

Nos adentramos pues en este artículo en el mundo de los ingenieros virtuales, asistentes cognitivos para ingenieros de sistemas. Comenzamos con un breve repaso histórico, en el que presentamos los sistemas expertos como precursores de los asistentes cognitivos. Después, presentamos dos ingenieros de sistemas virtuales que se encuentran en pleno desarrollo. El primero, Daphne (A. V. i. Martin & D. Selva, 2020), el más avanzado a día de hoy, ayuda al ingeniero humano a diseñar sistemas espaciales. El segundo, Houston (Salado & Tan, 2020), en un estado de desarrollo más prematuro, ayuda al ingeniero (humano) durante la definición de requisitos.

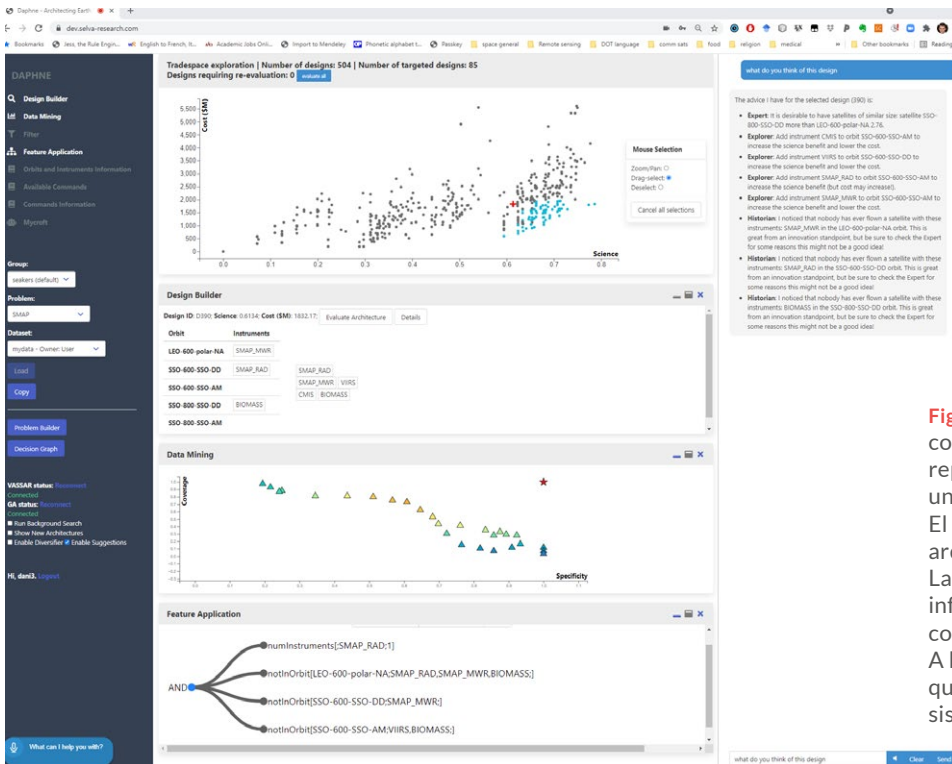
---

## UN POCO DE HISTORIA

En las décadas de 1970 y 1980 surgieron programas de ordenador que intentaban emular la capacidad de decisión de un humano, los sistemas expertos (Jackson, 1986). Utilizando un conjunto de reglas codificadas con estructuras de lógica simbólica y almacenadas en una base de datos, el sistema experto es capaz de recomendar soluciones a determinados problemas y, además, explicar por qué dicha solución es preferible. De esta manera, no era necesario tener conocimientos de informática especializados, sino que, una vez aprendida la forma de codificar conocimiento, un experto humano podía tanto incorporar nuevas reglas como utilizar el sistema experto. Muchos consideran a los sistemas expertos como los primeros ejemplos de éxito de la inteligencia artificial.

Si bien los sistemas expertos supusieron un gran avance en el apoyo a las decisiones, su estructura contaba con limitaciones importantes para escalarlos a problemas complejos. Por un lado, la codificación de conocimiento en forma de reglas funciona para áreas de conocimiento muy limitadas. Esto además hace que la adquisición de conocimiento sea compleja. Por otro lado, al aumentar el número de reglas se complica conseguir consistencia, es decir, evitar conflictos entre las reglas incluidas en la base de conocimiento. Estas limitaciones suponen problemas a la hora de escalar este tipo de sistemas.

La inteligencia artificial *moderna*, como la basada en redes neuronales, junto con la inmensa capacidad de almacenamiento y procesamiento de datos, han conseguido resolver estas limitaciones, dando paso a los asistentes cognitivos, eso sí, al precio de perder capacidad explicativa. Esto es, como en los asistentes cognitivos la generación y captura de conocimiento no depende únicamente de la codificación de reglas por parte de expertos humanos, sino que se pueden inyectar cantidades masivas de datos que el asistente cognitivo interpreta de manera autónoma, las recomenda-



**Figura 01** » Interfaz de usuario del asistente cognitivo Daphne. La nube de puntos representa el espacio de arquitecturas de una misión de observación de la Tierra. El usuario puede crear, modificar y evaluar arquitecturas en el Design Builder. La pestaña de Data Mining le proporciona información sobre las características comunes a las buenas arquitecturas. A la derecha está el cuadro de chat en el que Daphne dialoga con el ingeniero de sistemas.

ciones que el asistente cognitivo proporciona se basa en un complejo análisis de datos sin poder proporcionar una explicación lógica de cómo se ha seleccionado la recomendación. Además, los avances tecnológicos en procesamiento de lenguaje han cambiado la interacción entre el usuario y la máquina de forma dramática. Si en los sistemas expertos la interacción era rígida, los asistentes cognitivos interactúan imitando a los humanos, es decir, sin imponer estructuras de comunicación rígidas al humano.

## DAPHNE: EXPLORANDO ESPACIOS COMPLEJOS

Las primeras fases del desarrollo de un sistema complejo, en las que se definen los requisitos principales y se exploran diferentes arquitecturas para el sistema, tienen una importancia especial. Los estudios de exploración de alternativas y factibilidad que se realizan no requieren muchos recursos y sin embargo la mayoría del coste del ciclo de vida del sistema queda fijada después de esas etapas (Blanchard & Fabrycky, 1990). Además, estas decisiones se deben tomar a pesar de la gran incertidumbre que existe típicamente en estas fases, tanto en las necesidades de las partes interesadas como en los aspectos técnicos y programáticos de las tecnologías necesarias. A pesar de la importancia y complejidad de este proceso, las herramientas que se usan hoy en día para ayudar al ingeniero de sistemas están muy limitadas. Combinando herramientas de simulación y optimización se pueden explorar miles de soluciones (arquitecturas) distintas en espacios de decenas de dimensiones (Ross & Hastings, 2005), pero esto genera grandes cantidades de datos de los que es difícil sacar conclusiones claras, especialmente cuando el tiempo disponible para

estas actividades es muy limitado. Existen herramientas de visualización y análisis de datos que nos ayudan a navegar esos tsunamis de datos, pero no son suficientes para evitar la sobre carga cognitiva que lleva a fenómenos no deseables como la paralización en la toma de decisiones. Además, durante el proceso de analizar e interpretar los datos para llegar a la toma de decisiones, el ingeniero de sistemas necesita a menudo acceder a varias otras fuentes de información, como documentos técnicos relativos a sistemas y tecnologías parecidas a las del problema actual. Encontrar la información que se necesita de todas estas bases de datos se convierte a menudo en un cuello de botella que limita la duración del proceso o el alcance de sus objetivos.

Los asistentes cognitivos son una solución natural para este problema, ya que permiten extraer y analizar grandes cantidades de datos de varias fuentes de manera casi inmediata, y proporcionarle al ingeniero (humano) las respuestas que necesita directamente en lenguaje natural. Varios grupos de investigación estudian actualmente cómo los asistentes cognitivos pueden mejorar el proceso de diseño de la arquitectura de sistemas. Uno de los ejemplos más notables es el sistema Daphne (A. V. i. Martin & D. Selva, 2020), desarrollado por la universidad de Texas A&M para el diseño preliminar de misiones espaciales de observación de la Tierra. Integrado en una interfaz gráfica web que permite al usuario explorar el espacio de arquitecturas y realizar análisis de datos de manera interactiva, mostrada en la **Figura 01**, Daphne se presenta como un asistente que puede responder varios tipos de preguntas en inglés (por voz o chat), desde preguntas sobre misiones pasadas (por ejemplo, *¿qué misiones se usan actualmente para medir la humedad del suelo? ¿Cuál es la resolución espacial de esas misiones?*) a preguntas mucho más abiertas como *¿Qué te*

what do you think of this design

The advice I have for the selected design (390) is:

- **Expert:** It is desirable to have satellites of similar size: satellite SSO-800-SSO-DD more than LEO-600-polar-NA 2.76.
- **Explorer:** Add instrument CMIS to orbit SSO-600-SSO-AM to increase the science benefit and lower the cost.
- **Explorer:** Add instrument VIIRS to orbit SSO-600-SSO-DD to increase the science benefit and lower the cost.
- **Explorer:** Add instrument SMAP\_RAD to orbit SSO-600-SSO-AM to increase the science benefit (but cost may increase!).
- **Explorer:** Add instrument SMAP\_MWR to orbit SSO-600-SSO-AM to increase the science benefit and lower the cost.
- **Historian:** I noticed that nobody has ever flown a satellite with these instruments: SMAP\_MWR in the LEO-600-polar-NA orbit. This is great from an innovation standpoint, but be sure to check the Expert for some reasons this might not be a good idea!
- **Historian:** I noticed that nobody has ever flown a satellite with these instruments: SMAP\_RAD in the SSO-600-SSO-DD orbit. This is great from an innovation standpoint, but be sure to check the Expert for some reasons this might not be a good idea!
- **Historian:** I noticed that nobody has ever flown a satellite with these instruments: BIOMASS in the SSO-800-SSO-DD orbit. This is great from an innovation standpoint, but be sure to check the Expert for some reasons this might not be a good idea!

Figura 02 » Ejemplo de respuestas que Daphne proporciona a la pregunta ¿Qué piensas de este diseño?

parece esta arquitectura? Además de responder preguntas del ingeniero humano, Daphne también puede iniciar conversaciones de manera unilateral cuando cree que le pueden resultar útiles al ingeniero humano, como por ejemplo proporcionando sugerencias para mejorar una arquitectura en la que está trabajando actualmente. La Figura 02 proporciona ejemplos de respuestas que Daphne proporciona, como *Añade el instrumento CMIS a la órbita SSO-600-SSO-AM para incrementar el retorno científico y reducir coste* o *Me he dado cuenta de que nadie ha utilizado un satélite con el instrumento BIOMASS en la órbita SSO-800-SSO-DD. Esto parece fantástico desde un punto de vista de innovación, pero confirma con el módulo Experto por si por alguna razón no es una buena idea.*

El objetivo principal de Daphne es ayudar al ingeniero no sólo a explorar el espacio de arquitecturas más rápida y eficientemente, sino también entender los resultados de los análisis e identificar arquitecturas prometedoras para estudiar más a fondo. Entender los resultados es fundamental para poder justificar las decisiones a las partes implicadas. Naturalmente, si se van a invertir millones de euros (a menudo públicos) en una nueva misión, los gestores con el poder de tomar la decisión final se beneficiarán de, y en muchos casos exigirán, este tipo de justificación.

Para poder responder a estas preguntas y realizar todas sus funciones, Daphne usa procesos que están siempre activos en un segundo plano, buscando ayudar al ingeniero encontrando buenas arquitecturas o analizando los datos de las arquitecturas encontradas hasta el momento para comprobar si, por ejemplo, tienen algo en común. El es-

quema de la Figura 3 muestra la arquitectura de Daphne con el interfaz de usuario (front end) y la parte de servicio por detrás (back end). Un servidor web gestiona todas las preguntas y peticiones recibidas por Daphne. Para responder a una pregunta, el proceso es el siguiente. Primero, la pregunta pasa por un proceso estándar de procesamiento de texto (por ejemplo, extraer la raíz de cada palabra) para facilitar su interpretación. Después, el texto ya “limpio” se manda a un primer sistema de redes neuronales profundas cuyo objetivo es decidir cuál de varios “roles” es capaz de participar en la generación de la respuesta a la pregunta. Los roles son programas que se especializan en responder a ciertos tipos de preguntas. Se les dan nombres de ocupaciones para hacer referencia explícita a su especialidad. Por ejemplo, el “Historiador” es experto en responder a preguntas sobre misiones pasadas, el “Analista” se encarga de preguntas relativas al análisis de los datos del espacio de diseño, y el “Crítico” se dedica a identificar problemas con una arquitectura en particular y proporcionar consejos para mejorarla. Cuando la pregunta llega a uno de esos roles, entra en un segundo sistema de redes neuronales profundas cuyo objetivo es entender la intención de la pregunta y extraer los parámetros necesarios para responderla. Por ejemplo, para el Historiador, la pregunta en lenguaje natural se convierte en una consulta de una base de datos SQL que contiene todas las misiones pasadas y planeadas de Observación de la tierra que se conocen, obtenida de la web del Committee on Earth Observation Satellites (CEOS). Una vez obtenida la respuesta a esta consulta en la base de datos, se forma una respuesta completa en inglés con ella y se manda al ingeniero humano por chat y/o voz. Los distintos roles utilizan diferentes servicios (back-ends) y fuentes de datos (data sources) para generar las respuestas, como se puede ver en la Figura 03.

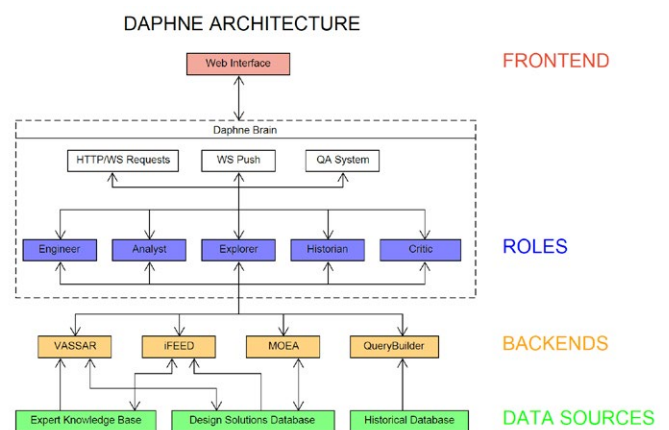


Figura 03 » Arquitectura del asistente cognitivo Daphne. La interfaz web se puede ver en la Figura 01. Un servidor web se encarga de gestionar todas las preguntas y mandarlas a un sistema de respuesta a preguntas (QA system) basado en un tipo de redes neuronales profundas llamadas transformers, que interpretan el tipo de pregunta y la mandan a los roles correspondientes. Estos extraen los parámetros relevantes usan varios microservicios y bases de datos para responder a las preguntas. Leyenda: Web Sockets (WS), Question Answering (QA), Value Assessment of System Architectures using Rules (VASSAR), Interactive Feature Extraction for Engineering Design (iFEED), Multi Objective Evolutionary Algorithm (MOEA)

Los roles también ejecutan servicios en segundo plano para detectar situaciones en las que sería positivo notificar al ingeniero de algo. Por ejemplo, si el ingeniero se pasa más de un minuto modificando un diseño en particular, eso se toma como una señal de que una intervención relevante podría ser bienvenida, y activa el rol de Crítico de Daphne para que le de sugerencias sobre cómo mejorar esa arquitectura en particular.

Varios estudios científicos realizados con Daphne, tanto con usuarios expertos (ingenieros del NASA Jet Propulsion Lab) como con principiantes (estudiantes de máster y doctorado en las universidades de Cornell y Texas A&M) han demostrado el potencial de tecnologías como Daphne para mejorar el rendimiento de los ingenieros de sistemas en las tareas de diseño preliminar de misiones espaciales (A. V. Martin & D. Selva, 2020; A. V. i. Martin & D. Selva, 2020; V. i. Martin & Selva, 2021). En esos estudios, se comparó el rendimiento de varios usuarios en tareas de exploración de espacios de arquitecturas con Daphne y sin Daphne. Daphne mejoró la calidad (prestaciones, costes, etc.) y cantidad de arquitecturas exploradas de manera significativa, y se observó que cuanto más se interactúa con Daphne, más se mejora el rendimiento. Por otra parte, esos estudios también revelaron lo que podría ser un potencial problema de la tecnología: no solo no se observó ningún cambio significativo en lo que el usuario aprende sobre el problema usando Daphne, sino que la tendencia parece ser negativa. Este resultado no es muy sorprendente si se tiene en cuenta lo que ha pasado en otras actividades que ahora se delegan a la inteligencia artificial, como por ejemplo la navegación. En el caso de la navegación, el hecho de que la gente ya no sepa navegar por las carreteras sin un navegador GPS no es un problema grave, pero en este caso podría serlo si uno sostiene que el objetivo principal de un sistema de apoyo a la toma de decisiones es ayudar al usuario a entender el problema, que va mucho más allá del número y la calidad de los diseños descubiertos por una herramienta.

## HOUSTON: CONOCIMIENTO COLECTIVO

Houston es un asistente cognitivo que ayuda al ingeniero humano en la derivación de requisitos. Los requisitos son condiciones que un sistema debe cumplir y se utilizan para formular problemas en proyectos de ingeniería (Salado, 2021). Derivar requisitos es una tarea clave en cualquier proyecto de ingeniería, ya que es muy difícil (de hecho prácticamente imposible) desarrollar una buena solución si se desconoce el problema a resolver o, incluso en muchos casos, si se intenta solucionar un problema inadecuado (Salado et al., 2017; Sols, 2014). De hecho, un porcentaje muy alto de los problemas que surgen durante el desarrollo de sistemas tienen su raíz en requisitos no identificados o erróneos.

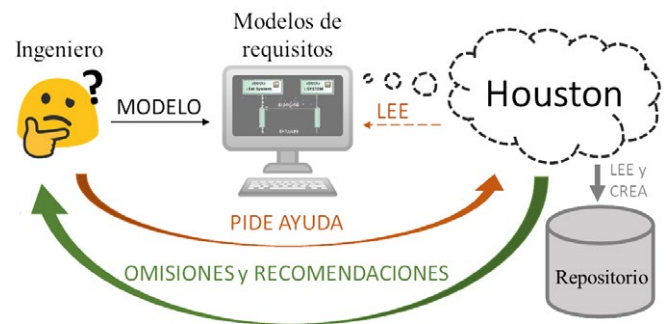


Figura 04 » Adaptada de (Salado & Tan, 2020). Funcionamiento básico de Houston.

Desafortunadamente, derivar requisitos es muy complicado (Hooks, 2010). Destacan dos razones. En primer lugar, es imposible demostrar que el pliego de requisitos está completo (Carson, 1998; Carson et al., 2004; Carson & Shell, 2001). No disponemos de ningún mecanismo que nos garantice que todos los requisitos que el sistema debe cumplir estén ahí. Todo lo contrario, la teoría nos dice que es imposible saber si están todos los necesarios. En segundo lugar, no disponemos de buenos métodos que nos permitan saber si existen conflictos o inconsistencias entre distintos requisitos que hagan imposible su cumplimiento (Salado & Nilchiani, 2017). Además, como los conflictos entre requisitos pueden darse entre subconjuntos, no necesariamente por pares (Salado & Nilchiani, 2016), y los sistemas suelen tener que satisfacer cientos o incluso miles de requisitos, es cognitivamente imposible para un humano explorar todo el espacio de conflictos posible para garantizar que el pliego está libre de inconsistencias. Houston está conceptualizado para ayudar en estas dos limitaciones, al menos inicialmente.

El funcionamiento básico de Houston se muestra en la **Figura 04**. Houston lee los requisitos que el ingeniero modela y los evalúa según su base de conocimiento. Si Houston encuentra una omisión potencial o una inconsistencia, entonces se los presenta al ingeniero (humano), quien decide qué hacer respecto a esas omisiones o inconsistencias. Houston captura las respuestas del ingeniero (humano) y las guarda en su repositorio de manera que pueda utilizar dicho conocimiento en el futuro.

Las omisiones se presentan siempre como *potenciales* y es el humano quien decide qué hacer con ellas. Como se ha explicado anteriormente, es imposible demostrar que los requisitos están completos. Por ello, el ingeniero virtual tampoco es capaz de confirmar que falte o no un requisito. Su trabajo consiste en sugerir posibles omisiones, basándose en su conocimiento y su capacidad de análisis. Pero depende del ingeniero humano confirmar que dicha omisión sea real, es decir, que sea realmente necesario que el sistema cumpla dicho requisito, y que merezca la pena

hacer un sistema que cumpla dicho requisito, ya que ambas requieren interactuar con stakeholders. Si bien un futuro Houston podría también tener dicha capacidad, ésta no está planificada actualmente.

El conocimiento de Houston se genera de tres fuentes principales: la estructura de los requisitos, buenas prácticas y normas de ingeniería y lecciones aprendidas en proyectos presentes y pasados.

Houston tiene preprogramado un conjunto de reglas estructurales que definen qué es un requisito. En este caso, se utilizan requisitos basados en modelos (Salado & Wach, 2019). Estas reglas indican qué constructos forman un requisito y qué relaciones pueden existir entre ellos. Dado entonces un requisito modelado con dichos constructos, Houston lee el modelo y *juego* con distintas formas de cambiar las relaciones en dichos constructos. Por ejemplo, imaginen un requisito que capture la siguiente idea:

*El sistema debe proporcionar señal C en menos de 3 s tras (i) recibir la señal B, (ii) siempre y cuando la señal B se haya recibido entre 2 s y 10 s tras haber recibido la señal A.*

Una vez modelado este requisito<sup>1</sup>, el ingeniero humano puede preguntarle al ingeniero virtual algo así como *Houston, ¿puede este escenario dar lugar a situaciones que no hemos anticipado?* Sin saber nada sobre las señales A, B, y C, Houston identifica escenarios potenciales que derivan de las señales y condiciones establecidas en el requisito. Por ejemplo:

*El sistema recibe la señal C más de 3 segundos después de recibir la señal B según la condición ii.*

*El sistema recibe la señal B en menos de 2 s después de recibir la señal A.*

*El sistema recibe la señal B tras pasar más de 10 s después de recibir la señal A.*

*El sistema recibe la señal B sin haber recibido la señal A.*

*El sistema recibe primero la señal B y luego la señal A.*

Y pregunta al ingeniero humano si quiere hacer algo al respecto para cada uno de esos escenarios, que debe decidir si es necesario establecer requisitos que definan cómo debe responder el sistema a estos escenarios o, por el contrario, no se restringe cómo el sistema debe reaccionar. Esto es importante, pues el hecho de que Houston identifique estos escenarios como posibles situaciones a las que el sistema se pueda enfrentar no implica que sean situaciones realistas para el problema que se esté tratando de resolver

o que se desee predefinir una reacción para el sistema. Es el ingeniero humano quien, estudiando el problema con los stakeholders, puede determinar que es necesario modelar requisitos adicionales. De este modo, Houston ayuda al ingeniero humano a explorar el espacio del problema de forma más exhaustiva.

Ahora bien, por muchas reglas estructurales que se implementen, esta capacidad sólo le permite a Houston ser un asistente cognitivo muy simple, del estilo de un sistema experto. Houston pasa a ser más sofisticado al añadirle capacidades interpretativas. Esto es, Houston no sólo lee la estructura del requisito, sino que comprende qué implican los distintos constructos en el modelo. Por ejemplo, en vez de leer simplemente señales A, B y C, Houston comprende que la señal C es una señal crítica de seguridad, la señal B actúa como un comando de activación de la señal crítica y la señal A es una señal de armado, sabiendo que un estándar obliga a siempre utilizar dos señales independientes para activar una señal crítica de seguridad. Así, si un ingeniero humano pide a Houston que evalúe el siguiente requisito:

*El sistema debe proporcionar señal C en menos de 3 s tras recibir la señal B, donde la señal C es una señal crítica de seguridad.*

Houston le sugiere que explore la necesidad de incorporar una señal de armado antes de la señal B como condición para proporcionar la señal C, e incluso le puede explicar que esta sugerencia se debe a un estándar que lo obliga o a múltiples experiencias previas capturadas en una guía de diseño.

Houston adquiere esta capacidad gracias a una ontología que permite capturar la información de fuentes como estándares, guías de diseño o bases de datos de lecciones aprendidas, y asociarla a los constructos de los modelos de requisitos. En el futuro, Houston podría codificar leyes naturales para detectar posibles inconsistencias entre constructos de los modelos de los requisitos (Kannan, 2021). Por ejemplo, dados los siguientes requisitos:

*El sistema debe transmitir una señal de XW.*

*El sistema debe observar la imagen A (infrarroja) con una resolución mejor de Y m.*

Houston sería capaz de relacionar automáticamente que la transmisión de una señal de alta potencia (información que se deriva del nivel de potencia en el requisito) conlleva (por lo general) una alta disipación de energía térmica, que conlleva a su vez (por lo general) una alta temperatura y que la temperatura está negativamente asociada con la resolución en la banda infrarroja. Así, Houston puede detectar un conflicto potencial en los requisitos incluso antes de que se comience a diseñar una solución (Salado

<sup>1</sup> Por simplicidad, no se proporcionan los modelos formales en este artículo.

& Nilchiani, 2017). Si bien este ejemplo es sencillo, el valor del ingeniero virtual radica en su capacidad para procesar un gran número de condiciones y relaciones en un tiempo considerablemente menor del que le llevaría a un humano. De nuevo, Houston actúa como un asesor, pues el conflicto se presenta como caso potencial a resolver por el ingeniero humano.

En su siguiente nivel de sofisticación, aún por desarrollar, Houston aprende. La idea fundamental es que Houston monitoriza el desarrollo del sistema con respecto a los requisitos que se han establecido, como por ejemplo re-

quisitos que se añaden durante el desarrollo del sistema o no conformidades y peticiones de desviaciones, junto con sus resultados. Utilizando la ontología establecida, Houston puede automáticamente desarrollar su propia base de conocimiento. De esta manera, Houston actúa verdaderamente como un acumulador del conocimiento de una organización, facilitando que el conocimiento colectivo que la organización genera durante su vida se utiliza en todos sus proyectos.

## CONCLUSIÓN

**Daphne:** *Andrea, podemos abrir un espacio de soluciones interesante si relejamos o simplemente quitamos la restricción X. Encuentro opciones con mucho menor coste de desarrollo, así como mejores prestaciones de Y y Z.*

**Andrea (humana):** *Houston, ¿puedes evaluar si asumimos algún riesgo al quitar el requisito X?*

**Houston:** *Andrea, no recomiendo quitar el requisito X. Según los últimos 12 proyectos que hemos desarrollado, el requisito X es fundamental para evitar problemas de integración en el futuro.*

**Andrea:** *Houston, ¿a qué te referes con problemas de integración?*

**Houston:** *Sin la restricción X, los proyectos P1 a P7 sufrieron retrasos de entre 10-25% debido a incompatibilidades entre equipos de tipo T1 y T2. Los proyectos P7 a P12, que implementaron dicha restricción, no presentaron dichas incompatibilidades.*

**Andrea:** *Daphne, ¿hay alguna arquitectura en ese espacio nuevo de soluciones que no utilice equipos de tipos T1 y T2 simultáneamente?*

**Daphne:** *La verdad que no muchos, pero las arquitecturas A122 y A451 no utilizan equipos de tipo T2.*

**Andrea:** *Houston, acepta quitar el requisito X. Daphne, vamos a explorar más a fondo A122 y A451.*

¿Se imaginan su día a día en el trabajo así? Sin duda, este diálogo suena a ciencia ficción. Aún estamos lejos, pero, como se ha mostrado en este artículo, no sólo se han empezado a dar los primeros pasos en esta dirección, sino que disponemos de la tecnología necesaria para conseguirlo. De la misma manera en la que los asistentes cognitivos se introducen cada vez más en nuestra vida para ayudarnos

con tareas cotidianas, los ingenieros virtuales pasarán a formar parte de equipos de ingeniería en el futuro. Los asistentes cognitivos pueden suponer un cambio de paradigma radical en la forma en la que trabaja un ingeniero. Si los ordenadores han permitido dedicar más tiempo a modelar y menos a calcular, nuestros compañeros virtuales nos pueden permitir invertir menos tiempo en ejecución y más tiempo en pensar: entender y enfocar el problema, planear estrategias o validar soluciones.

Los asistentes cognitivos no se limitarán a la ingeniería de sistemas, pues probablemente tienen un potencial similar para cualquier disciplina. Por ejemplo, Daphne se ha utilizado para diseñar el ascenso, descenso y aterrizaje de sistemas espaciales (León et al., 2019) y hay conceptos para modificar Houston de forma que ayude a evaluar la seguridad de sistemas. Se puede pensar también en selección de materiales, identificación de componentes electrónicos, diseño de placas de circuito impreso, rutado de cableados o tuberías... Las posibilidades son muchas.

En cualquier caso, queda muchísimo trabajo por delante en al menos tres planos. En primer lugar, a nivel tecnológico en aspectos como la captura y actualización de conocimiento, el aprendizaje autónomo, la precisión y corrección de las recomendaciones o la integración de distintos asistentes cognitivos. Esto requiere también el desarrollo de procesos de captura y almacenamiento de cantidades masivas de datos directamente de los proyectos de ingeniería en desarrollo. En segundo lugar, a nivel de rendimiento de equipos híbridos, donde se necesita entender cómo se genera la confianza entre un humano y un ingeniero virtual, cómo puede el ingeniero virtual adelantarse o predecir lo que el humano sabe y quiere a la hora de dar respuestas y recomendaciones, o cómo gestionar posibles dependencias entre individuos y máquinas. Por último, se necesitan explorar las consecuencias no intencionadas para el ingeniero humano de confiar y utilizar asistentes cognitivos en cuanto a la evolución de sus competencias, así como de la formación, capacitación y refresco. La investigación es clave para afrontar estos retos.

## REFERENCIAS BIBLIOGRÁFICAS

Blanchard, B. S., & Fabrycky, W. J. (1990). *Systems engineering and analysis (Vol. 4)*. Prentice Hall New Jersey.

Carson, R. S. (1998). *Requirement completeness: a deterministic approach*. INCOSE International Symposium, 8, 738-746.

Carson, R. S., Aslaksen, E., & Gonzales, R. (2004). *Requirements completeness*. INCOSE International Symposium.

Carson, R. S., & Shell, T. (2001). *Requirements completeness: absolute or relative?* Systems Engineering, 4, 230-231.

Hooks, I. F. (2010). *Why Johnny still can't write requirements* 22nd Annual SSTC Conference, Salt Lake City, UT, USA.

Jackson, P. (1986). *Introduction to expert systems*. Addison-Wesley Pub. Co., Reading, MA. <https://www.osti.gov/biblio/5675197>

Kannan, H. (2021). *Formal reasoning of knowledge in systems engineering through epistemic modal logic*. Systems Engineering, 24(1), 3-16. <https://doi.org/https://doi.org/10.1002/sys.21563>

León, S. S. D., Selva, D., & Way, D. W. (2019, 2-9 March 2019). *A Cognitive Assistant for Entry, Descent, and Landing Architecture Analysis*. 2019 IEEE Aerospace Conference.

Martin, A. V., & Selva, D. (2020). *From Design Assistants to Design Peers: Turning Daphne into an AI Companion for Mission Designers*. In AIAA Scitech 2019 Forum. <https://doi.org/10.2514/6.2019-0402>

Martin, A. V. i., & Selva, D. (2020). *Daphne: A Virtual Assistant for Designing Earth Observation Distributed Spacecraft Missions*. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 13, 30-48. <https://doi.org/10.1109/JSTARS.2019.2948921>

Ross, A. M., & Hastings, D. E. (2005). 11.4.3 *The Tradespace Exploration Paradigm*. INCOSE International Symposium, 15(1), 1706-1718. <https://doi.org/10.1002/j.2334-5837.2005.tb00783.x>

Salado, A. (2021). *A systems-theoretic articulation of stakeholder needs and system requirements*. Systems Engineering, 24, 83-99. <https://doi.org/https://doi.org/10.1002/sys.21568>

Salado, A., & Nilchiani, R. (2016). *The Concept of Order of Conflict in Requirements Engineering*. IEEE Systems Journal, 10(1), 25-35. <https://doi.org/10.1109/JSYST.2014.2315597>

Salado, A., & Nilchiani, R. (2017). *The Tension Matrix and the Concept of Elemental Decomposition: Improving Identification of Conflicting Requirements*. IEEE Systems Journal, 11(4), 2128-2139. <https://doi.org/10.1109/JSYST.2015.2423658>

Salado, A., Nilchiani, R., & Verma, D. (2017). *A contribution to the scientific foundations of systems engineering: Solution spaces and requirements* [journal article]. Journal of Systems Science and Systems Engineering, 26(5), 549-589. <https://doi.org/10.1007/s11518-016-5315-3>

Salado, A., & Tan, R. (2020). *Structural Rules for an Intelligent Advisor to Identify Requirements Gaps using Model-Based Requirements*. IEEE International Conference on Systems, Man and Cybernetics (SMC), Toronto, Canada.

Salado, A., & Wach, P. (2019). *Constructing True Model-Based Requirements in SysML*. Systems, 7(2), 19.

Sols, A. (2014). *Systems Engineering. Theory and Practice*. Universidad Pontificia de Comillas.

Viros Martin, A., & Selva, D. (2021). *Learning comes from Experience: The Effects on Human Learning and Performance of a Virtual Assistant for Design Space Exploration Design Computing and Cognition 20*.



## BIOGRAFÍA

Dr. Alejandro Salado cuenta con más de 15 años de experiencia como ingeniero, consultor, investigador y formador en ingeniería de sistemas. Actualmente, es profesor titular de ingeniería de sistemas en la Universidad de Arizona y ha trabajado con organizaciones como el U.S. Army, Apple, Thales Alenia Space y Airbus Defense and Space, entre otras. Su investigación se centra en la formulación de problemas, el diseño de estrategias de verificación y validación, la ingeniería de sistemas basada en modelos y la formación de ingenieros. Alejandro es autor de más de 100 publicaciones y su investigación ha sido financiada, entre otros, por el National Science Foundation (NSF), el Naval Surface Warfare Command (NSWC), el Naval Air System Command (NAVIR) y el Office of Naval Research (ONR), además de recibir varios premios, como el NSF CAREER Award y el International Fulbright Science and Technology Award. Alejandro es ingeniero de telecomunicaciones por la Universidad Politécnica de Valencia y cuenta con másteres en ingeniería electrónica y en gestión de proyectos por la Universidad Politécnica de Cataluña, máster en ingeniería de sistemas espaciales por la Universidad Técnica de Delft y doctorado en ingeniería de sistemas por el Instituto Tecnológico Stevens.

Dr. Daniel Selva es Profesor Ayudante Doctor (Assistant Professor) en el departamento de Ingeniería Aeroespacial de la Universidad Texas A&M. Su grupo de investigación (Systems Engineering, Architecture, and Knowledge o SEAK Lab) se dedica a estudiar aplicaciones de inteligencia artificial para mejorar el diseño y operación de misiones espaciales. Daniel es Ingeniero Superior de Telecomunicaciones por la Universidad Politécnica de Catalunya e Ingeniero Aeroespacial por la Universidad de Supaero en Toulouse, Francia. Después de trabajar cuatro años como especialista en Aviónica del cohete Ariane 5 para Arianespace en la Guayana francesa, Daniel hizo un doctorado en ingeniería de sistemas espaciales en el MIT. Antes de empezar en la Universidad Texas A&M, Daniel fue Profesor Ayudante Doctor en la Universidad de Cornell durante 4 años.