

# Cápsula 1: Validación y tipos de resolución

Hola, bienvenidxs a una cápsula del curso Visualización de Información. En esta hablaré sobre validación y acercamientos de resolución de problemas de visualización.

Este grupo de cápsulas está orientado a integrar el modelo de trabajo que hemos revisado en el curso, a modo de que recordemos la panorámica de esta propuesta de guía y consideremos cómo utilizarlo mejor en la práctica.

Específicamente hablaremos sobre validar soluciones para problemas de visualización. Con eso me refiero a la capacidad de aprobar o refutar una solución o partes de ella según criterios claros.

Esto fue un aspecto que nos cuestionamos al comienzo del curso, como una actividad no trivial si no se cuenta con experiencia previa en el tema. Como el espacio de posibilidades de visualización es tan grande, la idea de realmente validar una solución parecía difícil.

¡Y es difícil! Pero en ese sentido también nos ayuda nuestro modelo de trabajo utilizado.

Como hemos mencionado, tal vez demasiadas veces, el modelo anidado separa los procesos a realizar cuando atacamos un problema de visualización en cuatro niveles. Esta separación también ayuda a dividir momentos y criterios de validación en estos distintos niveles. Nos ayuda a cuestionarnos aspectos específicos en vez de la totalidad de una solución.

Por otro lado, el **tipo de resolución** para un problema de visualización puede diferir instancia a instancia.

En la generalidad de este curso hemos asumido un **enfoque centrado en un problema o necesidad**, que cuenta con una situación y datos específicos, donde hay tareas a ser resueltas por uno o varios usuarios. Donde además puede que no haya una visualización existente previamente, y puede concebirse por primera vez al resolver el problema.

Para este enfoque el modelo de trabajo funciona “de arriba hacia abajo”, donde comenzamos en los primeros niveles: caracterizando el dominio y situación, para luego abstraer dicho dominio. Luego se pasa a las etapas de codificación e implementación que perfectamente pueden tomar *idioms* e implementaciones ya existentes en el conocimiento común, pero que aún falta conectar con la necesidad específica a resolver.

Nos enfocamos en este tipo de enfoque por su aplicabilidad en variados contextos donde la Visualización de Información puede aportar y ayudar a personas a realizar algo o contar una historia. Pero hay otros tipos de enfoques que se pueden considerar inversos, o de “abajo hacia arriba” en el modelo.

Uno de estos es el **enfoque centrado en la técnica**, donde se trabaja principalmente en los niveles de codificación e implementación, y el objetivo es inventar nuevos *idioms* o implementaciones que aporten mejores resultados a dominios y abstracciones específicas.

Podemos pensar en Mike Bostock y en todos los contribuidores de D3.js, cuyo gran listado de utilidades existen con ese propósito: apoyar y mejorar la creación de *idioms* de visualización. Incluso llega al punto de que es un poco agnóstica al dominio o abstracción, por la gran cantidad de casos que cubre.

El modelo de trabajo revisado es compatible ambos enfoques, ya que este permite un avance no lineal entre niveles, además de iteración entre niveles al encontrar problemas en el camino. Particularmente se enfoca más en los niveles de abstracción y codificación, lo cual nos entrega conceptos para trabajar en cualquier sentido.

Para enfoques centrados en problemas, ayuda a trabajar hacia abajo buscando y filtrando posibilidades existentes o inexistentes, en base a decisiones de diseño apropiadas a las abstracciones identificadas.

Para enfoques centrado en la técnica, permite trabajar hacia arriba para clasificar la nueva propuesta de *idiom* o implementación dentro del rango de decisiones de diseño existentes. Lo cual, permite compararle con propuestas existentes y discutir posibles relaciones con ellas.

Con eso termina el contenido de esta cápsula. Recuerda que si tienes preguntas, puedes dejarlas en los comentarios del video para responderlas en la sesión en vivo de esta temática. ¡Chao!