

# El código del diseño y el código de la programación: Pedagogías en tiempos de pandemia

*Por Mario Alberto Morales Domínguez*

---

CANTIDAD DE PALABRAS 6444

HORA DE ENTREGA

10-OCT.-2020 12:41P. M.

NÚMERO DE  
IDENTIFICACIÓN DEL  
TRABAJO

63942348

# **El código del diseño y el código de la programación: Pedagogías en tiempos de pandemia**

## **The Design Code and the Programming Code: Pedagogies in Pandemic Times**

Mario Alberto Morales Domínguez, Universidad Autónoma Metropolitana, Cuajimalpa, México

*Resumen: En el marco del confinamiento al que nos ha obligado la pandemia del COVID-19, este trabajo aborda una metodología pedagógica para enseñar fundamentos de programación a estudiantes de Diseño vía remota. El objetivo es mostrar cómo, a partir de una comprensión conjunta del diseño y la programación en relación con el contexto actual, es posible llevar a cabo no sólo una vinculación teórica sino también un acercamiento pedagógico con resultados que logran conectar con las emociones que experimentan los estudiantes al mismo tiempo que se les permite dar cuenta de su propio proceso de aprendizaje. Siguiendo la metodología del curso, primero haremos un rastreo de los orígenes de ambas disciplinas para mostrar que en su nacimiento tienen en común haber surgido a partir de la búsqueda de un código, en el primer caso un código semántico y en el segundo un código informático. Para el primero se retomarán los planteamientos del artista y teórico Wassilly Kandinsky. Para el segundo, retomaremos el lenguaje y entorno de programación Processing. A partir de esta vinculación, expondremos la metodología seguida durante 9 semanas, comenzando por un diario de emociones y siguiendo una planificación informática. Más adelante, se mostrarán algunos de los productos de este curso, llevado a cabo con 60 alumnos, quienes desarrollaron un programa informático cada uno y realizaron una documentación de su proceso. El resultado fue sorpresivo, ya que se obtuvo una serie ejercicios que resuelven el reto propuesto para el curso, pero además funcionan como material didáctico del diseño para los usuarios. En la conclusión haremos notar que la complementación del código informático con el código del diseño no sólo facilita el aprendizaje de ambos, sino que promueve en los alumnos un pensamiento del diseño complejo, basado en una comprensión proyectual y sistemática.*

*Palabras clave:* Kandinsky, informática, Processing.

*Abstract: In the context of the confinement caused by the COVID-19 pandemic, this paper addresses a pedagogical on-line methodology to teach programming fundamentals to design students. The aim is to show how, from a joint understanding of design and programming in relation to the current context, it is possible to carry out not only a theoretical link but also a pedagogical approach with results that manage to connect with the emotions that students experience while allowing them to realize their own learning process. Following the methodology of the course, we will first trace the origins of both disciplines to show that at their birth they have in common to have emerged from the search for a code, in the first case a semantic code and in the second a computer code. For the first one we will take up the approaches of the artist and theorist Wassilly Kandinsky. For the second, we will take up Processing programming language and environment. From this relation, we will expose the methodology followed during 9 weeks, starting with a diary of emotions and following a computer planning. Later, we will show some of the products of this course, carried out with 60 students, who developed a computer program each one and made a documentation of their process. The result was surprising, since a series of exercises were obtained that solve the challenge proposed for the course, but also work as didactic material of design for the users. In the conclusion we will note that the complementation of the computer code with the design code not only facilitates the learning of both, but promotes in students a complex design thinking, based on a systematic and projective understanding.*

*Keywords: Kandinsky, computer science, Processing.*

## **Introducción.**

Tanto el término “programación” como el término “diseño” tienen un punto en común. Ambos pueden ser identificados como sinónimos de “planeación” o “proyección”. No es casualidad tampoco que sea en la misma época, el siglo XIX en que comienzan a desarrollarse ambos. Se trata de la época de la industrialización, lo cual conlleva tanto la maquinización como la automatización del mundo. En este trabajo daremos cuenta de cómo esta comprensión conjunta de estas disciplinas, que viene desde la historia y que puede ser aplicada en un método de enseñanza, puede dar como resultado no solamente el aprendizaje de dichas materias, sino motivar a quienes aprenden a generar su propia propuesta didáctica. Vale la pena advertir que el método que aquí seguimos responde a las situaciones de los estudiantes, precisamente prestando atención en la forma en que históricamente toda creación humana está íntimamente ligada con el contexto en que emerge. Por esta razón, antes de pasar a la exposición del método daremos una introducción sobre las bases sobre las cuales fue construido. Posteriormente daremos cuenta de como fue aplicado a un curso de programación en línea y asincrónico con estudiantes de la UAM-Cuajimalpa, obteniendo resultados óptimos.

Como lo ha planteado Luis Rodríguez Morales (1995), en *El diseño antes de la Bauhaus*, fueron una serie de debates en el siglo XIX, los que llevaron a la consolidación del diseño como disciplina formal y con objetivos racionales. Entre ellos, podríamos nombrar el desarrollo del Sistema Americano de Producción en nuestro continente, pero también a la escuela de Artes y oficios en Inglaterra. De acuerdo con Will Gompertz (2012), fue Herman Muthesius a principios del siglo XX en Alemania quien dio un impulso a lo que más tarde se configuraría como la primera escuela formal del diseño, la Bauhaus. Siguiendo una investigación en Inglaterra que le llevaba a la figura de William Morris y la forma en que respondieron él y sus afines a las exigencias de la industria, sin dejar de lado la tradición artesanal. Así, se dio cuenta de que era esta forma de pensar toda la sociedad como un diseño lo que llevó a un gran auge en la producción inglesa. Con esta idea, se le encargó al diseñador Peter Behrens esta tarea de darle una estética a la producción industrial estatal de Alemania, haciéndose cargo en primer lugar de la Sociedad Alemana de Electricidad.

Por otro lado, dentro del mundo del arte el cubismo, como una de las primeras vanguardias, estaba buscando formas de representación con mucha influencia de otras culturas, invitándonos a mirar la realidad de otras maneras. Los futuristas, por su parte, se preguntaban cuál sería la forma de representar en el futuro. Así, por un lado el cubismo propone representar la realidad en todas sus dimensiones dentro del lienzo pictórico, descomponiendo los objetos en todos sus aspectos y elementos, más allá de un solo punto de fuga y una sola perspectiva. El cubismo representaba el

mismo objeto desde varias perspectivas al mismo tiempo; mientras que el futurismo propone representar también el tiempo. Estas nuevas formas de representación abrieron un campo de experimentación hacia otras formas más radicales. Una de estas formas fue la propuesta por Wassily Kandinsky. Este personaje se convertiría en una de las figuras más importantes para el diseño. Es además el más importante impulsor de lo que se conoce como expresionismo abstracto. Kandinsky utilizaba solamente figuras geométricas y colores no mezclados. Kandinsky sienta las bases del diseño moderno. Fue un maestro e ideólogo de la Bauhaus, la cual, a pesar de su corta duración, dio todas las bases para lo que hoy conocemos como diseño moderno.

Con la utilización de simplemente tres figuras, el triángulo, el cuadrado y el círculo, se desarrolló gran parte de la teoría del diseño. Estas son las tres figuras primarias a las cuales se les asignó un color. El triángulo, por sí mismo tendría que ser amarillo, el cuadrado rojo y el círculo azul (1989). Kandinsky llegó a estas conclusiones por medio de varios métodos. Llevó a cabo una serie de encuestas por ejemplo entre los alumnos de la Bauhaus, recolectando todos estos datos (Lupton y Miller, 1994). De lo que se trataba era de componer un lenguaje visual, un código que sirviera no sólo en las imágenes, sino en los objetos industriales, en la arquitectura, pero también para los espacios y la vida urbana. De la Bauhaus salieron muchos teóricos y diseñadores que llevaron esto a diferentes niveles. Por ejemplo el alfabeto de Herbert Bayer. Trataron de dominar el lenguaje de las formas básicas y la composición de mensajes a partir de las formas primarias para antes de pasar a otro tipo de composiciones.

Una vez que tenemos esta comprensión básica podemos comprender lo que se trataba de expresar en un código del diseño. A través del uso de colores básicos y formas geométricas intentaron expresar sentimientos, pensamientos, generar ambientes, ideas, etc. Lo que se propone es el diseño como el pensamiento de toda una sociedad a partir de un código básico. Un código del diseño como configuración de un mundo. En lo que sigue, hablaremos sobre el código informático, haciendo un paralelismo para su posible aplicación conjunta.

La historia de la programación comienza con Ada Lovelace, quien logra desarrollar el primer lenguaje de programación a partir de experimentos previos para dar instrucciones precisas a telares. Programar, en el sentido informático significa, dar una serie de instrucciones estructuradas y ordenadas –a partir de algoritmos, los cuales son operaciones, procesos o las reglas que debe seguir el programa– para que una máquina las ejecute. Estos programas no pueden pensar por sí mismos. Son los seres humanos los que dotan a los programas de todas y cada una de las instrucciones exactamente como se deben ejecutar. Esto se logra a partir de lenguajes de programas o modos de comunicarse con las computadoras. Estos modos son mucho más sencillos que los modos de comunicarse entre los seres humanos, pues se hace solamente a partir algunos poco elementos se conforma todo un lenguaje.

Para que se lograra desarrollar los lenguajes de programación que conocemos hoy, hubo todo un desarrollo previo. Se habla de al menos 3 o 4 generaciones de lenguajes de programación, desde aquellos que resultaban inaccesibles para la lectura humana conformándose únicamente por ceros y unos (primera generación), hasta los que lo hacían más accesible, primero con instrucciones atómicas (segunda generación) y luego con instrucciones muy similares al lenguaje humano, integrando funciones como +, -, \*, /, for, if, not, and, or, etc (tercera generación). Por último, una cuarta generación incluiría los lenguajes de 3a generación e integran paquetes de herramientas para usarlos. Aquí está *Processing*, que es el lenguaje en el que nos vamos concentrar en este trabajo.

*Processing* es un lenguaje de programación desarrollado desde el año 2001. Fue desarrollado especialmente para diseñadores y artistas visuales. A diferencia de otros lenguajes y entornos de programación, *Processing* privilegia la imagen y los resultados visuales (Breas y Fry, 2007). De esta manera, mientras que en los lenguajes de programación más comunes lo primero que se enseña regularmente es el modo de realizar operaciones matemáticas o lógicas, en *Processing* sus propios desarrolladores prefieren iniciar con la formación de figuras geométricas y el uso de colores. Lo que le permite a *Processing* realizar esto es que incluye ya una librería a partir de la cual se pueden generar imágenes, mientras que normalmente para otros lenguajes las librerías que permitían trabajar con imagen tenían que ser importadas. Del mismo modo, había pocos entornos de programación que tuvieran bien desarrollada la parte visual. Por ello, *Processing* ha logrado posicionarse dentro del campo del diseño y las artes visuales como el lenguaje ideal para este tipo de proyectos.

En *Processing* las primeras funciones responden a la generación de un punto, un triángulo, un cuadrado o un círculo. a partir de estas primeras funciones y las de color, en el código de colores luz, rojo, verde y azul, es posible llegar a componer prácticamente cualquier imagen. Esto es lo que hace que este lenguaje se preste perfectamente para enseñarse junto con el código del diseño del que hemos hablado. En este caso se van a generar programas visuales y, por lo tanto, vamos a utilizar el código del diseño y el código informático a la vez.

En la escuela de la Bauhaus, el diseño llega a la integración de un código. Varios de los profesores de la Bauhaus eran marxistas. Las ideas sociales de Marx tienen alcances en la educación en muchos sentidos. El psicólogo Lev Vygotski (1979) sería quizá el primero que llevó de una forma directa sus planteamientos, al dar cuenta de que todo aprendizaje y todo desarrollo del lenguaje y del conocimiento está íntimamente ligado a un contexto social. Con la programación nos enfrentamos a un tipo de lenguaje y un tipo de aprendizaje muy similar al del habla. Pero, de la misma forma en que lo demostró Vygotski, éste no puede ser abordado en aislado. Con el diseño, por su parte, pasa lo mismo. Las formas geométricas básicas, las repeticiones de módulos o texturas, ritmos o contrastes, no sirven de nada si no se sitúan en un problema de diseño. Jamás un diseñador aprende primero eso que son llamados los “fundamentos” del diseño y luego va y los pone en práctica con un problema.

Si así lo hiciera, su diseño no solamente sería algo ya muchas veces visto, sino que sería aburrido, no tanto por costumbrista sino por falto de espontaneidad. Hay algo en el diseño que supone un salto respecto a esa posibilidad de aprender un conocimiento y simplemente repetirlo a la hora de enfrentarse a un problema profesional.

Con base en lo anterior, partimos de la comprensión de que no se puede enseñar ninguna materia aislada de su contexto. En el momento presente, además, bajo las circunstancias a que nos ha llevado la pandemia derivada del COVID-19, se vuelva aún más imperativo prestar atención a la situación en que se encuentran los estudiantes que intentan acercarse tanto a la disciplinas aquí mencionadas, como a todas las otras. Por esta razón, en este trabajo llevamos todos estos fundamentos hacia la enseñanza de la programación estructurada para diseñadores en el marco del segundo trimestre de la licenciatura de Diseño de la Universidad Autónoma Metropolitana, Unidad Cuajimalpa. A partir de la comprensión de los principios del diseño, los principios de la programación estructurada y los de psicología del color y las formas, se les instruyó a los alumnos para que ofrecieran soluciones con una salida enfocada directamente en el uso de formas geométricas para la expresión formal y en relación con la comprensión de uno de los momentos de la historia del arte cruciales para la historia del diseño, a saber, las propuestas de Kandinsky.

Hoy, año 2020, nos enfrentamos a un momento importante en la historia de la humanidad. No sabemos cuáles van a ser las consecuencias y los cambios que podrán ser graduales o radicales en cuanto a contacto y relaciones interpersonales. Al igual que en la escuela de la Bauhaus, donde se unían saberes artísticos, técnicos e incluso espirituales, actualmente nos toca pensar de qué maneras el diseño puede responder a nuestras condiciones concretas, utilizando nuestros saberes. Retomando a Vygotski, recordemos que si no hay necesidad vital no hay aprendizaje. Con todo el contexto desplegado anteriormente, se les enseñó a los alumnos una introducción al código el diseño y al código informático al mismo tiempo. El objetivo, en este recuento de lo sucedido, es mostrar cómo, a partir de una comprensión conjunta del diseño y la programación en relación con el contexto actual, es posible llevar a cabo no sólo una vinculación teórica sino también un acercamiento pedagógico con resultados que logran conectar con las emociones que experimentan los estudiantes al mismo tiempo que se les permite dar cuenta de su propio proceso de aprendizaje. Se trata de una aportación en torno a metodologías novedosas que responden al desasosiego de estos tiempos y donde se vuelve necesario tanto regresar a los fundamentos como generar nuevos planteamientos acerca de todas las disciplinas. A partir de esta vinculación en relación con la comprensión del código en sentido amplio, expondremos la metodología seguida. Más adelante, se expondrán algunos de los productos de la puesta en marcha de este curso, llevado a cabo con 60 alumnos, los cuales no sólo desarrollaron un programa informático cada uno, sino que también llevaron a cabo una documentación de todo su proceso.

## **Método.**

Participantes: 80 alumnos inscritos en la materia de “Fundamentos de programación estructurada”. Estos alumnos pertenecían a tres grupos diferentes inscritos al trimestre 20-P de la licenciatura de Diseño de la UAM-Cuajimalpa. De estos alumnos inscritos, sólo 60 entregaron en trabajo en tiempo y forma. Es conveniente señalar que en el trimestre anterior, estos alumnos cursaron tres materias de tronco común, como parte de la División de Ciencias de la Comunicación y Diseño, junto a alumnos de las licenciaturas de Ciencias y la Comunicación, y Tecnologías y Sistemas de la Información; y solo una materia especializada en el ámbito del diseño, con el nombre de “Taller de Lenguajes para el Diseño”. Además, vale la pena tomar en cuenta que, en el mismo trimestre en que cursaron la materia que aquí nos aqueja, los alumnos tomaron las materias de “Introducción a la Historia del Arte Universal”, “Taller de Representación y Expresión por Medio del Dibujo y del Trazo Geométrico” y “Laboratorio del Diseño Bi y Tridimensional”. Estos aprendizajes que corrieron simultáneamente con el curso del que aquí damos cuenta sirvieron también como complementos.

Criterios de participación: Es necesario tomar en cuenta que la materia fue impartida en el periodo que abarcó del lunes 11 de mayo al 11 de julio, de decir, casi desde al comienzo de la declaración de la contingencia sanitaria por las autoridades mexicanas. Por este motivo, el trimestre se llevó a cabo de acuerdo con criterios de flexibilidad del Proyecto Emergente de Enseñanza Remota (PEER), del Consejo Académico de la UAM, en los cuales se estableció que en la evaluación del curso los alumnos no verán perjudicadas sus trayectorias académicas, de manera que las calificaciones no aprobatorias obtenidas no pasarán a formar parte de su historial académico. Es importante mencionar esto no solamente porque refuerza la idea que ya señalamos en la introducción de que los alumnos se encontraban en una situación tan atípica como desconcertante; también nos habla de que los alumnos que decidieron inscribirse lo hicieron tomando en cuenta que se trataba de una oportunidad experimental que no les afectaría a su historial, pero a su vez también nos dice mucho sobre el hecho de que quienes decidieron mantenerse en el curso y terminarlo no hubo ningún tipo de temor a sanción alguna. El criterio de participación, en ese sentido fue totalmente libre para los alumnos que se encontraban cursando la carrera de Diseño en ese nivel.

Materiales: Para que los alumnos pudieran inscribirse en la materia, se les pidieron los siguientes requisitos técnicos mínimos:

- a) Tener computadora en casa, pudiendo ser de escritorio o personal, siempre y cuando fuera posible hacer uso de ella por lo menos 6 horas a la semana distribuidas en la forma que gusten.
- b) Bajar el entorno de programación *Processing*, en la página <https://processing.org>. Para este requisito no importaba el sistema operativo de la computadora.
- c) Tener acceso a internet en la computadora por lo menos 2 de las 6 horas semanales indicadas.

Todo el material necesario para este curso se dispuso en la página del profesor:  
<https://mariomoralesph.wordpress.com>

Procedimiento: A lo largo de 9 semanas, el profesor colocaba el material necesario en diferentes entradas de blog, una por semana, para trabajarla en este periodo de tiempo. El material podía consistir en videos, bibliografía y enlaces correspondientes al tema. Los alumnos podían consultar este material de manera asincrónica, es decir en el día y horario que tuvieran ellos consideraran adecuado. A continuación haremos una descripción breve de los contenidos de cada semana.

Primera semana: Se les pidió a los alumnos pensar cómo crear una programa en el que se pudieran traducir emociones en colores y formas básicas a través de un programa de computadora. Para completar esta tarea, se les pidió seguir los siguientes pasos: 1) Escribir un diario durante 4 días en el cual expresaran las emociones que habían experimentado durante la cuarentena. 2) Pensar cómo generaría un programa para traducir eso que registraran en su diario en colores y formas, creando relaciones entre las emociones y el lenguaje de diseño. Lo importante de esta primera planeación de los programs era incentivar la creatividad, sin importar aún si su planeación era correcta o incorrectamente formulada.

Segunda semana: En la segunda semana se le pidió a los alumnos formalizar y sistematizar la idea de su programa, dándoles las herramientas necesarias para que configuraran su planeación en un diagrama de flujo y lo que en programación se llama pseudocódigo. Todo esto, con base en el libro *Fundamentos de programación*, de Luis Joyanes Aguilar (2008), facilitándoles este material.

Tercera semana: En esta semana se comenzó a utilizar el entorno de programación de *Processing*, empezando a bocetar directamente en código informático. Siguiendo una progresión basada en los planteamientos de Kandinsky, se comenzó con el punto, luego la línea y luego el plano en sus formas básicas, así como los colores básicos y sus significados, explicando cada uno de estos elementos en relación al diseño. Como tarea para esta semana se les pidió a los alumnos generar por lo menos tres

composiciones con colores y formas básicas que expresaran alguna emoción, con base en los fundamentos ya mencionados.

Cuarta semana: En esta semana los alumnos generaron ya una primera versión de su programa a través de elementos muy sencillos. Se les explicó cómo colocar texto en la pantalla y cómo lograr que el usuario interactúe con la computadora, dándole instrucciones para que lo haga a través del teclado, emulando un lenguaje de programación de bajo nivel o de segunda generación. Del mismo modo, se les explicó el uso de variables para generar animaciones sencillas para aplicarse en sus composiciones.

Quinta semana: En esta semana se les enseño a los alumnos las funciones de interacción con el mouse, de modo que a partir de esto puedan generar botones en la pantalla. De esta forma, las interacciones que la semana anterior habían sido programadas con el teclado, esta vez serían con el mouse y, por lo tanto, se hace más complejo su diseño. Como guía para estas funciones se usó el libro *Hola mundo con Processing* (Abascal, López y Zepeda, 2015), publicado por la UAM-Cuajimalpa.

Sexta semana: En esta semana se les enseñó a los alumnos a integrar imágenes externas, fuentes y sonidos a su programa, de tal forma que pudieran complementar su programa con este tipo de elementos buscando una forma final en cuanto a su presentación para el usuario. Con esto el programa de cada alumno quedó concluido.

Séptima semana: En esta semana se les dio a los alumnos las instrucciones para poder pasar su programa de la versión de *Processing* de escritorio a p5.js, que es la versión en línea del mismo entorno. En las semanas anteriores se había estado trabajando con la versión de escritorio debido a que, bajo el entendido de que no es posible contar con que los alumnos tengan las condiciones tecnológicas adecuadas para trabajar en línea todo el tiempo, el desarrollo de su programa se hubiera visto afectado.

Octava semana: Se comenzó a trabajar en un pad colaborativo donde cada alumno fue capaz de subir su programa a la vista de los demás. Se implementaron algunos fundamentos de HTLM para poder insertar su ventana del programa en una página de internet y también se les dieron algunas bases de diseño editorial para que su documentación pudiera ser también mostrada en versión PDF dentro de la página.

Novena semana: Se realizó una revisión de todo el procesos y se afinaron detalles para la presentación final.

## **Resultados.**

El resultado fue sorpresivo, ya que se obtuvo una serie ejercicios lúdicos que resuelven de forma intuitiva este reto propuesto para el curso, pero que además funcionan como material didáctico del diseño para los usuarios. Uno de los resultados fue una exposición virtual que tuvo como salida un pad colaborativo bajo el título “Procesando la emergencia”, donde todos y cada uno de los estudiantes tenía la responsabilidad de insertar su propio trabajo usando código html. La exposición partió de una selección de 10 trabajos principales como entrada hacia los resultados del curso, pero a partir de ello se invitaba al usuario a consultar todo el material desarrollado por el resto de los alumnos. Esta exposición puede encontrarse en el siguiente enlace:

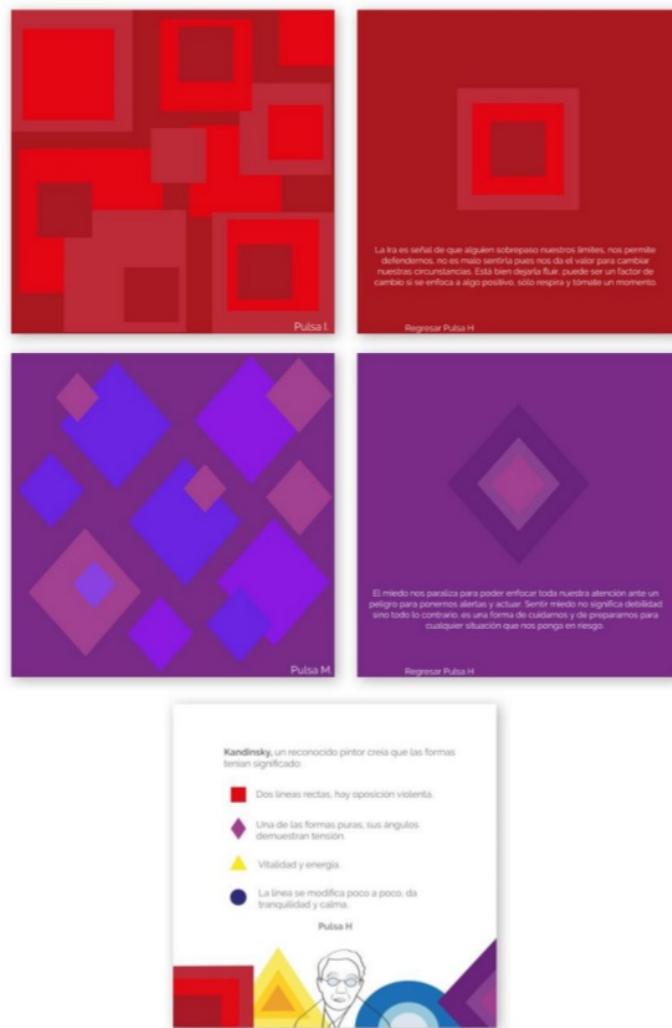
<https://pad.programando.li/s/ProcesandoLaEmergencia>

Parte importante del desarrollo de cada uno de los programas de los alumnos, y por lo cual quizás podríamos explicar estos resultados, es que ellos pudieron dar cuenta de su propio proceso a través de una documentación que fueron llevando a lo largo de todo el curso, la cual fue adjuntada en formato PDF a su entrega final de la materia, y que también acompaña a su presentación en la exposición virtual mencionada. Este requerimiento les introdujo también de manera sucinta a la edición y el diseño editorial en el sentido de que tuvieron que buscar cada uno la manera de mostrar su trabajo a un espectador que desconociera el proceso y el contexto desde el cual se realizó.

A continuación mostraremos sólo dos ejemplos de los trabajos desarrollados por los alumnos. Estos dos ejemplos muestran cómo los alumnos lograron no solamente expresarse a través del diseño y desarrollar un programa integrando composiciones propias, animaciones, loops, botones, elementos externos como pistas sonoras, etc., más importante aún resulta que aunque esto no estaba planteado directamente desde los objetivos del curso, a través de sus programas lograron una didáctica del diseño. Mostraremos también algunas páginas de los documentos en los cuales los alumnos describen su propio proceso de trabajo y llevan a cabo una reflexión sobre el mismo. Vale la pena decir que la selección de estos dos ejemplos no es aleatoria sino arbitraria. Se trata de los dos ejemplos que mejor demuestran la integración del código del diseño y el código de la informática, así como un desarrollo completo que termina sirviendo como material didáctico del diseño.

El primer caso corresponde al de la alumna Leslie Ximena Cruz Valencia, quien desarrolló un programa con el nombre IMAT. Experiencia emocional, elegido por ella misma siguiendo las iniciales de las emociones Ira, Miedo, Alegría y Tristeza; así como una descripción de lo que ella pretende lograr a través de su proyecto. En su programa identificó cada una de estas emociones con

un color y una figura geométrica distinta, siguiendo los planteamientos de Kandinsky. Además agregó una animación con movimientos diferenciados entre cada emoción/figura/color y música acorde con cada uno de estos elementos. En una segunda pantalla de su experiencia con cada elemento agregó la definición de la emoción y finalmente, como una pantalla común agregó una descripción básica de las figuras según Kandinsky (imágenes 1.1 y 1.2).



**Imagen 11.12.** Captura de pantalla de la interfaz del programa desarrollado por la alumna Leslie Ximena Cruz Valencia (primera parte), 2020. Imágenes por cortesía de la alumna.

En su documentación, Leslie Ximena dio cuenta de cada uno de sus procesos separados en apartados específicos para cada emoción. Del mismo modo, dio cuenta de sus decisiones en torno a los movimiento adecuados para cada animación y la música elegida. La documentación completa abarcó 52 páginas. A continuación mostramos algunas de estas páginas que la conforman, escogiendo solamente el diagrama de flujo y el proceso de una de las emociones, a saber, la alegría, identificada con el triángulo y el color amarillo (Imagen 2).



Imagen 2. Selección de páginas de la documentación del proceso de la alumna Leslie Ximena Cruz Valencia, 2020. Imágenes por cortesía de la alumna.

El segundo caso corresponde al programa desarrollado por el alumno Gilberto Camacho Escobar, quien decidió realizar su proyecto como una especie de recorrido audiovisual utilizando frases motivacionales, composiciones propias y obras de arte reconocidas en relación con una elección de colores a partir de una primera muestra. En su recorrido Gilberto da una serie de opciones entre cuadros de colores en una primera pantalla y a partir de esta elección, se identifica ésta con una emoción y va llevando al usuario por una serie de pantallas que terminan siempre en una obra de arte reinterpretada por Gilberto en formas geométricas y animaciones sencillas que le dan una especie de “brillo” a las imágenes, haciendo que algunos elementos aparezcan y desaparezcan, se muevan o cambien de tamaño o color. Al mismo tiempo, Gilberto escogió piezas musicales que acompañan a

las diferentes elecciones del usuario en relación con la emoción identificada. A continuación mostramos sólo algunas de las pantallas que componían en recorrido de Gilberto (imágenes 3.1 y 3.2).

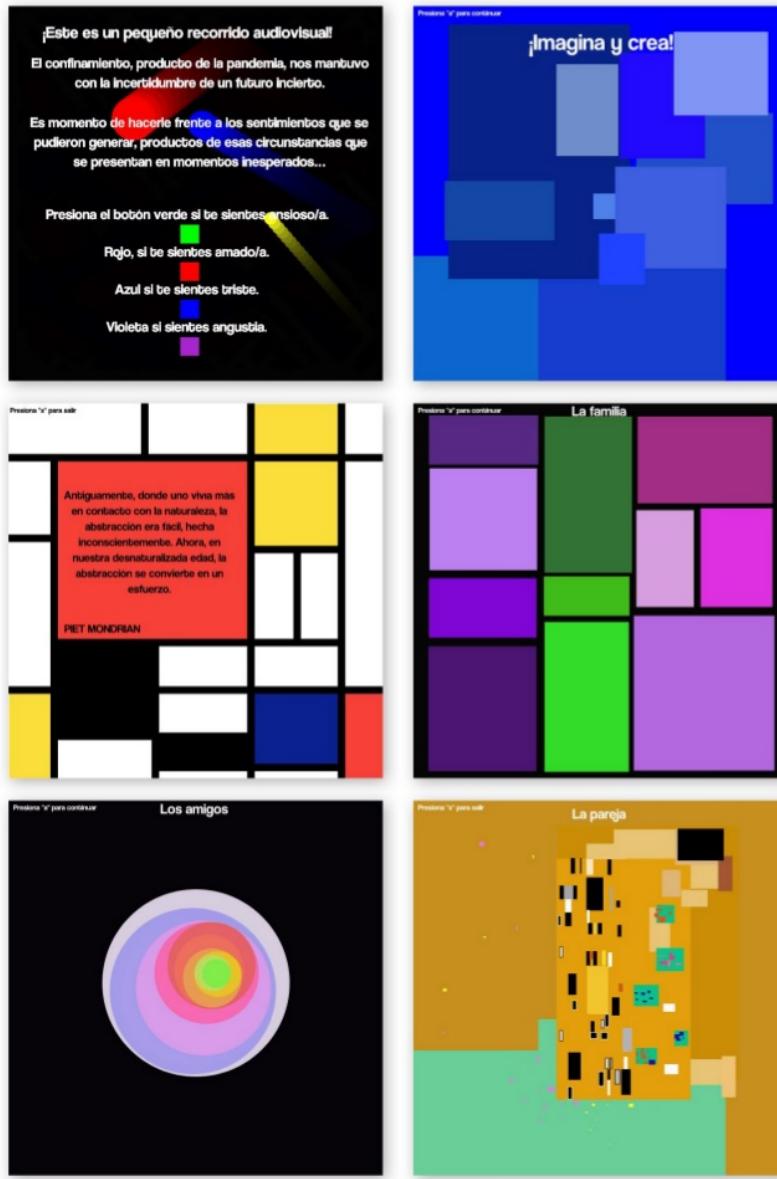


Imagen 3.1. Capturas de pantalla de la interfaz del programa desarrollado por el alumno Gilberto Camacho Escobar (primera parte), 2020. Imágenes por cortesía del alumno.

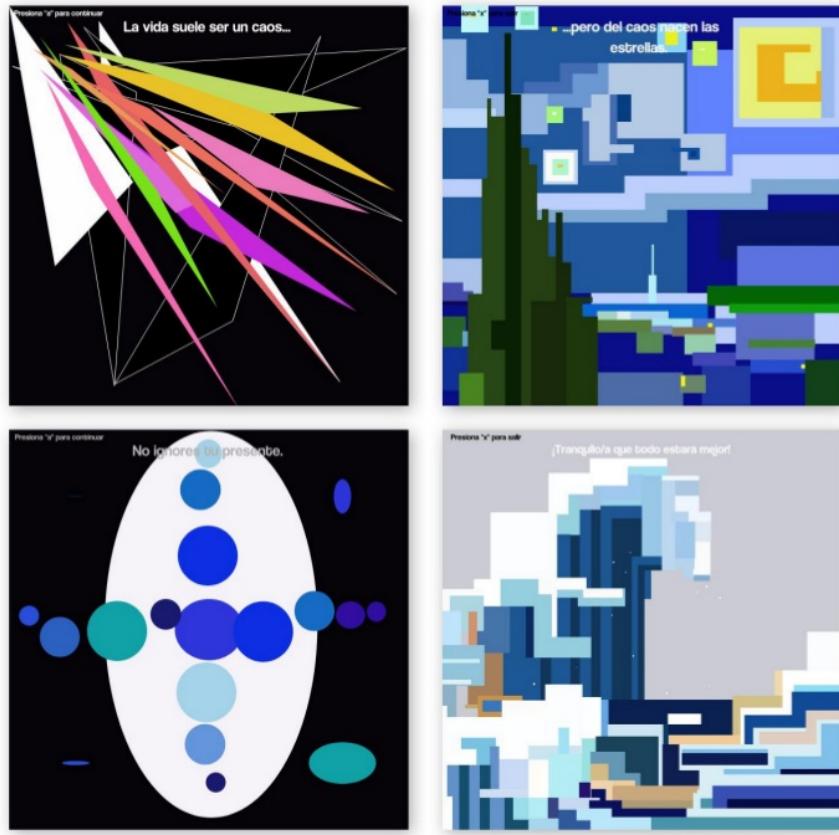


Imagen 3.2. Capturas de pantalla de la interfaz del programa desarrollado por el alumno Gilberto Camacho Escobar (segunda parte), 2020. Imágenes por cortesía del alumno.

En su documentación Gilberto dio cuenta del proceso de diseño que le llevó a una selección de pinturas que él estaba viendo en la materia de Introducción a la Historia del Arte Universal, puesto que encontró que estas materias le habían ayudado a sobrellevar el confinamiento relacionado con la contingencia sanitaria. Por este motivo, decidió asociar el contenido de esta materia relativa a las emociones y los colores con aquella otra donde se veían aplicados este tipo de conocimientos en obras específicas. Generó así un recorrido en el cual él comparte su experiencia en el confinamiento al mismo tiempo que involucra al usuario en ella. A continuación mostramos algunas de las 19 páginas que abarcó su documentación completa (Imagen 4).



Imagen 4. Selección de páginas de la documentación del proceso del alumno Gilberto Camacho Escobar, 2020. Imágenes por cortesía del alumno.

Como puede verse en estos resultados los estudiantes desarrollaron programas que no solamente integran el diseño y la programación de una manera conjunta. Las formas y los colores del código del diseño responden a cada uno de las funciones de visualización más sencillas de *Processing*, como son el triángulo, el cuadrado y el círculo, pero no lo hacen en ningún momento sin la intención del diseño. En los programas, además se asocia directamente el uso de estos códigos con su contexto, en el primer caso con las teorías de Kandinsky y en el segundo, más arriesgado, con otras obras de arte que el alumno decidió utilizar en relación con los conocimientos que estaba adquiriendo. Con esta conjunción, en la que los alumnos integraron las composiciones que realizaron desde las primeras semanas y fueron poco a poco desarrollando, animándolas y buscando la forma en que pudieran vincularlas a partir de instrucciones y botones adecuados para la interacción, no solamente

dimostraron resolver los requerimientos de la materia escolar que es exigía desarrollar un programa sencillo, sino que en el camino generaron un modo creativo de enseñanza del diseño en lo que respecta a sus vínculos con el estudio de la imagen y la historia del arte. Este resultado, como ya lo dijimos, fue en realidad inesperado. Rebasó las expectativas, las cuales se remitían únicamente a que los alumnos desarrollaran una programa donde el código del diseño sirviera para traducir emociones del usuario a través de un programa informático. Sin embargo, tal parece que la forma misma en que fue impartido el curso a partir de las asociaciones históricas de las que dimos cuenta en la introducción generó estos resultados.

### **Conclusión.**

Comenzamos este artículo dando cuenta de las vinculaciones históricas entre la programación y el diseño. Vimos que, de hecho, ambas esferas del saber podrían ser comprendidas no solamente de manera conjunta, sino como parte de un mismo impulso relacionado con el desarrollo industrial. El diseño, sin embargo, desde su nacimiento estuvo ligado con debates sociales que buscaban fortalecer el espíritu humano de una forma que no fuera solamente racional o maquínica. Y aún con todo este esfuerzo, finalmente la consolidación de la disciplina la llevó a reconciliarse con la industria y el pensamiento racionalista, de tal manera que con la escuela de la Bauhaus, el diseño buscó su propio código. Era un código visual que, no por sus pretensiones universalistas dejó de lado jamás su relación con el mundo del arte. Al contrario, se nutrió de ello y en sentido inverso, también le hizo muchas aportaciones. Con este breve recorrido dimos cuenta de un tipo particular de creatividad en el diseño, la cual se engarza siempre e inevitablemente con el contexto en que emerge.

Por su parte, el desarrollo de la informática tuvo su propia velocidad y particularidades. Dimos cuenta del esfuerzo realizado para que la comunicación entre el lenguaje de la computación y el lenguaje humano se diera de la forma más fluida posible, utilizando métodos de conversión de uno a otro para hacer la programación algo cada vez más accesible. Pero también señalamos que no fue sino hasta el año 2000 en que se desarrolló *Processing* como un lenguaje de programación específicamente dirigido hacia las artes visuales. Contando entonces con esta herramienta, decidimos abordar en un curso de Fundamentos de Programación Estructurada, el código del diseño junto con el código informático bajo una serie de premisas. En primer lugar, que las computadoras pueden realizar operaciones complejas y tan grandes que serían imposibles de procesar para un ser humano, pero requieren la capacidad de éste para generar nuevas operaciones para las que no han sido programadas. En segundo lugar, que tanto el código de diseño como el código informático han respondido y siguen respondiendo a las necesidades de una época y de un contexto histórico. Y en tercer lugar, que precisamente por este último punto, no solamente valdría la pena enseñarlos de forma

conjunta cuando se trata de una materia como la que nos aqueja en una licenciatura de Diseño, sino, más importante aún, no pueden ser aprendidos de manera aislada respecto a nuestro contexto social.

Acercándonos a las propuestas teóricas del psicólogo Lev Vygotsky, propusimos que en un momento como el nuestro, de contingencia sanitaria global, valía la pena conectar el código del diseño y el código de la informática a través de las emociones vividas por los estudiantes en el confinamiento. A través de un método que abarcó las 9 semanas del trimestre 20-P de la licenciatura de Diseño y tomando en cuenta las condiciones de los alumnos para participar en el curso, se les facilitaron todos los materiales por medio de una página de internet. Los resultados mostraron que este método resultó idóneo para el momento en que se encontraban los alumnos, pues no solamente se logró unificar el código del diseño y el código informático, sino que estas conjunciones funcionaron de una manera que rebasaron las expectativas.

En los productos finales de la materia, que consistían tanto en un programa informático sencillo como en la documentación de su proceso de diseño, demostraron que todos y cada uno de los elementos que fueron desarrollando durante cada semana quedaron integrados de una manera global. La planeación y las reflexiones que se les pidieron a los alumnos desde la primera semana, junto con la introducción donde se enseñó la relación diseño-informática, quedó de tal forma manifiesta en el desarrollo de sus programa que al final el resultado de los alumnos no sólo funcionó como traductor de emociones, que era el objetivo principal, sino como instrumento pedagógico del diseño. Del mismo modo, algunos de los elementos que componen la planeación del código informático, tales como el diagrama de flujo, adquirieron una estética acorde con un proyecto de diseño integral. Es decir, en su documentación no solamente cumplían con las expectativas de mostrar su proceso, sino que esta muestra estaba diseñada acorde con su propio programa, cada uno.

Cada uno de los programas cumplió con las especificaciones mínimas de integrar botones, animaciones y elementos externos como música o alguna imagen. Pero más allá de ello, todos lograron una integración de los elementos tanto al interior de sus programa como con la estética de su documentación. Se logró que cada uno desarrollara un concepto que atravesaba todo su proyecto. Vale la pena destacar, sin embargo, que estamos hablando aquí de los alumnos que entregaron un proyecto final. Es verdad que hubo muchas deserciones, a saber, 30 de 80 alumnos. Sin embargo, como lo mencionamos en el método, habría que tomar en cuenta las condiciones en que los alumnos optaron por tomar la materia sin que los resultados no aprobatorios afectaran a su historial académico, además, por supuesto, de encontrarse, como cualquiera de nosotros, en condiciones inéditas de desasosiego respecto a la situación sanitaria global.

Los resultados de lo que aquí hemos estado hablando se pueden constatar en la exposición virtual que fue programada por los mismo alumnos a través de un pad colaborativo en línea. Sin embargo, aquí mostramos dos ejemplos, a saber, los dos que mejor demostraban los puntos que aquí

hemos defendido. Si bien no pueden ser generalizados estos resultados, lo que intentamos demostrar es que en su mejor desarrollo, todos los puntos en los que está basada esta investigación puede dar como resultado no solamente la comprensión de los temas requeridos dentro de la currícula académica, sin que los alumnos pueden ir mucho más allá, hacia una propuesta que incluso sirva como herramienta didáctica del diseño.

Como pudo verse en los ejemplos mostrados, la comprensión conjunta del código del diseño y el código informático desde su historia, el contexto en que fueron desarrollados y sus intenciones, no sólo facilita el aprendizaje de ambos, sino que promueve en los alumnos un pensamiento del diseño complejo, basado en una concepción proyectual y sistemática. Al mismo tiempo, propone un acercamiento a la programación que va más allá del cumplimiento de tareas o funciones y se hace posible conectarlo con las emociones y la vida en las condiciones contemporáneas. Cada alumno desarrolló incluso un estilo estético para su programa y la documentación del mismo. Aunque ambos ejemplos que mostramos tuvieron una preparación basada en los planteamientos de Kandinsky, cada uno lo llevó por un camino diferente. Mientras que la primera alumna que mostramos decidió mantenerse apeada a la transmisión de los saberes de las teorías kandinskianas, realizando tal tarea de una manera impecable, el segundo lo llevó hacia la expresión artística que vinculó con otra de sus materias que cursó en el mismo periodo de tiempo. Lo que es cierto es que ambos no solamente se cumplieron los objetivos, sino que fueron más allá, dándole un sentido a su aprendizaje remoto y su práctica del diseño aún en tiempos de incertidumbre, algo que por sí mismo ya es valioso.

## Referencias

- Abascal, R., López, E. y Zepeda, S. (2015). *Hola Mundo con Processing*. México: Universidad Autónoma Metropolitana.
- Breas, C. y Fry, B. (2007). *Processing. A Programming Handbook for Visual Designers and Artists*. Massachusetts: The MIT Press.
- Gompertz, W. (2012). *¿Qué estás mirando? 150 años de arte moderno en un abrir y cerrar de ojos*. Madrid: Taurus.
- Joyanes, L. (2008). *Fundamentos de programación*. Madrid: McGraw-Hill.
- Kandinsky, W. (1989). *De lo espiritual en el arte*. México: Premia.
- Lupton, E. y Miller, A. (1994). *El abc de la Bauhaus y la teoría del diseño*. Barcelona: Gustavo Gili.
- Rodríguez Morales, Luis (1995). *El diseño antes de la Bauhaus*. México: Designio.
- Vygotski, L. (1979). *El desarrollo de los procesos psicológicos superiores*. Barcelona: Crítica.

**Mario Morales:** Universidad Autónoma Metropolitana, Unidad Cuajimalpa; Departamento de Teoría y Procesos del Diseño; Cuerpo Académico: Estudios sobre la Imagen, el Sonido y la Cultura Digital; CDMX, México; mmoralesd@cua.uam.mx.

# El código del diseño y el código de la programación: Pedagogías en tiempos de pandemia

---

INFORME DE ORIGINALIDAD

---

0%  
ÍNDICE DE SIMILITUD

---

FUENTES PRIMARIAS

---

EXCLUIR CITAS

DEACTIVADO

< 1%

EXCLUIR BIBLIOGRAFÍA ACTIVADO

EXCLUIR  
COINCIDENCIAS