

Lev Manovich

El software toma el mando



La imagen de arriba no pertenece a la obra original

Esta versión

20 de Noviembre 2008

Por favor tome en cuenta que esta versión no ha sido revisada y también le faltan ilustraciones.

Largo: 82.071 palabras (incluyendo notas al pie)

Licencia de Creative Commons

El software toma el mando Por Lev Manovich está bajo una licencia de Atribución- No comercial- No derivados 3.0 de Estados Unidos

Sobre las versiones

Una de las ventajas de la distribución online a la que puedo controlar es que no tengo que fijar permanentemente el contenido del libro. Como los servicios de software y web contemporáneos, el libro puede cambiar tantas veces como yo quiera, con nuevos “rasgos” y “arreglos de bugs” agregados periódicamente. Planeo tomar ventaja de estas posibilidades. De tanto en tanto, iré agregando nuevo material y haciendo cambios y correcciones al texto.

Última versión:

Busquen en www.softwarestudies.com/softbook la última versión del libro.

Sugerencias, correcciones y comentarios:

Enviarlos a manovich@ucsd.edu

con la palabra “softbook” en el encabezado

La presente traducción fue realizada por Adrián Yalij y Sofía Alamo, (Cátedra Procesamiento de Datos, Carrera de Ciencias de la Comunicación, Facultad de Ciencias Sociales, Universidad de Buenos Aires)

Introducción: Estudios de Software para principiantes

Software, o el motor de las sociedades contemporáneas

A principios de los 1990s, las marcas globales más famosas eran las compañías que estaban en el negocio de producir bienes materiales o procesar materia física. Hoy, sin embargo, las listas de las marcas globales más reconocidas están encabezadas con nombres tales como Google, Yahoo y Microsoft. (De hecho, Google era la número uno del mundo en el 2007 en términos de reconocimiento de marca). Y, al menos en los Estados Unidos, los diarios y revistas más leídos - New York Times, USA Today, Business Week, Etc - todos los días publican noticias e historias sobre Youtube, MySpace, Facebook, Apple, Google y otras compañías IT.

¿Qué hay de otros medios? Si acceden al sitio web de la CNN y navegan a la sección de negocios verán datos de mercado de sólo diez compañías e índices expuestos en la misma página principal. Aunque la lista cambia a diario, siempre tiende a incluir algunas de las mismas marcas IT. Tomemos el 21 de Enero de 2008 como ejemplo. En ese día la lista de CNN consistía en las siguientes compañías e índices: Google, Apple, índice S&P 500, índice compuesto Nasdaq, promedio industrial Dow Jones, Cisco Systems, General Electric, General Motors, Ford, Intel.

Esta lista es muy reveladora. Las compañías que tratan con bienes físicos y energía aparecen en la segunda parte de la lista: General Electric, General Motors, Ford. A continuación tenemos dos compañías IT que proveen hardware: Intel hace chips de computadoras, mientras que Cisco hace equipamiento de redes. ¿Qué hay de las dos compañías que están en la cabecera. Google y Apple? La primera parece estar en el negocio de la información, mientras que la segunda está haciendo artefactos electrónicos para consumidores: laptops, monitores, reproductores de música, etc. Pero de hecho ambas en realidad están haciendo algo más. Y aparentemente, este algo más es tan crucial para el funcionamiento de la economía de EEUU - y en consecuencia, la del mundo global también- que estas compañías aparecen de manera casi diaria en las noticias de negocios. Y las mayores compañías de internet que también aparecen diariamente en las noticias - Yahoo, Facebook, Amazon, eBay - ellas también están en el mismo negocio.

Este “algo más” es **software**. Motores de búsqueda, sistemas de recomendación, aplicaciones de mapeo, herramientas de blog, herramientas de subastas, clientes de mensajería

instantánea, y por supuesto, plataformas que permiten a otros escribir nuevo software - Facebook, Windows, Unix, Android - están en el centro de la economía, cultura, vida social y, cada vez más, en la política globales. Y este “software cultural” - cultural en el sentido de que es usado por cientos de millones de personas y que acarrea “átomos” de cultura (medios e información, además de interacción humana en torno a estos medios e información) - es sólo la parte visible de un universo de software mucho más amplio.

El software controla el vuelo de un misil inteligente hacia su blanco en la guerra, ajustando su curso a lo largo de su vuelo. El software maneja las bodegas y líneas de producción de Amazon, Gap, Dell, y numerosas otras compañías permitiéndoles ensamblar y despachar objetos materiales alrededor del mundo, casi en nada de tiempo. El software permite que tiendas y supermercados reabastezcan sus góndolas automáticamente, así como determina automáticamente qué ítems deben ir en oferta, por cuánto y cuándo y en qué lugar de la tienda. El software, por supuesto, es el que organiza la internet, ruteando mensajes de email, entregando páginas web de un servidor, controlando el tráfico de red, asignando direcciones IP, y representando páginas web en un navegador. La escuela y el hospital, la base militar y el laboratorio científico, el aeropuerto y la ciudad - todos los sistemas sociales, económicos y culturales de la sociedad moderna - corren en base a software. El software es el pegamento invisible que lo une todo. Mientras que varios sistemas de la sociedad moderna hablan en diferentes idiomas y tienen metas diferentes, todos comparten la sintaxis del software: declaraciones de control “si/entonces” y “mientras/haz”, operadores y tipos de datos incluyendo caracteres y números de punto flotante, estructuras de datos tales como listas, y convenciones de interfaz que abarcan menús y cajas de diálogo.

Si la electricidad y el motor de combustión hicieron a la sociedad industrial posible, el software de manera similar habilita a la sociedad global de la información. Los “trabajadores cognitivos”, los “analistas de símbolos”, las “industrias creativas” y las “industrias de servicios” - todos estos jugadores económicos claves de la sociedad de la información no pueden existir sin software. El software de visualización de datos usado por un científico, el software de planillas de cálculos usado por un analista financiero, el software de diseño web usado por un diseñador que trabaja para una transnacional publicitando energía, el software de reservaciones usado por una aerolínea, etc. El software es también lo que impulsa el proceso de globalización, permitiéndole a las compañías distribuir los nodos de administración, instalaciones de producción y almacenaje y consumos alrededor del mundo. Sin importar en qué nueva dimensión de la existencia contemporánea cualquier teoría social de las últimas décadas se haya decidido enfocar - sociedad de la información, sociedad del conocimiento, sociedad en red - todas estas nuevas dimensiones están habilitadas por el software.

Paradójicamente, mientras que científicos sociales, filósofos, críticos culturales y teóricos de medios y nuevos medios aparentan haber cubierto ya todos los aspectos de la revolución IT, creando un número de disciplinas tales como cibercultura, estudios de internet, teoría de nuevos medios y cultura digital, el motor subyacente que impulsa todos estos temas - el software - ha recibido poca o nula atención directa. El software sigue siendo invisible para la mayoría de los académicos, artistas y profesionales culturales interesados en IT y sus efectos culturales y sociales. (Una excepción importante es el movimiento Open Source y temas relacionados en torno al copyright y a propiedad intelectual que han sido discutidos extensamente en muchas disciplinas académicas). Pero si limitamos las discusiones críticas a las nociones de “ciber”, “digital”, “internet”, “redes”, “nuevos medios” o “medios sociales”, nunca podremos captar lo que hay detrás de los nuevos medios representacionales y comunicacionales y comprender lo que realmente son y lo que realmente hacen. Si no atendemos al software mismo corremos el peligro de siempre tratar con sus efectos más que con las causas: la salida que aparece en una pantalla de computadora más que los programas y culturas sociales que producen esas salidas.

“Sociedad de la información”, “sociedad del conocimiento”, “sociedad de redes”, “medios sociales” - sin importar en qué rasgo de la existencia contemporánea se haya decidido enfocar una teoría social particular, todos estos rasgos nuevos son habilitados por el software. Es hora de que nos enfoquemos en el software mismo.

¿Qué son los “estudios de software”?

Este libro apunta a contribuir al paradigma intelectual en desarrollo de los “estudios de software”. ¿Qué son los estudios de software? Aquí hay algunas definiciones. La primera proviene de mi propio libro *El lenguaje de los nuevos medios* (completado en 1999; publicado por MIT Press en 2001), donde, hasta donde yo sé, los términos “estudios de software” y “teoría del software” aparecieron por primera vez. Yo escribí: “Los nuevos medios piden una nueva etapa en la teoría de medios cuyos comienzos pueden rastrearse hasta los trabajos revolucionarios de Robert Innis y Marshall McLuhan en los 1950s. Para comprender la lógica de los nuevos medios necesitamos recurrir a la ciencia de las computadoras. Es allí donde podemos esperar encontrar los nuevos términos, categorías y operaciones que caracterizan a los medios que se han vuelto programables. De los estudios de medios nos movemos a algo que podemos llamar estudios de software; de teoría de medios a la teoría del software”.

Leyendo esta declaración hoy, siento que son necesarios algunos ajustes. Se posiciona a la ciencia informática como una especie de verdad absoluta, una cosa dada que puede explicarnos cómo funciona la cultura en la sociedad del software. Pero la ciencia informática es en sí misma parte de la cultura. Por lo tanto, pienso que los Estudios de Software deben estudiar tanto el rol del software en la formación de la cultura contemporánea como además las fuerzas culturales, sociales y económicas que están formando el desarrollo del software mismo.

El libro que primero demostró la necesidad de un segundo enfoque de manera comprehensiva fue *New Media Reader* editado por Noah Wardrip-Fruin y Nick Montfort (The MIT Press, 2003). La publicación de esta innovadora antología creó el marco para el estudio histórico del software en tanto se relaciona con la historia de la cultura. Aunque el *Reader* no utilizaba explícitamente el término “estudios de software”, sí proponía un nuevo modelo para cómo pensar sobre el software. Al juxtaponer sistemáticamente textos importantes de pioneros de la computación cultural y de artistas claves activos en los mismos periodos históricos, el *Reader* demostraba que ambos pertenecen a las mismas epistemes mayores. Es decir, muchas veces la misma idea era articulada simultáneamente en el pensamiento tanto de artistas como de científicos que estaban inventando la computación cultural. Por ejemplo, la antología abre con la historia de Jorge Borges (1941) y el artículo de Vannevar Bush (1945) las cuales ambas contienen la idea de una estructura masiva ramificada como la mejor manera de organizar datos y para representar la experiencia humana.

En febrero de 2006 Matthew Fuller, quien ya había publicado un libro pionero sobre el software como cultura (*Behind the Blip, essays on the culture of software, 2003*) organizó el *primer taller de estudios de software* en el instituto Piet Zwart en Rotterdam. Presentando el taller, Fuller escribió: “El Software es muchas veces un punto ciego en la teorización y el estudio de los medios digitales computacionales y en red. Es el mismo suelo y el “relleno” del diseño de medios. En cierto sentido, toda obra intelectual es ahora un “estudio de software”, en tanto el software provee su medio y su contexto, pero hay pocos lugares donde la naturaleza específica, la materialidad del software sea estudiada, excepto como un asunto de ingeniería”.

Estoy completamente de acuerdo con Fuller en que “todo trabajo intelectual ahora es “estudio de software””. Pero va a hacer falta un tiempo hasta que los intelectuales se den cuenta. Al momento de escribir esto (Primavera 2008), los estudios de software son un nuevo paradigma para la investigación intelectual que recién ahora comienza a emerger. MIT Press está por publicar el primer libro que lleva este término en su título más adelante este mismo año

(*Software Studies: A Lexicon*, editado por Matthew Fuller). Al mismo tiempo, un número de trabajos ya publicados por los teóricos de medios más prominentes de nuestros tiempos - Katherine Hayles, Friedrich A. Kittler, Lawrence Lessig, Manuel Castells, Alex Galloway, y otros - pueden identificarse retroactivamente como pertenecientes a los “estudios de software”.

Por lo tanto, creo fuertemente que este paradigma ya ha existido por algunos años pero no fue nombrado explícitamente hasta ahora (en otras palabras, el estado de los “estudios de software” es similar a donde estaban los “nuevos medios” a principios de los 1990s).

En su introducción al taller de 2006 en Rotterdam Fuller escribió que “el software puede ser visto como un objeto de estudio y un área de práctica para la teoría del arte y del diseño y para las Humanidades, para estudios culturales y estudios de ciencia y tecnología y para una rama reflexiva de las ciencias informáticas”. Dado que una nueva disciplina académica puede ser definida a través de un objeto único de estudio, un nuevo método de investigación o una combinación de ambos ¿Cómo pensaremos sobre los estudios de software? La declaración de Fuller implica que “software” es un nuevo objeto de estudio que debe ser puesto en la agenda de disciplinas existentes y que puede ser estudiado con métodos ya existentes - por ejemplo, teoría de Red-Objeto, semiótica social o arqueología de medios.

Pienso que hay buenas razones para apoyar esta perspectiva. Pienso en el software como **una capa que permea todas las áreas de las sociedades contemporáneas**. Por lo tanto, si queremos comprender las técnicas contemporáneas de control, comunicación representación, simulación, análisis, toma de decisiones, memoria, visión, escritura e interacción, nuestro análisis no puede estar completo hasta que consideremos esta capa del software. Lo que significa que todas las disciplinas que tratan con la sociedad y cultura contemporánea - arquitectura, diseño, crítica de arte, sociología, ciencias políticas, humanidades, estudios de ciencia y tecnología, etc - necesitan dar cuenta del rol del software y sus efectos en cualquiera de los temas que investiguen.

Al mismo tiempo, el trabajo existente en estudios de software ya demuestra que si nos enfocamos en el software mismo necesitamos una nueva metodología. Esto es, ayuda practicar aquellos sobre lo que uno escribe. No es accidental que los intelectuales que han escrito de manera más sistemática sobre el rol del software en la sociedad y la cultura hasta ahora hayan sido programadores ellos mismos o se hubieran involucrado sistemáticamente en proyectos culturales que involucraban centralmente escribir un nuevo software: Katherine Hales, Matthew Fuller, Alexander Galloway, Ian Bogust, Geet Lovink, Paul D. Miller, Peter

Lunenfeld, Katie Salen, Eric Zimmerman, Matthew Kirschenbaum, William J. Mitchell, Bruce Sterling, etc. En contraste, los eruditos sin experiencia, tales como Jay Bolter, Siegfried Zielinski, Manuel Castells y Bruno Latour no han incluido consideraciones sobre el software en sus por otra parte muy influyentes relatos de medios y tecnologías modernas.

En la década actual, el número de estudiantes en arte de medios, diseño, arquitectura y humanidades que usan programación o *scripting* en sus trabajos ha crecido sustancialmente - al menos en comparación con 1999 cuando mencioné por primera vez los “estudios de software” en *The Language of New media*. Fuera de las industrias culturales y académicas, muchas personas más están escribiendo software también. En una medida significativa esto es el resultado de nuevos lenguajes de programación y *scripting* tales como Javascript, ActionScript, PHP, Processing y otros. Otro factor importante es la publicación por parte de todas las grandes compañías Web 2.0 de sus API para mediados del 2000s. (API, o Interfaz de Programación de Aplicación en inglés, es un código que permite a otros programas de computadora acceder a servicios ofrecidos por una aplicación. Por ejemplo, la gente puede usar la API de Google Maps para insertar mapas Google completos en sus propios sitios web). Estos lenguajes de programación y *scripting* no hicieron a la programación en sí necesariamente más fácil. Más bien, la hicieron mucho más eficiente. Por ejemplo, cuando un joven diseñador puede crear un diseño interesante con solo un par de docenas de código escrito en Processing versus escribir un programa de Java realmente largo, él/ella es mucho más probable que se ponga a programar. Similarmente, si sólo unas pocas líneas de JavaScript te permiten integrar toda la funcionalidad ofrecida por Google Maps en tu sitio, esto es una gran motivación para ponerse a trabajar con JavaScript.

En un artículo del 2006 que evaluaba otros ejemplos de nuevas tecnologías que permiten a personas con poca o ninguna experiencia de programación crear nuevo software a medida (tales como Ning y Coghead), Martin LaMonica escribió sobre una posibilidad futura de “una *long tail* para apps”. Claramente, hoy las tecnologías de consumidor para capturar y editar medios son mucho más fáciles de usar que incluso los lenguajes de más alto nivel de programación y *scripting*. Pero no va a permanecer así necesariamente. Piensen, por ejemplo, en lo que hacía falta para armar un estudio de fotos y tomar fotografías en los 1850s frente a simplemente apretar un botón en una cámara digital o un teléfono móvil en los 2000s. Claramente, estamos muy lejos de tal simplicidad en programación. Pero no veo una razón lógica por la cual la programación no pueda un día volverse así de fácil.

Por ahora, el número de personas que pueden hacer el *script* y programar sigue creciendo. Aunque estamos lejos de una verdadera “Larga cola” para software, el desarrollo del software

está gradualmente volviéndose más democratizado. Es, por lo tanto, el momento adecuado para empezar a pensar teóricamente sobre cómo el software está dando forma a nuestra cultura y como es a su vez formado por la cultura. El momento de los “estudios de software” llegó.

Software Cultural

El teórico de medios y literatura alemán Friedrich Kittler escribió que los estudiantes de hoy deberían al menos conocer dos lenguajes de programación; “sólo entonces serán capaces de decir algo sobre qué es la “cultura” actualmente”. Kittler mismo programa en un lenguaje ensamblador (del inglés: *assembler*) - lo que probablemente determinó su desconfianza de las interfaces gráficas de usuarios y las aplicaciones de software modernas que utilizan estas interfaces. En una movida clásicamente modernista, Kittler argumentó que necesitamos enfocarnos en la “esencia” de la computadora - que para Kittler significaba los cimientos lógicos y matemáticos de la computadora moderna y su historia temprana caracterizada por herramientas tales como lenguajes assembler.

Este libro está determinado por mi propia historia de involucramiento con computadoras como programador, animador y diseñador informático, artista mediático y maestro. Este involucramiento práctico comenzó a principios de los 1980s, que fue la década de la programación procedimental (Pascal), más que de la programación de ensamblaje. También fue la década que vio la introducción de las PCs y el primer gran impacto cultural cuando la publicación de escritorio se hizo popular y el hipertexto empezó a ser discutido por algunos eruditos literarios. De hecho, llegué a Nueva York desde Moscú en 1981, que fue el año en que IBM presentó su primera PC. Mi primera experiencia con gráficos de computadora fue en 1983-1984 en la Apple IIE. En 1984 vi la Interfaz de Usuario Gráfica en su primera implementación comercial exitosa en la Apple Macintosh. El mismo año conseguí un trabajo en una de las primeras compañías de animación por computadora (Digital Effects) donde aprendí cómo programar modelos informáticos en 3D y animaciones. En 1986 estaba escribiendo programas de computadora que automáticamente procesaban fotografías para hacerlas parecer pinturas. en Enero de 1987 Adobe Systems publicó Illustrator, seguido de Photoshop en 1989. El mismo año vió el estreno de *El Abismo* dirigida por James Cameron. Esta película utilizó efectos pioneros CGI para crear el primer personaje virtual complejo. Y para Navidad de 1990s, Tim Berners-Lee ya había creado todos los componentes de la World Wide Web tal como existe hoy: un servidor web, páginas web y navegador web.

En resumen, en una década la computadora pasó de ser una tecnología culturalmente invisible a ser el nuevo motor de la cultura. Mientras que el progreso en hardware y la Ley de Moore por supuesto jugaron roles claves en esto, incluso más crucial fue la publicación de software dirigido a usuarios no-técnicos: nuevas interfaces de usuario gráficas, procesamiento de palabras, dibujo, pintura, modelado en 3D, animación, música, composición y edición, manejo de información, creación de hipermedios y multimedios (Hypercard, Director), y ambientes de información en red (World Wide Web). Con software fácil de usar disponible, la escena estaba preparada para la siguiente década de 1990 cuando la mayoría de las industrias culturales gradualmente se movieron a ambientes de software: Diseño gráfico, arquitectura, diseño de productos, diseño de espacios, filmación, animación, diseño de medios, música, enseñanza superior y administración cultural.

Aunque aprendí a programar por primera vez en 1975 cuando estaba en la escuela secundaria en Moscú, mi perspectiva sobre los estudios de software fue formada al observar como, comenzando a mediados de los 1980s, el software basado en GUI¹ rápidamente pusieron a las computadoras en el centro de la cultura. . Teóricamente, creo que deberíamos pensar sobre el tema del software de la manera mas amplia posible. Es decir, necesitamos considerar no sólo el software “visible” utilizado por los consumidores sino además el software “gris” que maneja todos los sistemas y procesos en la sociedad contemporánea. Sin embargo, ya que no tengo experiencia escribiendo software logístico o software de automatización industrial no voy a escribir sobre estos tópicos. Mi preocupación es con un subconjunto de software al cual utilice y enseñé en mi vida profesional y al cual llamo *software cultural*. Mientras que este término ya ha sido usado metafóricamente (ver J.M. Balkin, *Cultural Software: A theory of Ideology*, 2003), en este libro estoy buscando usar el término de manera literal para referirme a **programas de software que son utilizados para crear y acceder a objetos y ambientes de medios**. Los ejemplos son programas tales como Word, PowerPoint, Photoshop, Illustrator, Final Cut, After Effects, Flash, Firefox, Internet Explorer, etc. El software Cultural es, en otras palabras, es un **subconjunto de aplicaciones de software** que habilitan a la creación, publicación, acceso, la posibilidad de compartir y remixar imágenes, secuencias de imágenes en movimiento, diseños 3D, textos, mapas, elementos interactivos, así como también varias combinaciones de estos elementos tales como sitios web, diseños 2D, *motion graphics*, videojuegos, instalaciones interactivas comerciales y artísticas, etc. (Mientras que originalmente tal software de aplicación era diseñado para funcionar en máquinas de escritorio, hoy alguna de las herramientas de creación y edición de medios están también disponibles como webware, es decir, aplicaciones a las que se accede vía web tales como Google Docs).

¹ Nota del traductor: Interfaz gráfica de usuario

Dado que hoy la multimillonaria industria cultural global es habilitada por estos programas de software, es interesante que no hay una única manera aceptable de clasificarlos. Artículos de Wikipedia sobre “software de aplicación” incluyen las categorías de “software de desarrollo de medios” y “software de acceso a contenidos”. Esto es generalmente útil pero no completamente exacto - ya que hoy la mayoría del “software de acceso a contenidos” también incluye algunas funciones de edición de medios. QuickTime Player puede ser usado para cortar y pegar partes de un video; iPhoto permite una cantidad de operaciones de fotoedición, etc. Del mismo modo, en la mayoría de los casos el software de “desarrollo de medios” (o de “creación de contenido”) tal como Word o PowerPoint es el mismo software comúnmente usado tanto para desarrollar como para acceder a contenido (esta coexistencia de funciones de autoría y de acceso es en sí misma un rasgo distintivo del software cultural). Si visitamos los sitios web de los populares creadores de estas aplicaciones de software tales como Adobe y Autodesk encontraremos que estas compañías pueden dividir sus productos por mercado (web, transmisión, arquitectura, etc) o por subcategorías como “consumidor” y “pro”. Esto es lo máximo que comúnmente se encuentra - Otra razón por la cual debemos enfocar nuestras herramientas teóricas en interrogar al software cultural.

En este libro mi enfoque estará en estas aplicaciones para el desarrollo de medios (o “creación de contenidos”) - pero el software cultural también incluye otros tipos de programas y elementos IT. Una categoría importante son **las herramientas para la comunicación social y el compartir medios, información y conocimiento** tales como navegadores de web, clientes de email, clientes de mensajería instantánea, wikis, etiquetado social, herramientas de citado social, mundos virtuales y otros - en breve, el software social (nótese que tal uso del término “software social” en parte se superpone pero no es equivalente a la forma en que el término empezó a ser utilizado durante los 2000s para referirse a las plataformas Web 2.0 tales como wikipedia, Flickr, Youtube y demás). Otra categoría son **las herramientas para el manejo personal de la información** tales como libreta de direcciones, aplicaciones de administración de proyectos y motores de búsqueda de escritorio. (Estas categorías cambian con el tiempo: por ejemplo, durante los 2000s la frontera entre “información personal” e “información pública” comenzó a disolverse y desapareció en la medida que la gente comenzó rutinariamente a colocar sus medios en redes sociales y sus calendarios en línea. Similarmente, el motor de búsqueda de Google muestra los resultados tanto de tu máquina local como los de la web - así conceptual y prácticamente borrando la frontera entre el “yo” y el “mundo”). Ya que la creación de medios interactivos muchas veces involucra como mínimo algo de programación original y *scripting* además de lo que es posible dentro de las aplicaciones de desarrollo de medios tales como Dreamweaver o Flash, los **ambientes de programación** también pueden

ser considerados bajo el software cultural. Lo que es más, **las interfaces de medios** mismas - íconos, carpetas, sonidos, animaciones e interacciones de usuarios - son también software cultural, ya que estas interfaces median entre las interacciones de la gente con los medios y otras personas (aunque el viejo término de GUI, o interfaz gráfica de usuario, continúa siendo ampliamente utilizado, el término más reciente “interfaz de medios” es usualmente más apropiado ya que muchas interfaces hoy - incluyendo interfaces en Windows, Mac OS, consolas de juego, teléfonos móviles y pantallas interactivas de tiendas o museos tales como los proyectos Nanika de Nokia e instalaciones Diesel en el cetro Noble de la Paz en Oslo - utilizan todo tipo de medios además de los gráficos para comunicarse con el usuario). Me detendré aquí pero esta lista puede ampliarse fácilmente para incluir categorías adicionales de software también.

Cualquier definición es pasible de deleitar a algunas personas e irritar a otras. Por lo tanto antes de seguir adelante me gustaría salirle al encuentro a una posible objeción a la forma en que he definido “software cultural”. Por supuesto, el término “cultura” no es reducible a medios y “objetos” de diseño separados que pueden existir como archivos en una computadora y/o programas de software ejecutables y scripts. Incluye símbolos, significados, valores, lenguajes, ideologías, rituales, religión, vestimenta y códigos de comportamiento, y muchos otros elementos y dimensiones materiales e inmateriales. en consecuencia, antropólogos culturales, lingüistas, sociólogos y muchos humanistas pueden verse molestos por lo que puede parecer una reducción acrítica de todas estas dimensiones a un conjunto de herramientas creadoras de medios. ¿Estoy diciendo que la “cultura” de hoy es equivalente a un subconjunto particular de aplicaciones de software y que los objetos culturales pueden ser creados con su ayuda? Por supuesto que no. Sin embargo, lo que estoy diciendo - y lo que espero que este libro explicita en más detalle - es que a finales del siglo XX los humanos han agregado una dimensión fundamentalmente nueva a su cultura. Esta dimensión es el software en general, y las aplicaciones de software para la creación y el acceso a contenidos en particular.

Siento que la metáfora de una nueva dimensión agregada al espacio es muy apropiada aquí. Es decir, “software cultural” no es meramente un nuevo objeto - no importa cuán grande e importante - que ha sido soltado en el espacio que llamamos “cultura”. En otras palabras, sería impreciso pensar en el software como simplemente otro término para agregar al conjunto que incluye la música, el diseño visual, los espacios construidos, los códigos de vestimenta, lenguajes, comidas, culturas de clubs, normas corporativas y otros. Así que mientras que ciertamente podemos estudiar “la cultura del software” - observar cosas tales como las prácticas de programación, valores e ideologías de programadores y compañías de

software, las culturas de Silicon Valley y Bangalore, etc. - Si solo hacemos esto nos perderemos la importancia real del software. Como el alfabeto, las matemáticas, la imprenta, el motor de combustión interna, la electricidad y los circuitos integrados, el software reajusta y reconfigura todas las cosas a las que es aplicado - o al menos, tiene el potencial de hacerlo. En otras palabras, así como agregar una nueva dimensión del espacio agrega una nueva coordenada a todos los elementos en ese espacio, “agregar” el software a la cultura cambia la identidad de todas las cosas de las que una cultura está hecha.

En otras palabras, nuestra sociedad contemporánea puede ser caracterizada como una **sociedad de software** y nuestra cultura puede llamarse justificadamente una **cultura de software** - porque hoy el software juega un papel central en dar forma tanto a los elementos materiales como a muchos de las estructuras inmateriales que conjuntamente componen la “cultura”.

Como sólo un ejemplo de cómo el uso del software reconfigura incluso las prácticas sociales y culturales más básicas y nos hace repensar los conceptos y teorías que desarrollamos para describirlos, consideren al “átomo” de la creación cultural, transmisión y memoria: un “documento” (o una “obra”, es decir, un contenido almacenado en algún medio. En una cultura de software ya no lidiamos con “documentos”, “obras”, “mensajes” o “medios” en términos del siglo XX. En lugar de documentos fijos cuyo contenido y significado puede ser plenamente determinado al examinar su estructura (lo que la mayoría de las teorías de la cultura del siglo XX han hecho) ahora interactuamos con “performances de software” dinámicas. Uso la palabra “performances” porque lo que estamos experimentando es construido por el software en tiempo real. Así que sea que estemos navegando un sitio web, usando Gmail, jugando un videojuego, o usando un teléfono móvil con capacidad GPS para localizar lugares particulares o amigos cercanos, no estamos involucrados con documentos estáticos predefinidos sino con outputs dinámicos de computación en tiempo real. Los programas de computadora pueden usar una variedad de componentes para crear estos “outputs”: plantillas de diseño, archivos almacenados en una máquina local, medios extraídos de las bases de datos del servidor de la red, el input de un mouse, pantalla táctil u otro componente de interfaz, y otras fuentes. Así, aunque algunos documentos estáticos pueden estar involucrados, la experiencia de medios final construida por el software no puede ser reducida a un único documento almacenado en algún medio. En otras palabras, en contraste con las pinturas, obras literarias, piezas de música, películas o edificios, un crítico no puede simplemente consultar un único “archivo” conteniendo todo el contenido de la obra.

Leer el código - es decir, examinar el código escrito de un programa - tampoco nos ayudará. En primer lugar, si se tratara de un proyecto de comunicación interactiva de la vida real, el

código del programa sería tan largo y complejo que no permitiría una lectura significativa - además de que uno tendría que examinar todas las bibliotecas de código que éste puede utilizar; si se tratara de una aplicación web (o "webware") o un sitio web dinámico, estas utilizarían una arquitectura de software de varios niveles donde un número de módulos de software independientes interactuarían juntos (por ejemplo, un cliente web, servidor de aplicaciones, y una base de datos). En el caso del sitio web de dinámica comercial a gran escala, tales como amazon.com, lo que el usuario experimenta como una sola página web puede implicar interacciones entre más de sesenta procesos de software independientes.

En segundo lugar, incluso si un programa es relativamente corto y un crítico entiende exactamente lo que se supone que el programa hace examinando el código, esta comprensión de la estructura lógica del programa no puede ser traducida a imaginar la experiencia real del usuario. (Si se pudiera, el proceso de extensas pruebas con los usuarios reales que todo el software o compañía de medios hace antes de lanzar nuevos productos- o nuevas aplicaciones para un nuevo juego- no sería necesario). En resumen, estoy sugiriendo que los "estudios de software" no deben confundirse con "estudios del código". Y mientras que según otro enfoque -la comparación de código de computadora para una cuenta de la música que consigue interpretarse durante la ejecución (lo que sugiere que la teoría de la música se puede utilizar para entender la cultura de software)- parece más prometedora, también es muy limitada ya que no puede hacer frente a la más fundamental dimensión de software impulsada por experiencia multimedia y la interactividad.

Incluso en los casos aparentemente tan sencillos como ver un documento PDF o la apertura de una foto en un reproductor de medios, todavía estamos tratando con "acciones de software"- ya que es un software el que define las opciones para navegar, editar y compartir el documento, en lugar del propio documento. Por lo tanto, examinar el archivo PDF o un archivo JPEG de la manera los críticos del siglo XX examinaron una novela, una película o un programa de televisión, sólo nos dirá algunas cosas acerca de la experiencia de interacción con este documento a través de software. Mientras que el contenido del archivo, obviamente, forma una parte de esta experiencia, está también formada por la interfaz y las herramientas proporcionadas por el software. Por ello, el examen de las hipótesis, conceptos, y la historia de la cultura de software -incluyendo las teorías de sus diseñadores- es esencial si hemos de dar sentido a la "cultura contemporánea".

El cambio de lo que hoy constituye un "objeto" cultural, también trae preguntas incluso en las teorías más estables. Consideren lo que ha sido probablemente uno de los paradigmas más populares desde la década de 1950, la idea de la "transmisión" de la cultura desarrollada por

las ciencias de la comunicación. Este paradigma describe la comunicación de masas (y, a veces la cultura en general) como un proceso de comunicación entre los autores que crean "mensajes" y las audiencias las "reciben". Estos mensajes no son totalmente decodificado por el público, por razones técnicas (ruido en la transmisión) o por razones semánticas (no entendieron el significado previstos). La teoría de la comunicación y la industria de los medios de comunicación clásicos consideran a la recepción parcial un problema. Luego, a partir de la década de 1970 Stuart Hall, Dick Hebdige y otros críticos que más tarde fueron asociados con los estudios culturales, argumentaron que este tipo de fenómenos son positivos: las audiencias construyen sus propios significados a partir de la información que reciben. Pero en ambos casos teóricos implícitamente supone que el mensaje era algo completo y definido -con independencia de que se haya almacenado en algunos medios de comunicación o construida en "tiempo real" (como en los programas de televisión en directo). Por lo tanto, el miembro de la audiencia leería todos los textos publicitarios, vería una película entera, o escucharía la canción entera y sólo después los interpretaría, malinterpretaría, le asignaría sus propios significados, remezclaría, y así sucesivamente....

Si bien esta hipótesis ya han sido cuestionadas por la introducción de las tecnologías de TimeShift y DVR (grabadores de vídeo digital), esto no se aplica a los medios de software interactivos que ya han "nacidos digitales". Cuando un usuario interactúa con una aplicación de software que presenta un contenido cultural, este contenido a menudo no tiene límites finitos definidos. Por ejemplo, un usuario de Google Earth probablemente encuentre la información de manera diferente cada vez que se accede a la aplicación. Google podría haber actualizado algunas de las fotografías satelitales o añadido nuevas vistas de la Calle, o desarrollado nuevos modelos de edificios en 3D, nuevas capas y nueva información sobre las capas ya existentes podrían estar disponibles. Por otra parte, en cualquier momento el usuario puede cargar más datos geoespaciales creados por otros usuarios y empresas haciendo clic en "Añadir contenido" en el Panel Lugares o abrir directamente un archivo KLM. Google Earth es un ejemplo de un nuevo "documento" interactivo que no tiene todo su contenido predefinido. Su contenido cambia y crece con el tiempo.

Pero incluso en los casos de los documentos que corresponden a un solo archivo de la computadora, que están totalmente predefinidos y que no permiten ser modificados (por ejemplo, un archivo de sólo lectura en formato PDF), la experiencia del usuario está definida parcialmente por el archivo de contenido. El usuario es libre de navegar por el documento, elegir tanto la información que debe ver y la secuencia en la que ella está viendo. En otras palabras, el "mensaje" que "recibe" el usuario no es sólo es activamente "construido" por él

(a través de una interpretación cognitiva), sino también activamente "gestionado" (definiendo qué información recibir y cómo hacerlo).

Por qué la historia de los Estudios de Software no existe

“Всякое описание мира сильно отстает от его развития.”

(Traducción del ruso: “Cada descripción del mundo se queda muy por detrás de su desarrollo actual.”)

Тая Катюша, VJ en MTV

Vivimos en una cultura del software - es decir, una cultura de la producción, distribución y recepción de grandes cantidades de contenido - y cada vez más, la experiencia es mediada por el software. Sin embargo, los profesionales más creativos no saben nada sobre la historia intelectual de software que utilizan todos los días, ya sea Photoshop, GIMP, Final Cut, After Effects, Blender, Flash, Maya o MAX.

¿De dónde viene el software cultural contemporáneo? ¿Cómo llegaron sus metáforas y técnicas? Y ¿por qué se desarrolló? No lo sabemos. A pesar de las declaraciones comunes que la revolución digital es al menos tan importante como la invención de la imprenta, ignoramos cómo fue inventada la parte clave de esta revolución, es decir, el software cultural. Entonces se piensa en ello, es increíble. Todo el mundo en el mundo de la cultura sabe de Gutenberg (la imprenta), Brunelleschi (la perspectiva), los hermanos Lumiere, Griffith y Eisenstein (el cine), Le Corbusier (la arquitectura moderna), Isadora Duncan (la danza contemporánea), y Saul Bass (los motion graphics). (Bueno, si no sabés alguno de estos nombres, estoy seguro de que tenés otros amigos que sí lo hacen). Y, sin embargo, pocas personas se enteraron que JC Liicklider, Ivan Sutherland, Ted Nelson, Douglas Engelbart, Alan Kay, Nicholas Negroponte y sus colaboradores, aproximadamente entre 1960 y 1978, han creado gradualmente una computadora en la máquina cultural que representa hoy en día.

Notablemente, la historia del software cultural aún no existe. Lo que tenemos son unos cuantos libros en gran medida biográficos sobre algunas de las figuras individuales fundamentales y los laboratorios de investigación, tales como Xerox PARC o el Media Lab - pero no de la síntesis global que traza el árbol genealógico de software cultural. Y tampoco tenemos estudios detallados que relacionan la historia de programas culturales y la historia de los medios de comunicación, la teoría de los medios de comunicación, o de la historia de la cultura visual.

Instituciones del arte moderno (museos como el MOMA y la Tate, los editores de libros de arte como Phaidon y Rizzoli, etc) promueven la historia del arte moderno. Hollywood está

igualmente orgulloso de su historia (las estrellas, los directores, los directores de fotografía, y las películas clásicas). Entonces, ¿cómo podemos entender el olvido de la historia de la informática cultural por nuestras instituciones culturales y de la industria informática en sí? ¿Por qué, por ejemplo, Silicon Valley no tiene un museo de software cultural? (El museo de Historia de la Computación en Mountain View, California, cuenta con una amplia exposición permanente, que se centra en hardware, sistemas operativos y lenguajes de programación - pero no en la historia del software cultural).

Creo que la razón principal tiene que ver con la economía. Originalmente el mal entendido y ridiculizado arte moderno con el tiempo se ha convertido en una categoría de inversión legítima -de hecho, hacia mediados de la década del 2000, las pinturas de varios artistas del siglo XX se vendían incluso a mayor valor que las de los artistas clásicos más famosos. Del mismo modo, Hollywood sigue obteniendo beneficios de las viejas películas a medida que estas continúan siendo reeditadas en los nuevos formatos. ¿Qué pasa con la industria de tecnología de la información? No deriva de ningún beneficio del software antiguo - y por lo tanto, no hace nada para promover su historia. Por supuesto, las versiones actuales de Microsoft Word, Adobe Photoshop, AutoCAD de AutoDesk, y muchas otras aplicaciones culturales populares se basan en las primeras versiones que a menudo datan de la década de 1980, y las empresas siguen beneficiándose de las patentes que se declaran en nuevas tecnologías que utilizan estas versiones originales - pero, a diferencia de los videojuegos de la década de 1980, estas versiones de software tempranas no son tratados como productos separados, que vuelven a emitirse hoy. (En principio, puedo imaginar la industria de software como la creación de un mercado completamente nuevo para las viejas versiones de software o aplicaciones que en algún momento fueron muy importantes, pero ya no existen hoy en día - por ejemplo, Aldus PageMaker. De hecho, dado que la cultura de consumo explota de forma sistemática la nostalgia de los adultos hacia sus experiencias culturales de cuando eran adolescentes o jóvenes, es realmente sorprendente que las versiones de software más tempranas no se hayan convertido en un mercado todavía. Si usé MacWrite diario y MacPaint hacia mediados de la década de 1980, o Photoshop 1.0 y 2.0 entre 1990 y 1993, creo que estas experiencias formaban parte de mi "genealogía cultural" tanto como el cine y el arte que vi en el mismo tiempo. Si bien no creo necesario defender la creación de una nueva categoría de productos comerciales, si el software inicial fue ampliamente disponible en la simulación, este podrá catalizar el interés cultural de software de manera similar a la forma en que amplía la disponibilidad de los primeros juegos de computadora alimenta el campo de los estudios de videojuegos.

Como la mayoría de los teóricos hasta el momento no ha considerado el software cultural como un tema propio, distinto de "nuevos medios" "media art", "internet", "ciberespacio", "cibercultura" y "código", no falta sólo una historia conceptual de software de edición de medios de comunicación sino también a las investigaciones sistemáticas de sus funciones en la producción cultural. Por ejemplo, ¿cómo el uso de la aplicación de animación y composición populares After Effects ha reconfigurado el lenguaje del movimiento imágenes? ¿Cómo fue la adopción de Alias, Maya y otros paquetes 3D por estudiantes de arquitectura y arquitectos jóvenes en la década de 1990, y cómo ha influido en el lenguaje de la arquitectura? ¿y la co-evolución de las herramientas de diseño web y la estética de sitios web - desde el código HTML escueto en 1994 hasta los sitios de gran riqueza visual realizados con Flash, cinco años más tarde? Se encontrarán con frecuentes menciones y corta discusiones de estas y preguntas similares en artículos y conferencias, pero hasta donde yo sé, no ha habido ningún estudio de extensión sobre cualquiera de estos temas. A menudo, los libros de arquitectura, movimiento gráficos, diseño gráfico y otros ámbitos del diseño discuten brevemente la importancia de las herramientas de software por facilitar nuevas posibilidades y oportunidades, pero estas discusiones por lo general no se desarrollan más.

Resumen del argumento y los capítulos del libro

Entre principios de 1990 y mediados de la década del 2000, el software cultural ha sustituido a la mayoría de los medios tecnológicos que surgieron en el siglo XIX y XX. La mayor parte de la cultura de hoy se crea y se accede a través de software cultural -y, sin embargo, sorprendentemente, pocas personas saben acerca de su historia. ¿Cuál era el pensamiento y las motivaciones de las personas que entre 1960 y finales de 1970 crearon los conceptos y las técnicas prácticas que subyacen al software cultural de hoy? ¿Cómo afecta el cambio a métodos de producción basados en software que en la década de 1990 cambió nuestro concepto de "medios de comunicación"? ¿Cómo funcionan las interfaces y las herramientas que han reformado y siguen dando forma a las estéticas y lenguajes visuales que vemos empleados en el diseño y los medios de comunicación contemporáneos? Por último, ¿cómo una nueva categoría de software cultural que surgió en la década del 2000 -"software social" (o "social media") redefine el funcionamiento de los medios de comunicación y su identidad una vez más? Estas son las preguntas que he tomado en este libro.

Mi objetivo no es proporcionar una historia completa del software cultural en general, o la de la creación de medios en particular. Tampoco me propongo discutir las nuevas técnicas creativas que aparecen a través de diferentes ámbitos culturales. En cambio, voy a trazar un

camino en particular a través de esta historia que nos llevará desde 1960 hasta el día de hoy, y que pasará a través de algunos de sus puntos cruciales.

Mientras que los nuevos teóricos de la comunicación han dedicado considerables esfuerzos en tratar de entender las relaciones entre los medios digitales y los medios físicos y electrónicos viejos, las fuentes importantes (la redacción de los proyectos de Ivan Sutherland , Douglas Englehardt , Ted Nelson , Alan Kay y otros pioneros de la cultura software en la década de 1960 y 1970) siguen quedando en gran parte sin examinar. ¿Cuáles fueron sus razones para inventar los conceptos y técnicas que hoy en día hacen posible que las computadoras representen, o "medien" otros medios de comunicación? ¿Por qué estas personas y sus colegas han trabajado a su vez de manera sistemática en una computadora y en una máquina para la creación de medios de comunicación y su manipulación? Estas son las preguntas que he tomado en la parte 1, donde los analizo, centrándome en las ideas y el trabajo del protagonista fundamental del "movimiento por el software cultural": Alan Kay .

Sugiero que Kay y otros lograron crear un tipo particular de nuevos medios, y no sólo simular la apariencia de los antiguos. Estos nuevos medios utilizan formatos de representación existentes como componentes básicos, pero al mismo tiempo, se le añadieron muchas nuevas propiedades anteriormente inexistentes. Según lo previsto por Kay, estos medios de comunicación son expandibles, es decir, los propios usuarios deben ser capaces de agregar fácilmente nuevas propiedades, así como inventar nuevos medios.

En consecuencia, Kay llama a las computadoras el primer "metamedio" cuyo contenido es "una amplia gama de medios de comunicación ya existentes y que aún no fueron inventados".

Se establecieron las bases necesarias para la existencia de este metamedio entre 1960 y finales de 1980. Durante este período, los medios físicos y electrónicos disponibles anteriormente eran sistemáticamente simulados por el software, y así se inventó una serie de nuevos medios de comunicación. Este desarrollo nos lleva desde el primer programa de diseño interactivo (Dibujos de Ivan Sutherland en 1962) a las aplicaciones de escritorio comerciales que se hicieron la creación basada en software y el diseño muy accesible para los miembros de diferentes profesiones creativas y, finalmente, los consumidores de los medios de comunicación, como Word (1984), PageMaker (1985), Illustrator (1987), Photoshop (1989), After Effects (1993), entre otros.

Entonces, ¿qué pasa después? Las formulaciones teóricas de Kay, tal como las articuló en 1977, ¿pueden predecir con precisión la evolución de los próximos treinta años, o se han

producido nuevos acontecimientos que su concepto de "metamedio" no tuvo en cuenta? Hoy utilizamos los diversos medios ya existentes que ahora el software simula, así como nuevos medios de comunicación que antes no existían. Ambos se ha ampliado continuamente con nuevas propiedades. ¿Estos procesos de invención y de amplificación se llevan a cabo de forma aleatoria, o siguen caminos particulares? En otras palabras, ¿cuáles son los principales mecanismos responsables de la extensión de la computadora como "metamedio"?

En la parte 2 abordo a la siguiente etapa en el desarrollo de software de creación de medios de comunicación que históricamente se centra en la década de 1990. Aunque no discuto los diferentes mecanismos responsables para el desarrollo continuo y la expansión de metamedio, si analizo algunos de ellos en detalle. ¿Cuáles son? En una primera aproximación, podemos considerar estos mecanismos como formas de remezcla . Esto no debería ser sorprendente. En la década de 1990, el *remix* ha surgido gradualmente a medida hasta convertirse en la estética dominante de la era de la globalización, que afecta y reformula la configuración de todo, desde la música y el cine hasta la alimentación y la moda. (Si Fredric Jameson se refirió una vez al posmodernismo como "la lógica cultural del capitalismo tardío", tal vez podamos llamar al *remix* la lógica cultural del capitalismo global). Dada la dominación cultural de la remezcla, también podemos esperar encontrar lógicas remezcla en software cultural. Pero si afirmamos esto, entonces no hemos terminado. Todavía hay mucho trabajo por hacer. Ya que no tenemos teorías detalladas de la cultura del *remix* (con la posible excepción de la historia y usos de remezcla en música), al llamar a algo un "remix" esto requiere al mismo tiempo el desarrollo de su teoría. En otras palabras, si nos limitamos nombrar un fenómeno cultural "remezcla", esto no es en sí misma una explicación. ¿Cuáles son las operaciones de remezclas que están trabajando en el software cultural? ¿Son diferentes de las operaciones de remezcla en otras áreas culturales?

Mis argumentos que se desarrollan en la segunda parte del libro se pueden resumir de la siguiente manera: en el proceso de la traducción de las tecnologías de los medios físicos y electrónicos de software, todas las técnicas y herramientas que antes eran exclusivos de los diferentes medios de comunicación "se reunieron" en el mismo entorno de software. Esta unificación tuvo consecuencias fundamentales para el desarrollo cultural de la humanidad y por la evolución de los medios. Se rompe y transforma todo el panorama de las tecnologías de los medios de comunicación, de los profesionales creativos que las utilizan, y el propio concepto de "medios de comunicación" también lo hace.

Para describir cómo los medios de comunicación anteriormente separados ahora pueden trabajar juntos en un entorno basado en un software común, he acuñado el nuevo término de

“remix profundo”² Aunque el “remix profundo” tiene una conexión con lo que comúnmente se entiende por *remix*, además tiene sus propios mecanismos. Los entornos de producción de software permite a los diseñadores *remixar* no sólo el contenido de los diferentes medios de comunicación, sino también sus técnicas fundamentales: los métodos de trabajo y las formas de representación y expresión.

Una vez que se han simulado en una computadora las técnicas anteriormente no compatibles de diferentes medios de comunicación, comienzan a combinarse en nuevas formas infinitas, lo que lleva a nuevos híbridos, o, para usar una metáfora biológica, una nueva “especie de medios de comunicación”. A modo de ejemplo, entre muchos otros que se nos pueden ocurrir, piensen por ejemplo en la popular aplicación Google Earth, que combina las técnicas de la cartografía tradicional, el campo de los Sistemas de Información Geográfica (SIG), la infografía y animación 3D, el software social, la búsqueda y otros elementos y funciones. En mi opinión, esta capacidad de combinar técnicas de los medios de comunicación que antes estaban separadas representa fundamentalmente una nueva etapa en la historia de los medios, la semiosis humana, y la comunicación humana, habilitados por su “*softwarización*”.

Si bien hoy el “remix profundo” se puede encontrar en el trabajo y en todos los ámbitos de la cultura donde se utiliza software, me centraré en áreas concretas para demostrar cómo funciona en detalle. La primera área serán los *motion graphics*, una parte dinámica de la cultura contemporánea, que en la medida que yo sé, aún no ha sido teóricamente analizado en detalle en la web. A pesar de los precedentes seleccionados para “gráficos en movimiento” contemporáneos ya se pueden encontrar en los años 1950 y 1960 en las obras de Saul Bass y Pablo Ferro; su crecimiento exponencial desde mediados del decenio de 1990 está directamente relacionado con la adopción de software para diseño de imagen en movimiento (en concreto, After Effects, software lanzado por Adobe en 1993). El “*remix profundo*” es fundamental para la estética de los *motion graphics*. Es decir, la mayor proporción de los proyectos con *motion graphics* ejecutados hoy en todo el mundo derivan los efectos estéticos de la combinación de diferentes técnicas y tradiciones de los anteriores medios: animación, dibujo, fotografía, tipografía, gráficos 3D, video, etc, pero utilizadas de formas nuevas. Como parte de mi análisis, veo cómo la típica producción basada en software se convierte en un estudio de diseño contemporáneo: la forma en que un proyecto se mueve de una aplicación de software a otro da forma a la estética de los *motion graphics* y el diseño visual en general.

² Nota del traductor: el término utilizado por el autor es “deep remixability”, que podría traducirse como “remezclabilidad profunda”, pero ya ha sido traducida por otros traductores como “remix profundo”

¿Por qué elijo "*motion graphics*" como objeto de estudio central, en lugar de cualquier otra área de la cultura contemporánea que haya sido igualmente afectada por el cambio de los procesos de producción basados en software o bien que haya nacido con las computadoras? Los ejemplos de los primeros, son los que a veces se denominan "productos u áreas digitalizados" como la arquitectura, el diseño gráfico, diseño de producto, diseño de la información y la música; y los ejemplos de los segundos (referidos como los "productos o áreas que nacieron digitales") son el diseño de juegos, diseño de interacción, diseño de experiencia de usuario, el usuario diseño de interfaces, diseño de páginas web y la visualización interactiva de información. Ciertamente, la mayoría de las nuevas áreas de diseño que tienen una palabra "interacción" o "información" como parte de sus títulos y que se desarrolla desde mediados de la década de 1990 han sido tan ignorado por los críticos culturales y por lo tanto exigen tanta atención.

La razón tiene que ver con la riqueza de nuevas formas - visual, espacial y temporal - que se desarrolló en el campo de los *motion graphics* desde que comenzaron a crecer rápidamente tras la introducción de After Effects (1993). Si nos acercamos a *motion graphics* en términos de estas formas y técnicas (y no sólo su contenido), nos daremos cuenta de que representan un punto de inflexión importante en la historia de las técnicas de comunicación humanas. Mapas, pictogramas, jeroglíficos, ideogramas, varios guiones, alfabeto, gráficos, sistemas de proyección, gráficos de información, fotografía, el lenguaje moderno de las formas abstractas (desarrollado por primera vez en la pintura europea y desde 1920 adoptada en el diseño gráfico, diseño de producto y arquitectura), las técnicas de la cinematografía del siglo XX, la infografía 3D, y por supuesto, la variedad de los "nacidos digitales" efectos visuales - prácticamente todas las técnicas de comunicación desarrolladas por los seres humanos hasta ahora rutinariamente se consiguen al combinar proyectos de *motion graphics*. Aunque es posible que todavía tengan que encontrar la manera de utilizar plenamente este nuevo metalenguaje semiótico, la importancia de su aparición es difícil de sobreestimar.

Continúo la discusión de "remix profundo" a través de otra área de diseño de medios: los efectos visuales en películas. Películas como la serie Matrix de los hermanos Larry y Andy Wachowski (1999-2003), Sin City de Robert Rodríguez (2005) y 300 de Zack Snyder (2007) forman parte de una tendencia creciente de filmar gran parte o toda la película con un "digital backlot " o pantalla verde.

Estas películas combinan varias técnicas de comunicación para crear varias estilizadas estéticas que no se pueden reducir a la mirada del cine de acción en vivo del siglo XX o la animación por computadora en 3D. Como objeto de estudio, se analizan en detalle los

métodos de producción llamados *Total Capture* y *Virtual Cinematography*. Estos fueron desarrollados originalmente para las películas de Matrix y desde entonces se han utilizado en otras películas y videojuegos como EA SPORTS Tiger Woods 2007. Estos métodos combinan varias técnicas de los medios de comunicación de una manera particularmente compleja, lo que nos ofrece uno de los ejemplos más extremos de "remix profundo".

Si el desarrollo de software de medios de creación en la década de 1990 ha transformado los medios más profesionales y los campos de diseño, los desarrollos de la década de 2000 (como el paso de las aplicaciones de escritorio a las *webware*, aplicaciones que se ejecutan en la web, redes sociales, blogs fáciles de usar y herramientas de edición multimedial como Blogger, iPhoto e iMovie, junto con el continuo aumento de velocidad de los procesadores, la disminución del costo de las notebooks, netbooks y de almacenamiento, y la ampliación de las capacidades de los teléfonos móviles) han transformado la forma en que la gente común utiliza los medios. La explosión exponencial de la cantidad de personas que están creando y compartiendo contenidos de los medios, los números alucinantes de fotos y videos que suben, la facilidad con que las fotos y los vídeos se comparten entre personas, dispositivos, sitios web y blogs, la más amplia disponibilidad de redes más rápidas: todos estos factores contribuyen a una totalmente nueva "ecología de los medios". Estoy pensando, por ejemplo, en los estudios detallados de la economía sobre el fenómeno de la "larga cola", discusiones de la cultura *fans*, la producción social en red y la colaboración, o de la investigación dentro de un nuevo paradigma de la "ciencia de la web": estas dimensiones teóricas y estéticas de los medios de comunicación no han sido aún discutidas al momento en que yo escribo.

En consecuencia, la parte 3 se centra en la nueva etapa en la historia del software cultural, cambiando el enfoque de los medios de comunicación de autoría profesional a la web social y los medios de consumo. Las nuevas categorías de software incluyen sitios web de redes sociales (MySpace, Facebook, etc.), los sitios web de intercambio multimedial (Flickr, Photobucket, YouTube, Vimeo, etc.), software al nivel de los consumidores edición multimedial (por ejemplo, iPhoto), editores de blogs (Blogger, Wordpres), lectores de RSS y páginas de inicio personalizadas (Google Reader, iGoogle, Netvibes, etc). (Tengan en cuenta que estas plataformas -webwares especialmente diseñadas para los consumidores- evolucionan continuamente, por lo que algunas de las categorías anteriores, su popularidad, y la identidad de determinadas aplicaciones y sitios web puede cambiar en el momento en que están leyendo esto: un ejemplo gráfico es el cambio de la identidad de Facebook. Suring 2007 pasó de ser una nueva aplicación de medios sociales que competía con MySpace para convertirse en "OS sociales" con el objetivo de combinar la funcionalidad de diferentes

aplicaciones en un solo lugar - sustituyendo , por ejemplo, software de correo electrónico independiente a un software que puede ser utilizado por muchos.

Esta parte del libro también ofrece un punto de vista adicional sobre la forma de estudiar el software cultural en la sociedad. Ninguno de los programas de software y sitios web mencionados en el párrafo anterior funcionan de manera aislada. Por el contrario, participan en una ecología más grande, que incluye los motores de búsqueda, feeds RSS y otras tecnologías web: dispositivos electrónicos baratos de captura y medios de comunicación accesibles (cámaras digitales, teléfonos móviles, reproductores de música, reproductores de vídeo, marcos de fotos digitales), y tecnologías que permiten la transferencia de los contenidos entre los dispositivos, las personas y la web (dispositivos de almacenamiento, tecnologías inalámbricas como Wi-Fi y WiMax, estándares de comunicación como Firewire, USB y 3G). Sin esta ecología el software social no sería posible. Por lo tanto, toda esta ecología debe tenerse en cuenta en cualquier discusión sobre software social, así como el contenido y el acceso al nivel de consumidor, el desarrollo de software multimedia diseñado para trabajar con los medios de comunicación y los sitios de intercambio basadas en la web.

Y mientras que los elementos particulares y sus relaciones en esta ecología posiblemente cambien con el tiempo (por ejemplo, la mayoría del contenido de medios puede llegar a estar disponible en la red, la comunicación entre los dispositivos puede ser totalmente transparente, y la separación física entre las personas y los dispositivos que controlan, y los espacios "no inteligentes" pueden comenzar a borrarse) la idea de una ecología tecnológica, que consiste en muchas partes interactuantes que incluyen software, probablemente no vaya a desaparecer pronto. Un ejemplo de cómo la tercera parte de este libro comienza a utilizar esta nueva perspectiva es el análisis de "movilidad multimedial", es el caso de un nuevo concepto que puede permitirnos hablar de la nueva ecología tecnosocial en su conjunto, a diferencia de sus elementos por separado.