

La ontología del software: un análisis de su dimensión estructural por medio del esquema
ontológico de Platón

Deisy Julieth Bernal Navas

Trabajo de Grado para Optar por el título de Magister en Filosofía

Director

Jorge Francisco Maldonado Serrano

Doctor en Filosofía

Universidad Industrial de Santander

Facultad de Ciencias Humanas

Escuela de Filosofía

Maestría en Filosofía

Bucaramanga

2025

Dedicatoria

Este documento es la materialización de los sueños de una familia que siempre ha creído en la educación. Esta tesis está dedicada a las mujeres de mi familia, primas, tías, abuelas, que devinieron en madres y optaron por canjear sus sueños por los de otros. Estas líneas están inspiradas en los hombres de mi familia que trabajaron desde sus primeros años de vida y vivieron el yugo de la desigualdad, sacrificios que nos han permitido soñar con lo que hoy se ha alcanzado, pues soy la primera persona de mi familia en obtener un título de Maestría. Anhele que este logro sea uno de los muchos que acompañen nuestra historia.

Este logro también está dedicado a Jefferson Albarracín Rojas, su existencia material trascendió en todos aquellos que le conocimos y ha inspirado actos de solidaridad y amor. Su deseo por estudiar Filosofía no pudo ser conseguido, pero su nombre estará siempre en mis reflexiones filosóficas más profundas.

Agradecimientos

A mis padres, pues por su apoyo y amor incondicional he conseguido alcanzar este sueño, gracias por enseñarme el valor de trabajo y el sacrificio, por creer en mis sueños y celebrar mis logros más pequeños. A mi hermosa familia, tíos, tías, primos, primas, nonos, porque siempre he conseguido refugio, cuidado, amor y apoyo en sus palabras. Gracias a Leito, Brandon e Iván por ser mis hermanitos y siempre apoyarme incondicionalmente, sus palabras fueron el impulso que necesitaba en muchos momentos de mi formación.

A Fabio Jaimes por no solo ser mi pareja; también, mi mejor amigo, gracias por hacer tuyos mis sueños, por levantarme cuando más lo necesitaba, por acompañarme, por confiar y creer en mí desde siempre, por construir un amor desde la admiración y el respeto. A mis amigas, Paola Bravo y Melisa Barragán, por sus cálidos abrazos y palabras de afecto que me han acompañado en los momentos en los que más lo necesitaba, gracias por crecer junto a mi y apoyarme en cada paso.

Gracias a la Universidad Industrial de Santander, la Escuela de Filosofía y sus administrativos, a sus profesoras, profesores y a mi director Jorge Francisco Maldonado Serrano, que me acompañaron en esta formación, gracias por la dedicación y la paciencia que siempre han tenido. Sin sus múltiples formas de apoyo no hubiese sido posible alcanzar este logro, gracias por buscar superar las barreras de desigualdad que permitieron mi formación.

Tabla de contenido

Introducción.....	9
1. El mundo está escrito en un lenguaje matemático	12
1.1 Del <i>devenir</i> al <i>eidos</i> , un estudio de la ontología de Platón	13
1.1.1 El <i>devenir</i> mediado por la <i>Aisthesis</i> , aquello que se genera siempre y nunca es. 14	
1.1.2 El <i>Eidos</i> mediado por la <i>episteme</i> , lo que siempre es y no se genera	17
1.1.3 La línea ontoepistemológica de los libros VI y VII de <i>República</i>	18
1.2 El triángulo como la primera especie.....	24
1.2.1 El <i>Timeo</i> como un compendio de la matemática de su tiempo	25
1.2.2 El <i>Eidos</i> y el devenir en <i>Timeo</i>	26
1.2.3 Los elementos y su correspondencia geométrica	27
2. El software: su concepción ontológica.....	35
2.1 Origen, desarrollo y evolución del software	36
2.1.1 Etapas de evolución y desarrollo del software	38
2.1.2 Clasificación del software	40
2.2 La programación como escritura que define el Software	41
2.2.2 Etapas del desarrollo histórico de la programación.....	42
2.2.3 Paradigma de programación.....	43

2.2.4 La Programación Orientada a Objetos y el <i>Software de Desarrollo</i>	46
2.2.5 La Interfaz de Usuario (UI) y la Unidad de Procesamiento Gráfico (GPU)	48
3 Ontología del software, una lectura desde Platón.....	50
3.1 La ontología de Platón, conceptos y nociones	51
3.2 El software, nociones y conceptos	54
3.3 La dimensión estructural del software a la luz de la ontología de Platón	55
Conclusiones	61
Referencias bibliográficas	63

Índice de ilustraciones

Ilustración 1. Primer esquema de la Línea ontoepistemológica	19
Ilustración 2. Segundo esquema de la Línea ontoepistemológica	21
Ilustración 3. Tercer esquema de la Línea Ontoepistemológica.....	53
Ilustración 4. Esquema final de la Línea Ontoepistemológica en relación con la Ontología del Software.....	58

Resumen

Título: La ontología del software: Un análisis de su dimensión estructural por medio del esquema ontológico de Platón.*

Autora: Deisy Julieth Bernal Navas**

Palabras clave: Software, dimensión estructural, ontología, pantalla, Platón.

Descripción:

Los dispositivos digitales necesitan para su funcionamiento un sistema interno que permite su correcto desempeño, este sistema se reconoce como software y se teoriza que se presenta en al menos dos dimensiones básicas, la dimensión serial y la dimensión estructural. Esta investigación se enfoca en aportar a la comprensión de la dimensión estructural, para esto se usa como recurso el pensamiento del filósofo griego Platón, quien en su proyecto ontológico expone condiciones y características que permiten entender la realidad y diversos fenómenos; incluso, encuentra réplicas en otra forma de existencia como lo es la del software. De esta forma, esta investigación está fundamentada en las siguientes aseveraciones: primero, que la forma de realidad que implica la existencia del software puede llamarse ontológica; y segundo, que esta forma de existencia puede entenderse bajo la interpretación del esquema ontológico de Platón. Según lo dicho, entonces, esta investigación espera ofrecer un aporte a la respuesta acerca de cómo puede entenderse la ontología del software en su dimensión estructural; para eso, se usará el esquema ontológico del filósofo ateniense como instrumento que permite ilustrar la idea del software como proceso de aparición fenoménica y, por tanto, sujeto a un estudio ontológico. Es decir, se afirma que, en el pensamiento ontológico del filósofo pueden encontrarse soportes suficientes para comprender el software en su dimensión estructural.

* Trabajo de Grado

** Facultad de Ciencias Humanas. Escuela Filosofía. Maestría en Filosofía. Director: Jorge Francisco Maldonado Serrano. Doctor en Filosofía

Abstract

Title: Software Ontology: an analysis of its structural dimension through Plato's Ontological Framework*

Author(s): Deisy Julieth Bernal Navas**

Key Words: Software, structural dimensión, ontology, screen, Plato

Description: Digital devices require an internal system for their proper functioning, which is recognized as software and is theorized to present itself in at least two basic dimensions: the serial dimension and the structural dimension. This research focuses on contributing to the understanding of the structural dimension, using the thought of the Greek philosopher Plato as a resource. In his ontological project, Plato exposes conditions and characteristics that allow us to understand reality and various phenomena, even finding replicas in another form of existence, such as software. This research is based on the following assertions: first, that the form of reality implied by the existence of software can be called ontological; and second, that this form of existence can be understood under the interpretation of Plato's ontological schema. As stated, this research aims to provide a contribution to the answer regarding how the ontology of software in its structural dimension can be understood. To achieve this, Plato's ontological schema will be used as an instrument to illustrate the idea of software as a process of phenomenal appearance, and therefore subject to ontological study. In other words, it is affirmed that sufficient supports can be found in Plato's ontological thought to understand software in its structural dimension.

* Degree Work

** Faculty of Human Sciences. Philosophy School. Master's Degree in Philosophy.
Director: Jorge Francisco Maldonado Serrano. Ph.D. in Philosophy.

Introducción

La llamada era digital ha tenido un amplio impacto en la forma como se experimenta el mundo. Los dispositivos digitales son objetos indispensables en la vida diaria, fundamentales para los diferentes roles y tareas del quehacer diario. Así, las personas cada vez más jóvenes usan y pasan más tiempo frente a las pantallas, a la vez que perciben el fenómeno proyectado en los monitores, es decir, la humanidad está expuesta, con mayor frecuencia, a la aprehensión de las imágenes proyectadas por estos dispositivos digitales. Pero ¿se conoce qué existe detrás de toda esta proyección? ¿Qué se podría decir sobre el *ser* de estos fenómenos?

Académicos del mundo de la tecnología podrían entregar una respuesta a estas preguntas, por ejemplo, apelar al funcionamiento de código, el lenguaje binario, los algoritmos y la relación que estos tienen entre sí, es decir, apelar al funcionamiento del software. Pero esta respuesta está lejos de satisfacer una perspectiva filosófica, pues, sería pertinente examinar, dado que se afirma que estos fenómenos funcionan desde el Software, qué es eso del software y cómo puede entenderse. Por lo menos, se reconoce que este puede entenderse como un dispositivo (aparato) tecnológico digital-computacional que se conforma de energía, circuitos, código e intencionalidad y se presenta en al menos dos dimensiones básicas: (I) la dimensión serial que se refiere al flujo de energía, circuitos y código y, (II) la dimensión estructural que se refiere a la estructura abstracta de todo software, es decir, su código, variables y cálculos (Maldonado Serrano et al., 2020).

De esta forma, esta investigación busca acercarse a la concepción de la *Ontología del Software*, tiene que ver con su ontología porque se estudia su forma de existencia y realidad. Para ello se basa en su dimensión estructural, pues se encarga de examinar, principalmente,

la forma abstracta del software. Con el objetivo de entender esta dimensión se toma como referencia el esquema ontológico de Platón, cuyo pensamiento y propuesta ontológica ha sido motivo de influencia filosófica durante siglos, pues se tiene la hipótesis de que en el filósofo ateniense se encuentran recursos suficientes para entender este tipo de realidad.

Según lo dicho, este trabajo se divide en tres partes fundamentales. En primera instancia, se dedicará a señalar dos elementos importantes del llamado *dualismo platónico*, a saber: el *Eidos* y el *devenir*, además, como es frecuente dentro del pensamiento del filósofo griego, la exposición de los intermedios resulta clara para acceder a un contenido holístico de la realidad, el elemento que completa triada ontológica de Platón es la *matemática*, presentada para la noción de *chora*, pues, según el filósofo, la realidad perceptible está escrita en un lenguaje matemático:

El orden matemático instaurado por el demiurgo (...) se trata de una realidad intermedia entre lo sensible [se refiere a al devenir] y lo inteligible [se refiere al *Eidos*] que presenta el aspecto de un encabestramiento de círculos que guardan entre ellos relaciones matemáticas y que en el universo explican todos los movimientos tanto físicos como psíquicos (Zamora, 2010, p. 69).

Para conseguir este objetivo, esta investigación se centrará en la descripción geométrica-matemática del *Timeo*. Además, tomará como referencia los libros VI y VII de *República* y algunos pasajes clave del *Teeteto*. El uso de estas obras tiene una intención estratégica, por un lado, al tratarse de obras escritas en la etapa de madurez y vejez del pensador, tiene como consecuencia que su pensamiento y teoría ha sido desarrollados de manera más completa y precisa; además, es en estas obras donde describe en mejor medida su esquema ontológico. Las traducciones que se usarán de las obras son las siguientes: en lo

que respecta a la traducción del *Timeo*, se tomará como referencia el texto traducido por Jorge María Zamora Calvo, publicado en el año 2010 por Abada Editores. Para el *Teeteto*, se usará la traducción de Álvaro Vallejo Campos, publicada por la editorial Gredos en el año 1988. Y, para *República*, también se usará la versión de Conrado Eggers Lan, publicada por la editorial Gredos en el año 1988. Se considera que estas versiones son las más apropiadas para esta investigación, pues guardan fidelidad en su traducción con los textos originales.

En segunda instancia, este trabajo indicará las nociones básicas que son fundamentales para comprender el funcionamiento del software, en específico, se enfoca en los elementos necesarios para comprender su dimensión estructural, esto es: algoritmo, código y lenguaje de máquina. Para este acápite se tomarán como referencias diversas fuentes bibliográficas, especialmente provenientes de la Ingeniería del Software, como el texto, *Ingeniería del Software* de Guillermo Pantaleo y Ludmila Rinaudo, publicado en el año 2015 por la editorial Alfaomega.

Finalmente, en el último momento de esta investigación, se relacionará la dimensión estructural del software y sus elementos con las nociones desarrolladas de la Ontología de Platón, a la vez que se usa como intermedio la descripción geométrica-matemática del *Timeo* y la línea ontoepistemológica de *República*, que se ilustrará y acompañará esta investigación desde el primer capítulo hasta el último, además, en el capítulo final, también se señalará, las distancias entre ambas teorías.

1. El mundo está escrito en un lenguaje matemático

Al explorar la ontología de Platón resulta inevitable aludir a su célebre *teoría de las ideas*, de la cual deviene el también conocido *dualismo platónico*, pues estos elementos constituyen las nociones más paradigmáticas y representativas de su filosofía. No obstante, antes de adentrarse en la definición y la comprensión de esta teoría, es necesario, para los objetivos de esta investigación, reconocer que el pensador ateniense se negó a separar el proyecto ontológico del epistemológico. Esto tiene como consecuencia, que, en busca de ofrecer explicaciones acerca de la naturaleza del conocimiento, el filósofo se remita a los lugares propios de su ontología, de esta forma la separación entre ontología y epistemología puede resultar indiscernible.

Así como Platón infirió que los objetos que la ciencia define no pueden ser identificados con las cosas indefinidamente variables del mundo físico sensible, reconoce que existe una relación entre los elementos del mundo sensible y las ideas, esta relación fue llamada *participación* (Taylor, 2005). En razón de la escisión ontológica de Platón y, en vista de la figura de la participación como alternativa de vínculo y mediación entre lo material y lo inmaterial, este texto empieza por centrarse en el primer lugar descrito por el pensador en su proyecto, esto es, en términos epistemológicos: la *percepción* o *aisthesis*; y, en términos ontológicos: el *devenir*. Estos se proponen como un punto de partida natural y necesario para comprender la realidad¹.

¹ Sobre cómo se relaciona la sensibilidad y la percepción con el conocimiento verdadero en Platón, confróntese mi tesis de pregrado *El lugar del eros en la episteme de Platón: una interpretación de la aisthesis en la anábasis del conocimiento* (2021).

Con el fin de obtener una comprensión más profunda de estos conceptos, en este capítulo se abordará, en primera instancia, como se advirtió previamente, y, en diálogo con la teoría ontoepistemológica del filósofo, el lugar del *devenir* y su referente epistemológico la *aisthesis* o percepción; enseguida de esto se encontrará el estudio del siguiente momento de su proyecto, es decir, se trata del análisis de la noción de *eidos* y su equivalente en términos del conocimiento, la *episteme*. De esta forma, la primera parte de este capítulo se aventura a señalar los conceptos más importantes y necesarios para comprender el proyecto ontológico y epistemológico del autor, y con esto entender también la reconciliación que realiza el pensador griego con los conocimientos de geometría adquiridos en su tiempo, idea que se aborda en la segunda parte de este capítulo.

1.1 Del *devenir* al *eidos*, un estudio de la ontología de Platón

Los conceptos que acompañan al fundador de la academia de principio a fin en su doctrina sobre el *ser* son fundamentalmente el *devenir* y el *eidos*. Tal como lo menciona el filósofo griego, Platón (2010) “Sin duda, lo más importante de todo es comenzar por el principio según la naturaleza del asunto” (29b). Y el principio de la naturaleza consiste en identificar, en primera instancia, qué son y qué significan estas nociones. Genealógicamente estos elementos están unidos gracias al lenguaje matemático usado por el demiurgo, artífice del mundo, pues como sostiene:

Como el dios había querido que todas las cosas fueran buenas y no hubiera en lo posible nada malo, tomó entonces todo cuanto era visible, que no estaba en reposo, sino que se movía sin

orden ni concierto, y lo condujo del desorden al orden, por considerar este absolutamente mejor que aquel (Platón, *Ti*, 29e).

El discernimiento que se conseguirá de estos elementos, gracias al marco conceptual ofrecido por el discípulo de Sócrates, permitirá comprender y entender que, al menos términos de la Filosofía de Platón, es posible entablar un diálogo que permite comprender la Ontología del Software.

1.1.1 El *devenir* mediado por la *Aisthesis*, aquello que se genera siempre y nunca es

La noción de *devenir* en Platón parece ser el resultado de la herencia heraclítica, pues, en algunos lugares de su pensamiento alude al filósofo de Éfeso, como *Banquete* (207d), *Crátilo* (402a) y *Teeteto* (152a). Es posible que la noción de devenir provenga de Πάντα ῥεῖ; idea que se traduce como *todo fluye*, se trata de una concepción cosmológica sobre la cual se sostiene que todo está en constante cambio. Para entender la noción de *devenir* este término se presentará en conjunto con la explicación epistemológica que le acompaña. En el gran marco de este proyecto el equivalente del *devenir* es la *aisthesis*.

Así pues, un lugar donde se discute la realidad de la *aisthesis*, entendida como percepción, y con esta, la del *devenir*, es el *Teeteto*, obra que se pregunta sobre la naturaleza del conocimiento. En la búsqueda de la respuesta a esta cuestión se menciona la postura de Protágoras, según la cual “el hombre es medida de todas las cosas” (Platón, *Tht*, 152a), definición que es acompañada por el pensamiento de Heráclito, de donde se sostiene que el universo es movimiento, y hay dos clases de movimiento ilimitadas en número: la que actúa

y la que recibe la actuación, y de la unión de ambas se engendra, por un lado, lo perceptible, y, por otro, la percepción:

nada es en sí y por sí, tal y como decíamos antes, sino que es en la unión de unas con otras como todas las cosas surgen en toda su diversidad a partir del movimiento, ya que, como ellos dicen, no es posible concebir en firme que lo que ejerce la acción y lo que la recibe sean algo definido independientemente uno de otro (Platón, *Tht*, 157a).

Estas alusiones conducen a los interlocutores del *Teeteto* a la premisa principal de que el saber es percepción. Pero, debido al constante movimiento de todas las cosas, es decir, debido al *devenir*, ninguna tiene un ser único en sí misma y por sí misma, ni lo activo, ni lo pasivo, así, no es correcto afirmar que las cosas son, porque ello les atribuye estabilidad, de acuerdo con su naturaleza; es más adecuado afirmar que *están en proceso de llegar a ser* (Platón, *Tht* 157a-c). Para llegar a tener conocimiento de un objeto sensible es necesario que se reflexione acerca del ser y de la utilidad del objeto, es decir, el ser no se puede alcanzar en la percepción (Gerena, 2009), ya que como se sostiene en Cornford (2007) “Es verdad que los órganos y objetos de la percepción cambian continuamente; y si esta fuera la única forma de conocer, no habría conocimiento. El conocimiento requiere términos que tengan un significado fijo y verdades que sean siempre verdaderas” (pág. 136). La respuesta a la afirmación de que el saber es percepción se desarrolla en mejor medida en el siguiente argumento:

Sóc. - Y bien, ¿Puede uno alcanzar la verdad de algo sin alcanzar su ser?

Teet. - Imposible

Sóc. - Pero, si uno no alcanza la verdad acerca de una cosa, ¿puede llegar a saberla?

Teet. -Claro que no, Sócrates.

Sóc. - Por consiguiente, el saber no radica en nuestras impresiones, sino, en el razonamiento que hacemos acerca de éstas. Aquí, efectivamente, es posible aprehender el ser y la verdad, pero allí es imposible. (...) Y decimos que esto [el percibir] no participa en la aprehensión de la verdad, pues no participa en la aprehensión del ser (...) luego tampoco en la aprehensión del saber (Platón, *Tht*, 186d - 186e).

Como consecuencia del análisis realizado por el pensador griego en *Teeteto*, se reconoce que la *aisthesis* y el *devenir* son dos elementos que están intrínsecamente relacionados. Por un lado, debido a las características de la *aisthesis* esta es designada en el terreno epistemológico, es decir, esta puede ser estudiada en relación con el proceso del conocimiento; por otro, el *devenir* es presentado como el equivalente ontológico de la *aisthesis*. En otras palabras, la *aisthesis* es el estudio del *devenir*, por esto Platón considera que, en tanto conocimiento verdadero, esta no es confiable, de ahí que, en *República* le atribuya otro concepto para referirse a este espacio del saber, le denomina *doxa*. El conocimiento verdadero no puede tener su raíz en el *devenir*, en contraste, su génesis debe estar fundamentado en aquello que no cambia y siempre permanece idéntico, elemento que se desarrolla en el siguiente acápite.

1.1.2 El *Eidos* mediado por la *episteme*, lo que siempre es y no se genera

Según lo dicho, si la percepción no es algo que es, sino, que está en proceso de llegar a ser, debe existir, entonces, algo que es, esto es el *Eidos*. Este concepto tiene un sentido lógico, denota principalmente una clase de cosas permanentes, tanto εἶδος como ἰδέα provienen del mismo verbo ἰδεῖν que significa ver, paradójicamente el sentido original de esta palabra tiene que ver con el aspecto o apariencia y no tiene una connotación intelectual (Gómez, 1986). Con todo, las características del *eidos* difieren considerablemente de los objetos de la *aisthesis*:

Las formas [también traducido como *eidos*] no admiten ninguna clase de cambio, mientras que la multiplicidad de las cosas sensibles [propias del devenir] nunca permanece igual. En *Fedón* y *República* se habla constantemente del mundo ideal como excluyendo cualquier cambio, y siempre se ha considerado que ésa es la condición necesaria para la existencia del conocimiento (Confort, 2007, pág. 306).

De esta forma, como se sigue en *República*, el filósofo ama “aquel estudio que les hace patente la realidad siempre existente y que no deambula sometida a la generación y a la corrupción” (Platón, *Rep*, 485b). El estudio al que se alude, aquel que se encarga de la realidad siempre existente es denominado *episteme*. En esta obra, a la percepción, el *devenir* y los elementos que la acompañan, es decir, el género de lo que se ve, les es entregado una noción que funge como contraste de la *episteme*, esta es: la opinión o *doxa* y es contraria al estudio que es propio del filósofo. Si el alma presta atención a estos objetos no obtiene consecuencias favorables: “cuando se vuelve hacia lo sumergido en la oscuridad, que nace y perece, entonces opina y percibe débilmente con opiniones que la hacen ir de aquí para allá, y da impresión de no tener inteligencia” (Platón, *Rep*, 508d).

En correspondencia con la teoría ontológica del autor, siempre en concordancia con su teoría epistemológica, se puede inferir que por sus características, el equivalente epistemológico del *eidos* es la *episteme*; del mismo modo como el *devenir* se presenta como el objeto de estudio de la *aisthesis*, el *eidos* se expone como el objeto de estudio de la *episteme*, además, se reconoce que este y no aquel es el estudio que es propio del filósofo, ya que se trata del estudio de aquello que es siempre permanente.

1.1.3 La línea ontoepistemológica de los libros VI y VII de *República*

La dualidad entre *devenir* y *eidos* se ilustra en mejor medida con la distinguida imagen de la línea expuesta por el pensador en el libro VI de *República*. Allí, Platón hace uso de una analogía geométrica con el objetivo de esclarecer su teoría, afirma:

Piensa entonces, como decíamos, cuáles son los dos que reinan: uno el del género y ámbito inteligibles [es decir, el *eidos*]; el otro, el del visible [es decir, el *devenir*], y no digo “el del cielo” para que no creas que hago juego de palabras. ¿Captas estas dos especies, la visible y la inteligible? (...) Toma ahora una línea dividida en dos partes desiguales; divide nuevamente cada sección según la misma proporción, la del género de lo que se ve y la otra la del que se entiende, y tendrás distinta oscuridad y claridad relativas” (Platón, *Rep*, 510a).

Según esta descripción la línea tendría, en primer lugar, una división desigual; en segundo lugar, cada uno de los segmentos estaría subdividido en dos partes iguales, y, en tercer lugar, uno de estos segmentos está designado a la *doxa* referido al *devenir*; y el otro, a la *episteme*, referida al *eidos*. Una primera representación de la línea podría ser la siguiente:

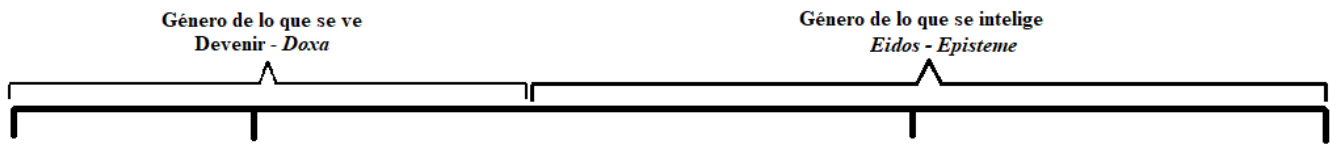


Ilustración 1. Primer esquema de la Línea ontoepistemológica

Esta imagen se complementa y relaciona estrechamente con la icónica alegoría de la caverna, una de las metáforas más emblemáticas de la filosofía de Platón, expuesta en el libro VII de *República*. En relación con ello, la línea empieza por marcar un fuerte contraste de: claridad-oscuridad, es decir, a cada extremo de la línea le corresponde una representación visual, esta puede ser clara u oscura. En seguida de ello, el filósofo empieza por referirse al primer segmento de la línea, es decir, el referido a la *doxa*, y prosigue con describir los elementos que contiene cada una de sus subdivisiones. Sostiene:

(...) así tenemos, primeramente, en el género de lo que se ve, una sección de imágenes. Llamo “imágenes” en primer lugar a las sombras, luego a los reflejos en el agua y en todas las cosas que, por su constitución, son densas, lisas y brillantes, y a todo lo de esa índole (...) Pon ahora la otra sección de la que ésta ofrece imágenes, a la que corresponden los animales que

viven en nuestro derredor, así como todo lo que crece, y también el género integro de las cosas fabricadas por el hombre (Platón, *Rep*, 510a-b).

De esta forma, Platón le otorga un nombre a cada uno de estos subsegmentos, en ambos se encuentran imágenes, aunque difieren entre ellas, en el primero de estos se ubican las sombras y los reflejos en el agua, este es llamado *eikasia* o *conjetura*. En el segundo sostiene que se encuentran los animales y también las cosas fabricadas por el hombre, este subsegmento es llamado *pistis* o *creencia*. Después de describir cómo estaría distribuida la primera parte de la línea, procede a ilustrar el segundo segmento:

Ahora examina si no hay que dividir también la sección de lo inteligible (...) Por un lado, en la primera parte de ella, el alma, sirviéndose de las cosas antes imitadas como si fueran imágenes se ve forzada a indagar a partir de supuestos, marchando no hasta un principio sino hacia una conclusión. Por otro lado, en la segunda parte, avanza hasta un principio no supuesto, partiendo de un supuesto y sin recurrir a imágenes -a diferencia del otro caso-, efectuando el camino de las Ideas mismas y por medio de las Ideas (Platón, *Rep*, 510b-c).

También afirma que en este estado se hace uso de la geometría y otras artes afines (Platón, 511b). En la sección inteligible se hace uso de la facultad dialéctica y los supuestos se usan como peldaños hasta llegar a una conclusión. En este estado no se hace uso para nada de lo sensible, sino de las Ideas, y en dirección con ellas se busca concluir con Ideas (Platón, 511c). De esta forma, los dos subsegmentos del segmento de lo inteligible son llamados, respectivamente: pensamiento discursivo (*dianoia*) e inteligencia (*noesis*), en tanto son cognoscibles, participan de la verdad y son más cercanos a la claridad (Platón, *Rep*, 509e-511e).

Según la descripción entregada por el autor de *República*, una representación de la línea que incluya lo expuesto hasta ahora podría ser:

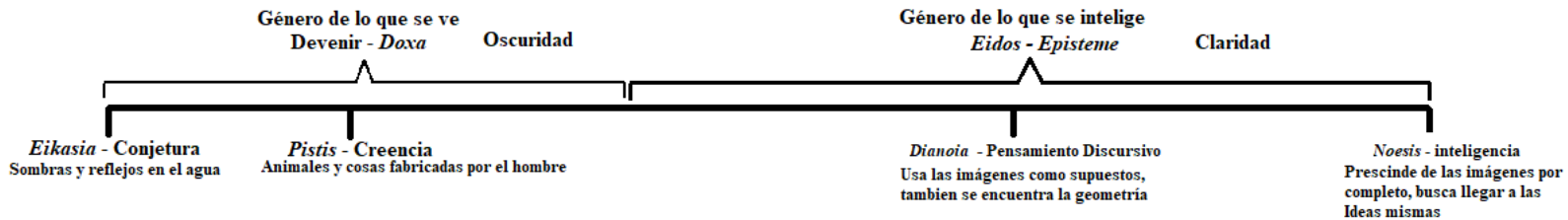


Ilustración 2. Segundo esquema de la Línea ontoepistemológica

Como es habitual en Platón en *Teeteto* la pregunta por la naturaleza del conocimiento termina por no ser dirimida, y las tres respuestas entregadas, el conocimiento es percepción (151d-e), percepción acompañada de opinión verdadera (187a-c) y la opinión verdadera acompañada de una explicación (201c), no satisfacen la búsqueda socrática del conocimiento. Con todo, pese a que no existe una respuesta clara sobre qué es el saber, en *República* se identifica un proceso ilustrado con la imagen de la línea que ha sido representada. De esta forma, se puede inferir que la actividad del hombre consiste en partir de sus opiniones sobre las imágenes del mundo o *eikasias* hasta alcanzar las ideas, como sostiene Perez Hernanz (2007): “No se empieza por los dioses, sino por las opiniones que se tienen de los dioses; no se empieza por las matemáticas, sino por las opiniones que se tienen de las matemáticas” (Pág. 241).

Este proceso de ascensión también se encuentra descrito en *Banquete*, donde Platón, a través de Diotima, describe el recorrido para ascender a la idea de lo bello, idea que se propone como el conocimiento más elevado y sublime que puede ser alcanzado por el hombre

(...) empezando por las cosas bellas de aquí y sirviéndose de ellas como de peldaños ir ascendiendo continuamente, en base a aquella belleza, de uno solo a dos y de dos a todos los cuerpos bellos y de los cuerpos bellos a las bellas normas de conducta, y de las normas de conducta a los bellos conocimientos, y partiendo de éstos terminar en aquel conocimiento que es conocimiento no de otra cosa sino de aquella belleza absoluta, para que conozca al fin lo que es la belleza en sí (Platón, *Smp*, 211c-d).

La imagen de la línea es una cuestión ampliamente discutida por diversos estudiosos de Platón; sin embargo, el objetivo de esta investigación busca señalar, especialmente, la relación *devenir- Eidos*, que puede identificarse en la descripción de la línea ofrecida por el ateniense. Como se sigue con Migliori (2009), “es fácil establecer una marcada separación entre imagen y el objeto del cual es imagen” (Pág., 206). Una de las múltiples réplicas que se realizan a la línea ontoepistemológica propuesta por Platón tiene que ver con el reproche de que, si el pensador le entrega la verdad y el ser a la segunda parte de la línea, cuál es la necesidad, entonces, de retratar y describir la primera parte que, además, está relegada al ámbito de lo que no es conocimiento y conduce a la ignorancia. Como sostiene Migliori (2009):

Creo que, al confirmar esta única línea, él quería subrayar la necesidad de amar siempre el todo, o bien, de recordar las limitaciones humanas, que también marcan al filósofo (...). La ‘línea’ instituye, pues, como es habitual en Platón, una relación bicondicional entre niveles de conocimiento y niveles de realidad: si hay distintos grados de ser (lo que también significa

ser-verdadero) de los entes, entonces existen diferentes grados de conocimiento; pero hay diversos grados de conocimiento (en primer lugar ‘opinión’ y ‘ciencia’, doxa y episteme); por lo tanto, existen diversos grados de realidad (pág. 216).

En el pasaje 510a Sócrates le pide a Glaucón que divida una línea de dos partes desiguales. Una pregunta que puede surgir después de esta afirmación es la siguiente: ¿cuál de los segmentos es más largo? Nicholas Denyer (2007) sostiene que la separación desigual de los segmentos se trata de una imperfección que es adecuada para los objetivos que Platón desarrollará más adelante:

It is central to Plato’s thought about images that an image always falls short of the original of which it is an image. That is why he finds it so appropriate to describe the relationship between perceptible and intelligible as a relationship between image and original. The divided line is itself another image. So if it too is defective, that would be all to the good (Denyer, 2007, pág. 296).

Según lo dicho existe, entonces, dos posibilidades de representación de la línea, una en la que el segmento correspondiente a aquello que es inteligible sea más largo, y otra, que indique el segmento correspondiente a aquello que es sensible sea más largo. Para la representación gráfica que se realiza en este trabajo, se toma como referencia la primera posibilidad, la razón de esto estriba en la postura expuesta por Miglori (2009), al tratarse de una metáfora que marca la dicotomía entre la relación imagen-modelo, o sí si se quiere, *devenir-eidos*, resulta razonable que el segmento que corresponde aquello que es sensible, en tanto copia de aquello lo que es modelo, sea imperfecto con respecto a aquel, de ahí que su longitud sea menor.

1.2 El triángulo como la primera especie

Dentro de su amplio corpus filosófico Platón empleará la figura triádica como recurso para sus explicaciones ontológicas, cosmológicas, epistemológicas e, incluso, políticas. De esta forma, el autor sostiene una división del alma en tres partes, la racional (*logos*), la irascible (*thymos*) y la concupiscible (*epithymia*) (*Rep.*, 436a– 441c); en su proyecto político establece una división social y comunitaria en tres clases: los productores, los guerreros y los filósofos (*Rep.*, 368c – 445e); también sostiene que cada una de estas clases tiene un tipo de virtud específica, con ello propone una composición triádica de la virtud: la templanza (*sophrosyne*), la valentía (*andreia*) y la sabiduría (*phronesis*) (*Rep.*, 427e- 429a).

Bajo el mismo razonamiento el filósofo ateniense sostiene que la realidad también está compuesta desde una figura triádica, dos de estos elementos ya fueron expuestos anteriormente: *eidos* y *devenir*. El tercer elemento que viene a completar este conjunto es llamado la $\chi\acute{o}\rho\alpha$ o *chora*, se trata de una realidad intermedia. “Entre los objetos sensibles y las formas inteligibles hay <<un tercer género que es eterno>> (52a), el <<espacio>> ($\chi\acute{o}\rho\alpha$), que constituye su principio de diferenciación” (Zamora, 2010, pág. 41). Pese a que la *chora* es un elemento de difícil comprensión debido a su carácter ambiguo, Platón sostiene que este, también llamado *material espacial*, es ordenado por el demiurgo a partir de una composición matemática y geométrica.

Esta descripción se encuentra en el *Timeo*, obra cúlpe de su proyecto filosófico. En este capítulo se examinará el estudio de esta realidad intermedia en relación con la explicación matemática y geométrica proporcionada por el fundador de la academia.

1.2.1 El *Timeo* como un compendio de la matemática de su tiempo

El *Timeo* o *sobre la naturaleza* es el diálogo que busca explicar el origen del universo conocido y fue escrito aproximadamente entre los años 388 y 386 a.C., al final de la vida de Platón. El filósofo espera describir en este libro el origen del universo, presentado como un macrocosmos y el origen el hombre, que se describe como un microcosmos, para esto toma como referencia los conocimientos matemáticos alcanzados en su momento: “De ahí que los límites de la explicación del universo que Platón propone coinciden con los límites de la matemática de su época” (Zamora, 2010, pág.10). Además, se considera que el *Timeo* es la primera obra escrita conservada donde aparecen los poliedros, que entraron en la reflexión filosófica y se conocieron como *los sólidos platónicos* (Lanza, 2015).

La disertación de *Timeo* inicia cuando termina la exposición realizada por Critias sobre la *polis* ideal, exposición que según algunos pensadores se puede leer como un resumen de *República* (Zamora, 2010). Critias le pide a *Timeo* que sea él quien haga la exposición sobre la naturaleza del universo, y que prosiga con la naturaleza de los hombres, el motivo de esto radica en que es él quien más sabe sobre astronomía de los otros interlocutores presentes, además es quien más ha dedicado tiempo a estas cuestiones (Platón, 27b). Antes de adentrarse en la lectura profunda del diálogo, es importante no pasar por alto los preludios que anteceden la explicación cosmológica de *Timeo*, especialmente el referente al *eikos logos* que *Timeo* se aproxima a pronunciar:

En cambio, si proponemos explicaciones que no sean menos verosímiles que las de otro, habrá que contentarse, al recordar que yo, el que habla, y vosotros, los jueces, tenemos una naturaleza humana, de modo que acerca de esto conviene que aceptemos un mito verosímil y no buscar más allá (Platón, *Timeo*, 29d).

En otras palabras, de acuerdo con los límites de la naturaleza humana, es pertinente que se acepte el discurso que se aproxima no sólo como algo verosímil, además, como lo más verdadero a lo que se puede acceder según la naturaleza humana. Platón propone su discurso verosímil como el único modo de explicar sistemáticamente lo que está sujeto al devenir, para lo cual introduce su *eikos logos* como un discurso explicativo que no es verdadero ni tampoco hipotético (Santa Cruz, 1986).

1.2.2 El *Eidos* y el devenir en *Timeo*

Antes de empezar su disertación, Timeo les pregunta a sus interlocutores: “¿Qué es lo que siempre es y no se genera, y qué es lo que se genera siempre y nunca es?” (Platón, *Tim.*, 28a). En otras palabras, les pregunta por los conceptos expuestos anteriormente, el *Eidos* y el *devenir*. Prosigue: uno de estos puede ser comprendido por la inteligencia, con la ayuda de la razón, esto es, el *Eidos* (Platón, 28a). El otro, es opinable por medio de la *doxa*, la cual se ayuda de la percepción sensible (*aisthesis*) y es irracional. Este se genera y se destruye, nunca es realmente, esto es, el *devenir* (Platón, 28a). Enseguida de ofrecer este axioma, se introduce la figura del demiurgo, que se presenta como la mejor de las causas, que copia mirando un modelo eterno, inmutable y perfecto, pues es necesario que este mundo

sea imagen de algo. De esta forma se revela, una vez más, de la misma manera como en *República*, la relación intrínseca entre la imagen y el modelo:

De este modo, entonces, acerca de la imagen y su modelo hay que establecer la siguiente distinción, al admitir que los discursos guardan una afinidad con aquello de lo que son intérpretes: los discursos acerca de lo que es firme, estable y manifiesto a la inteligencia deben ser firmes e infalibles (...) Pero, los discursos que se refieren a lo que es una copia de aquello, como es una imagen, han de ser verosímiles proporcionalmente a los primeros. Lo que es el ser a la generación, la verdad es la creencia (Platón, 29c).

En el libro VII de *República* se encuentra también una referencia a la relación imagen-copia, y la alusión a los límites de los objetos que hacen parte del mundo perceptible. “Estos bordados que hay en el cielo están bordados en lo visible, y aunque sean lo más bellos y perfectos de su índole, les falta mucho en relación con los verdaderos” (Platón, *República*, 529d). Y sostiene que estos bordados deben usarse como ejemplo para el estudio de los otros bordados, aquellos que están trazados “excelentemente por Dédalo o algún otro artesano o pintor” (Platón, *República*, 529e). Es decir, en relación con el *Timeo*, aquellos trazados por el artífice del mundo, el demiurgo, que se presenta entonces como aquello que permite explicar la unión de ambos géneros el *eidos* y el devenir, este tomó todo lo que era visible y lo condujo al orden (Platón, 29e).

1.2.3 Los elementos y su correspondencia geométrica

Pese a que el *Timeo* se presenta como un compendio de las matemáticas de su tiempo, no es el primer lugar donde Platón se refiere a la mismas. Por ejemplo, en *República*

se encuentra también una importante alusión a esta ciencia como una herramienta epistemológica

Sóc. - Pues éstos [los matemáticos] hacen lo mismo en la armonía que los otros en la astronomía, pues buscan número en los acordes que se oyen, pero no se elevan a los problemas ni examinan cuáles son los números armónicos y cuáles no, y por qué en cada caso.

Glau. - Hablas de una tarea digna de los dioses.

Sóc. -Más bien diría que es una tarea útil para la búsqueda de lo Bello y de lo Bueno, e inútil si se persigue de otro modo (Platón, *República*, 531c).

El lugar de las matemáticas en *República* escapa a las intenciones epistemológicas. Estas se presentan como un requisito necesario para la formación de los ciudadanos, ya que sólo bajo esta forma es cómo se adoptan dentro del proyecto educativo de Platón. Las matemáticas se presentan como unas articuladoras de la *polis*, pues son un elemento mediador que permite acceder desde una vida ordinaria a un ámbito político, filosófico y racional (Pérez, 2007). Y, como es frecuente dentro del corpus de Platón al tener una importancia epistemológica, tiene también una importancia ontológica; se presentan como un tipo de realidad que se encuentra entre lo sensible y lo inteligible, tal como lo describe su discípulo estagirita en *Metafísica*:

(Platón) afirma, además, que entre las cosas sensibles y las Formas existen las Realidades Matemáticas, distintas de las cosas sensibles por ser eternas e inmóviles, y de las Formas porque hay muchas semejantes, mientras que cada Forma es solamente una y ella misma (Aristóteles, *Metafísica*, 987b15).

Es en *Timeo* donde las matemáticas pasan de ocupar un lugar en la educación de los ciudadanos de la *polis* a un lugar en la explicación cosmológica dentro del pensamiento de Platón, y es que el alma del mundo mantiene un orden matemático entregado por el demiurgo. Esta está compuesta de una mezcla entre el ser indivisible, el ser divisible que deviene de los cuerpos y el ser intermedio, una mezcla de los dos precedentes (Zamora, 2010). La explicación matemática se presenta desde el momento en que Timeo se dispone a explicar la relación de los elementos naturales, fuego, tierra, agua y aire en la descripción del cuerpo del mundo, aunque la descripción del cuerpo del mundo (31b - 34a) precede, en *Timeo*, a la explicación del alma del mundo (34a - 40d). El alma es genealógicamente anterior a aquel: “Ahora bien, tanto en nacimiento como en excelencia, el dios constituyó al alma anterior y más antigua al cuerpo, para que fuera su dueña y gobernante” (Platón, *Timeo*, 34c).

Es en el pasaje 31b donde se hace mención al primero de los cuatro elementos, el fuego, que acompaña al demiurgo en la tarea de la creación del macro y microcosmos. Fuego, tierra, agua y aire, en ese orden, son los elementos usados por el demiurgo para la formación de todo que hace parte del mundo. El orden que introduce el demiurgo, por medio de los elementos y su correspondencia geométrica, manifiesta la simetría que hace posible la descripción matemática del universo, los poliedros regulares son la estructura eidética de los elementos (Lanza, 2015). De esta forma, el artífice necesita del fuego, pues es necesario debido a que lo que se genera debe ser visible y “sin fuego nada podría nunca llegar a ser visible” (Platón, *Tim.*, 31b). Bajo el mismo razonamiento emplea la tierra, pues lo generado debe ser tangible y corpóreo, y no podría ser sólido sin la tierra (Platón, *Tim.*, 31c). La justificación del uso de los elementos agua y aire tiene que ver con la idea de que para unir dos elementos debe existir uno tercero que los ponga en conexión y sostiene:

Pero, en realidad, convenía que este mundo fuera un sólido y, para armonizar los sólidos, nunca basta un único término medio, sino siempre dos. De este modo, el dios puso el agua y el aire en medio del fuego y la tierra, y lo hizo, en la medida de lo posible, siguiendo la misma relación proporcional mutua, así lo que es el fuego respecto al aire, lo sea el aire respecto al agua, y lo que es el aire respecto al agua, lo sea el agua respecto a la tierra, y, de esta forma, unió y constituyó el universo visible y tangible (Platón, *Tim.*, 32b).

Es en la creación del cuerpo del macrocosmos donde Platón menciona por primera vez las figuras geométricas, pues, le entrega al cuerpo del macrocosmos la forma geométrica más perfecta, la circular, esto para que tenga la característica de ser autárquico:

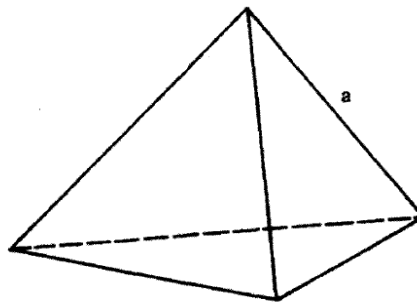
Le dio una figura conveniente y afín. Ahora bien, la conveniente al viviente que ha de contener en sí mismo a todos los vivientes sería una figura que contuviera en sí mismas a todas las figuras. Por eso la torneó con forma esférica, equidistante en todas sus partes del centro a los extremos, circular, la más perfecta y semejante a sí misma de todas las figuras, porque consideró que lo semejante es mil veces más bello que lo desemejante (Platón, *Timeo*, 33c).

A cada uno de los elementos del macrocosmos, fuego, tierra, agua y aire, les entrega respectivamente una figura poliédrica regular, pero, antes de esto, sostiene que la primera especie, esto es, la especie que antecede y permite la creación de los poliedros, es el triángulo rectángulo escaleno:

Comenzaremos con la primera especie, la más pequeña en su composición, cuyo elemento es el triángulo que tiene la hipotenusa de una longitud del doble del lado menor. Si dos de estos triángulos se unen por la diagonal, y esto se repite tres veces, haciendo coincidir las diagonales y los lados menores del mismo punto, se genera un único triángulo equilátero a partir de los seis (Platón, *Tim.*, 54e).

Este triángulo permite la formación de los tres primeros poliedros, pero, para la formación del último poliedro, es decir, del hexaedro, se hace necesaria la intervención de otro triángulo, el triángulo isósceles. Como afirma Lanza, el hecho de que no todos los poliedros se puedan constituir del mismo tipo de triángulo, y que sea necesaria la intervención de otro triángulo hace afirmar a Timeo, en 53c, lo siguiente: “todos los triángulos se desarrollan a partir de dos” (2015). Por medio de las formas, los cambios y las transformaciones de estas primeras especies, se crea el primer sólido, el poliedro regular de cuatro caras y seis ángulos equiláteros (Platón, 55a), el tetraedro, atribuido al fuego por ser el cuerpo más pequeño (Platón, 56a).

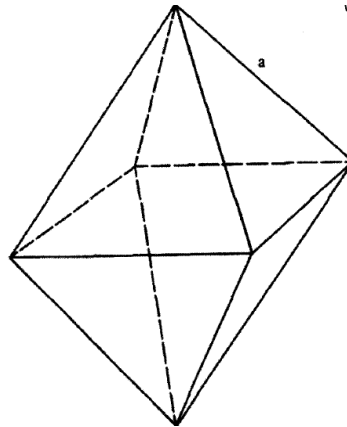
Figura 1: Tetraedro regular



Créditos: Zamora (2010, Pág. 439)

El segundo sólido también está compuesto por el mismo tipo de triángulo: tiene ocho caras triangulares equiláteras y seis aristas (Platón, 55a). El octaedro, atribuido al aire, por poseer un tamaño intermedio entre los poliedros (Platón, 56a).

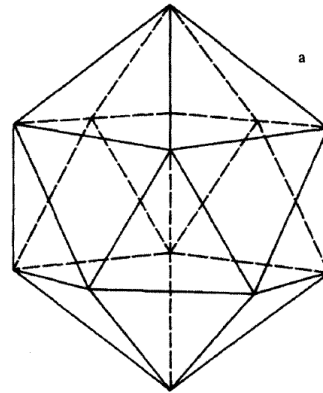
Figura 2: octaedro regular



Créditos: Zamora (2010, pág. 439).

El tercer sólido, compuesto también por la primera especie, tiene veinte caras triangulares equiláteras y treinta aristas, (Platón, 55b), el icosaedro, atribuido al agua por ser el más grande de los sólidos (Platón, 55a).

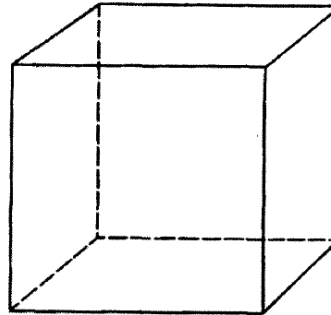
Figura 3: icosaedro regular



Créditos: Zamora (2010, pág. 440).

Finalmente, el cuarto sólido está compuesto por el triángulo isósceles, el cual se trata de la combinación de seis caras cuadradas y doce aristas, que dan como resultado el hexaedro (Platón, 55c), figura asignada al elemento de la tierra por considerarse este el más difícil de mover de los cuatro géneros y aquél que necesita las caras más estables (Platón, 55e).

Figura 4: hexaedro regular



Créditos: Zamora (2010, pág. 440)

En este punto es importante mencionar que las figuras en sí mismas no pueden ser percibidas por su pequeñez, de manera tal que se necesita de la unión de estas para que puedan ser aprehendidas:

Hemos de concebir, por cierto, todas estas figuras tan pequeñas que individualmente ninguna de las pertenecientes a cada uno de los géneros pueda ser percibida por nosotros en virtud de su pequeñez, pero cuando muchas se aglutinan, las masas que forman pueden verse (Platón, *Timeo*, 56b).

Esta idea tiene semejanza con la descripción de la percepción que se hace en el *Teeteto*, a través del argumento holístico del pasaje 204a-e, donde Platón sostiene que las únicas cosas que podemos percibir son individuales, concretas y naturales, complejas o simples, y una cosa compleja es la unión de cosas simples o elementos que pueden ser enumerados, por ese motivo podemos dar razón de ellos, solo cuando la enumeración es completa, conocemos todo cuanto podemos de esa cosa (Confort, 2007).

En definitiva, en este capítulo se hizo un recorrido por el proyecto ontológico de Platón, se examinaron los conceptos que enmarcan su propuesta teórica, así, se presenta el

concepto de *eidos* como forma de existencia inmutable, que existe independientemente de una realidad sensible, este concepto está mediado por la *episteme* presentada como el conocimiento que se basa en la comprensión de formas o esencias inmutables. También se abordó el concepto de *devenir* como una forma de existencia mutable, este existe en una realidad sensible y está mediado por la *aisthesis* o *percepción*; es decir, gracias a la percepción se puede acceder a un conocimiento en los objetos de la sensibilidad, este estado se clasifica, en contraste con la noción de *episteme*, en la categoría de *doxa*, que se traduce como opinión.

De la misma forma como se presentó la relación bicondicional entre *eidos* y *devenir*, y, por consiguiente, entre *episteme* y *doxa*, se enunció los lugares intermedios entre estos estados, en primer lugar, la *eikasia* o conjetura; en segundo lugar, la *pistis* o creencia, estos son estados del proceso de conocimiento, relegados al ámbito de la *aisthesis*, es decir, se tratan de un conocimiento jerárquico de los objetos de la sensibilidad. Mientras que, por el lado de la *episteme*, se encuentran, en tercer lugar, la *dianoia* o pensamiento discursivo y la *noesis* o inteligencia; es decir, se trata de un tipo de conocimiento que escapa del acto de percepción. En este estado el hombre se aleja de los objetos de la sensibilidad y emprende el recorrido en busca de los conceptos y los principios.

La parte final de este capítulo se dedicó a estudiar la matemática y su posición dentro del proyecto de Platón, pues, como se sostiene en *Timeo*, no solo se trata de una realidad intermedia entre el *devenir* y el *eidos*, también, esta es la condición necesaria que permite que todo aquello que hace parte de la percepción y, por consiguiente, del *devenir*, sea percibido. Además, el demiurgo, hacedor del macro y microcosmos, emplea la matemática para ordenar el mundo. Enseguida de eso, se describió la relación de los elementos naturales

con la correspondencia geométrica empleada por el artífice, de esta forma, se expuso la correspondencia fuego, tierra, agua y aire con los elementos, tetraedro, hexaedro, icosaedro y octaedro, respectivamente, con ello se demuestra que el uso de los poliedros determina, según el filósofo griego, la forma cómo se ven los objetos de la sensibilidad.

A partir de ahora, una vez comprendidas las nociones de la teoría ontológica de Platón, esta investigación se centrará en exponer los elementos y los principios correspondientes al software, pese a que se trata de una separación considerable del capítulo que le precede, esta exposición tiene como intención comprender los elementos básicos necesarios para llevar a cabo del acto de la programación, esto permitirá la comprensión de la articulación conceptual que se propone, según la cual, es posible comprender la ontología del software, en su dimensión estructural, desde la teoría ontológica de Platón. De esta forma, se expondrá la teoría alrededor del software, un recorrido por su historia, una aproximación a su clasificación y su relación con la programación. Posteriormente, a la luz de la comprensión de estos elementos, se articulará con las nociones propias de la ontología del filósofo ateniense.

2. El software: su concepción ontológica

Desde la última década, la humanidad ha sido testigo de una transformación sin precedentes de la forma en que vivimos y nos relacionamos. El avance tecnológico se presenta como el motivo principal de este cambio. El desarrollo de las tecnologías como la Inteligencia Artificial, la Realidad Virtual y el Internet de las Cosas, permite una creación de

sistemas que revolucionan la forma como interactuamos entre nosotros y también con el mundo que nos rodea.

Podría decirse que el fenómeno de transformación tecnológica fue el motivo de interés principal de esta investigación; pero, dada la dificultad de elegir solo uno de estos fenómenos, se decidió estudiar el software, que puede entenderse, en primera instancia, como un dispositivo primigenio que subyace en la mayoría de las tecnologías y los sistemas actuales. Así, este podría exponerse *como un alma*, en sentido platónico, que anima las máquinas y les permite procesar información o tomar decisiones.

Por esta razón, este capítulo está motivado, en primera instancia, en entender la noción de software de manera individual, para lo que se empieza por realizar un recorrido histórico de esta noción, a la vez que se reconocen sus etapas y desarrollo. Posteriormente, se alude a la clasificación del software, lo que permite identificar el *Software de Desarrollo*, una categoría representativa en términos de programación, que será fundamental para el recorrido de este capítulo. Una vez realizado lo anterior, se procederá a mencionar los tipos de programación y, con ello, se espera comprender a la programación como la actividad de crear el software. Por ello se mencionan brevemente los paradigmas de programación que se reducen esencialmente a dos: antes y después de las redes neuronales. Este recorrido tendrá como consecuencia la comprensión del proceso de Programación Orientada a Objetos o POO.

2.1 Origen, desarrollo y evolución del software

Según la Ingeniería de Software, el software se puede definir como un conjunto de instrucciones, datos y programas que permiten a una computadora o dispositivo electrónico realizar tareas específicas, procesar información y proporcionar un servicio (Pantaleo y Rinaudo, 2015). En esta disciplina se considera al software como “un conjunto de elementos que interactúan entre sí, y no como un bloque indiviso” (Cristiá, 2024, p. 2). Desde esta perspectiva, se asume como una herramienta que se encuentra en todos los dispositivos electrónicos que usamos en la actualidad. Entre sus características fundamentales se destaca que esta herramienta no tiene una forma física tangible, lo que quiere decir que el software depende del hardware para su funcionamiento, y que puede y debe ser programado, actualizado y modificado para ajustarse a las diferentes arquitecturas y plataformas. Con todo, la concepción que se tendrá del software en esta investigación no proviene exclusivamente de la Ingeniería del Software, a saber, en este trabajo se comprende el software como un dispositivo (aparato) tecnológico digital-computacional que se conforma de energía, circuitos, código e intencionalidad (Maldonado et al., 2020).

Una forma de entender este dispositivo puede ser a la luz de tres dimensiones, llamadas por Maldonado et al. en *Una ontología del software: series, estructura y función*, como: dimensión serial, estructural e intencional (2018). La *dimensión serial* permite interpretar el software como un proceso en que el convergen circuitos, energía y código, lo que quiere decir que el software no se reduce a solo uno de estos elementos. En este caso, se entiende el software como un dispositivo que requiere de la acción humana, es decir, requiere ser completada por un usuario. En lo que corresponde a la *dimensión estructural*, se trata de la estructura abstracta que responde a cualquier software, se refiere particularmente a su código, se entiende como la relación entre comandos y variables, esta dimensión permite

comprender el alcance de la tecnología digital y sus posibilidades (Maldonado et al., 2020). Por lo que corresponde a la *dimensión intencional*, esta, pese a que es mencionada aún no ha sido desarrollada por los autores del artículo mencionado.

Esta investigación inspira el título y el enfoque del presente texto, pues, al menos en términos de la ontología de Platón, es posible comprender la ontología del software a través, particularmente, de su *dimensión estructural* y no de su *dimensión serial*. El motivo de esto es que, la correspondencia con la línea ontoepistemológica puede hacerse con mayor claridad desde el examen del código de programación y sus implicaciones; mientras que, la dimensión serial requiere el estudio del hardware, estudio que no se abordará en esta investigación. Por lo pronto, las etapas de evolución y desarrollo del software se exponen de la siguiente forma.

2.1.1 Etapas de evolución y desarrollo del software

La palabra software se traduce literalmente como “mercancía suave” pues proviene de las palabras inglesas: “soft” que significa suave o flexible y “ware” que significa mercancía o producto. Esta palabra se originó en la década de 1950 y fue empleada por primera vez por el estadístico estadounidense John Wilder Tukey en su artículo *The Teaching of Concrete Mathematics*, publicado por la revista *American Mathematical Monthly* en el año 1958. En este artículo se usó el término software para distinguirlo del hardware, que se refiere a todos los componentes tangibles o circuitos de una computadora.

Los primeros desarrollos del software fueron inspirados en aplicaciones militares como el control de tiro y radares durante los años 50. Posteriormente, cerca de la década del 60 y con el advenimiento de los lenguajes de alto nivel y programas en el desarrollo del hardware, se empezó la producción de sistemas independientes del hardware. Allí los sistemas empezaron a tener un propósito comercial y eran empleados para resolver problemas referentes a las áreas de: explotación de petróleos, compañías de seguros y ventas en mercados (Pantaleo y Rinaudo, 2015).

En los años 70 tuvo lugar la llamada *crisis del software*, se trató de un periodo en el cual el tamaño de los sistemas y la complejidad de los problemas a resolver se vio afectada por la falta de madurez de las herramientas y los procesos de desarrollo, en este periodo, al desarrollar un sistema fue necesario implementar sus componentes por medio del hardware, esto debido a la poca confiabilidad del código y las dificultades de depurarlo (Pantaleo y Rinaudo, 2015). Posteriormente, en los años 80, se consideró necesaria la creación de una disciplina que se ocupara sistemáticamente del diseño y la construcción de los productos del software, así surgió la *Ingeniería del Software*. Como consecuencia de ello, en el año 1990, se publica el primer glosario de términos para la esta disciplina.

Por ese entonces estaba claro que el desarrollo de software consistía en administrar el proceso de construir o un producto o un sistema que cubriera las necesidades de los usuarios, probarlo, instalarlo en el ambiente productivo, mantenerlo y hacerlo evolucionar con los cambios del negocio. Por lo tanto, un aspecto de la calidad vinculado al software consistía en llevar adelante este proceso de forma que permitiera al proyecto asociado terminar en el tiempo planificado y dentro del presupuesto asignado (Pantaleo, 2012, pág. 21).

En esta misma década tiene lugar la aparición de un dispositivo que cambiaría de manera definitiva la forma como se entienden los sistemas de información: en 1981 IBM presenta su *Personal Computer* o PC, y diez años más tarde Microsoft se acentúa como un protagonista y líder mundial en el desarrollo de sistemas operativos. A mediados de los años 90 empiezan a aparecer sistemas operativos orientados en la interconexión de las PCs; así se llega al surgimiento de la Internet.

Desde los años 2000 el escenario de desarrollo del software ha tenido una gran ampliación debido a la aplicación de un hardware más poderoso. Se trata de una evolución que conlleva a un software de última generación con “lenguajes orientados a objetos, lenguajes deductivos, interpretados, intermedios, multiplataformas, arquitecturas orientadas a servicios, modelos de desarrollo (CMM, ITIL), metodologías ágiles (XP, Scrum)” (Pantaleo y Rinaudo, 2015, pág. 6). De esta forma, en ingeniería se propone una clasificación del software.

2.1.2 Clasificación del software

Debido al amplio uso que se tiene de este dispositivo se han clasificado sus funciones en cuatro categorías, pero, estas no se consideran mutuamente excluyentes; lo que quiere decir que algunos programas pueden pertenecer a más de una categoría.

- I) *Software de sistema*: se refiere a los programas que se encargan de la gestión de recursos del sistema, ofreciendo servicios básicos para el correcto funcionamiento del hardware. Este tipo de software se puede encontrar en sistemas operativos como Windows, controladores de impresoras o tarjetas gráficas.

- II) *Software de servicio*: se encarga de proporcionar servicios en línea, por ejemplo, en aplicaciones web o servicios en la nube, en aplicaciones como Facebook o Google Cloud.
- III) *Software de aplicación*: se encarga de realizar tareas específicas para el usuario, como procesar textos o realizar cálculos. Se puede encontrar en procesadores del texto como Microsoft, hojas de cálculo como Excel o videojuegos.
- IV) *Software de desarrollo*: es empleado por los desarrolladores para crear, probar y depurar software. Aquí se encuentran lenguajes de programación como Java, Python, C++ o entornos de desarrollo y herramientas de prueba (Palacios, 2020).

Al tratarse de un dispositivo en constante evolución esta clasificación también se renovará conforme a los cambios que se avecinan. Pese a que todos los momentos de esta clasificación implican en gran medida los lineamientos básicos de la programación, el *Software de desarrollo* cobija los objetivos de esta investigación, pues en este se puede señalar una relación cercana a la Programación Orientada a Objetos, tipo de programación fundamental para comprender la articulación conceptual que se propone.

2.2 La programación como escritura que define el Software

La programación puede definirse como un proceso de diseño, creación, prueba y mantenimiento del código que permite a una computadora o dispositivo electrónico realizar tareas y procesos específicos; en este se implica la escritura de instrucciones y algoritmos que se ejecutan y llevan a una máquina a resolver problemas, satisfacer necesidades o automatizar procesos (Valencia, 2018). La

programación es una actividad que implica diseñar, escribir, probar y mantener el código que constituye el software, dicho de otro modo, la programación es el proceso que permite crear el software, es decir, el software es el producto final de la programación. Desde una perspectiva diferente, el software podría entenderse como el espacio sensible, en términos de Platón, asequible al hombre pues puede llegar a ser percibida, en tanto programa; mientras que, la programación se trata del espacio inteligible, no se percibe, al menos para el usuario final, pero es necesaria para la existencia de la sensibilidad.

2.2.2 Etapas del desarrollo histórico de la programación

Para comprender en mejor medida la noción de programación es pertinente entender su historia desde un breve recorrido por su desarrollo. La primera etapa se ubica aproximadamente desde los años 40 y hasta los 50 se reconoce como *programación manual*, se trata de un periodo donde los programadores escribían el código en lenguajes de máquina, usaban tarjetas perforadas o cintas magnéticas. Posteriormente, tuvo lugar el desarrollo de los *lenguajes de programación de alto nivel* (1950 – 1970), en el cual se hizo uso de los lenguajes de programación como Fortran, COBOL, Lisp, allí los programadores consiguieron escribir el código de forma más eficiente y legible. Una década después se consolidó la *Programación estructurada* (1970 – 1980); en esta etapa se desarrollaron técnicas de programación estructuradas como la programación modular y la Programación Orientada a Objetos, que permitieron la creación de un software más complejo y mantenible. Finalmente, desde el 80 y hasta los 2000, se ubica la *Programación Orientada a Objetos*; en esta etapa los programadores pudieron crear un software más flexible y reusable gracias a

los lenguajes de programación como C++, Java y Python. (Valencia, 2018). Otra forma de comprender los desarrollos y la clasificación de los tipos de programación puede ser a la luz del conocido *paradigma de programación*.

2.2.3 Paradigma de programación

La noción de paradigma tiene su origen en la palabra *παράδειγμα* o *parádeigma* que deviene de los vocablos *para* que se traduce como *junto* y *déigma* que se traduce como *modelo*. Usualmente esta palabra se suele usar como sinónimo de *ejemplo* o modelo. En Ingeniería del Software, este término se refiere a tareas y actividades que se agrupan en metodologías que

son comunes a todas las formas de trabajo y otras que son particulares según el enfoque determinado (...) su significado aplicado al software define el **enfoque** con que se piensa una solución a un problema que se resolverá con un programa de computación. Esta diferencia de enfoque hizo que a lo largo de los años de evolución de desarrollo del software surgieran distintos lenguajes de programación basados en los diferentes paradigmas que fueron surgiendo (Pantaleo y Rinaudo, 2015, pág. 36).

Según esto, en Ingeniería del Software la noción de *paradigma* está estrechamente ligada con la idea de *enfoque*. Cuando se habla de enfoque se refiere al punto de vista del desarrollador, donde se analiza un problema y se diseña una solución, para lo que se hace uso de los *lenguajes de programación* que

son lenguajes formales que permiten describir, definir y controlar el comportamiento de los procesadores al manipular datos. Estos lenguajes poseen una sintaxis que

determina la precisión de sus expresiones y una semántica que define el significado de cada uno de sus elementos (Pantaleo y Rinaudo, 2015, pág. 36).

En síntesis, un *paradigma* se trata del *enfoque* que un desarrollador determina para resolver un problema, para esto hace uso de un *lenguaje de programación* determinado.

2.2.3.1 Programación antes de las redes neuronales.

En busca de ofrecer una *clasificación de la programación*, es necesario, después de comprender la idea de *paradigma de programación*, reconocer que esta puede clasificarse desde antes y después de las *redes neuronales*. En esencia, el *enfoque* de programación referido a las *redes neuronales* busca imitar o recrear artificialmente el funcionamiento del cerebro humano (Contreras, 2022). Antes de ello, la programación estuvo enfocada en resolver problemas mediante un procedimiento específico, la clasificación de este periodo consiste en: la programación funcional, programación lógica, programación orientada a objetos, programación orientada a aspectos y la programación genérica (Pantaleo y Rinaudo, 2015).

De esta forma, este tipo de programación antes de las redes neuronales funciona porque el código del software se elabora para resolver un problema, en este caso lo importante es que el programador sepa cómo se resuelve el problema. El programador es quien decide qué instrucciones establecer para que la respuesta al problema se logre. Todo software necesita una entrada, también conocida como *input*, que se toma como

insumo para dar una respuesta. La operación que da respuesta, según la entrada (*input*) queda establecida por el programador en el código, lo que quiere decir que, “el programa está constituido por un conjunto de objetos relacionados entre sí, los cuales permiten hacer la entrada de datos, los cálculos necesarios para resolver el problema y dar salida a los datos convertidos en información” (López, 2013, pág., 223).

2.2.3.2 Programación después de las redes neuronales

En este tipo de programación que encontró su momento cúlmine cerca de la década del 80 con Geoffrey Hinton (Contreras, 2022). El objetivo de este tipo de programación “no es resolver problemas complejos como secuencia de pasos, sino como la evolución de un sistema computacional inspirado en el cerebro humano” (Varela & Campbells, 2011, p. 19). Este tipo de programación es fundamental para el desarrollo y el estudio de la inteligencia artificial, ya que combina herramientas como la lógica difusa, algoritmos genéticos y sistemas expertos (Varela & Campbells, 2011). En contraste con la programación antes de las redes neuronales, en este paradigma el programador no necesita saber cómo solucionar un problema, sino que entrena una red neuronal para que esta dé la solución del problema. El computador mismo tiene que encontrar la solución. El reto, en este caso, no es saber cómo dar respuesta a un *input*; sino, cómo construir una red neuronal que, al ser entrenada, pueda dar la respuesta deseada, independiente del caso. Para conseguir esto la red debe ser entrenada, por eso en primer lugar, se le entrega una gran cantidad de datos o *dataset* y estos son ordenados. Luego, se hace un análisis estadístico matemático para clasificar los datos de muchas maneras. Este análisis permite construir las operaciones estadístico-probabilísticas que van a conformar la red neuronal.

Posteriormente, se construye la red neuronal como miles, cientos de miles, millones y hasta miles de millones de neuronas y perceptrones.

En seguida de esto, se hace el entrenamiento, este consiste en parametrizar todas las neuronas, allí se calibran todas las ecuaciones de cada neurona para que pueda clasificar correctamente los fragmentos de los datos. Después se realiza la verificación de que funcione, en este paso se hacen ajustes de los parámetros que llaman *fine-tuning*. Y, finalmente, se pone a funcionar para que resuelva problemas. Con todo, la manera en que la red neuronal queda parametrizada (entrenada) para dar una salida, también conocida como *output*, no lo sabe el programador. Dicho de otro modo, la forma de clasificar la información que tiene la red neuronal es un misterio de la ingeniería. Pese a ello en este paradigma se identifican algunas ventajas, por ejemplo, el aprendizaje adaptativo, la auto-organización y la operación en tiempo real (Varela & Campbells, 2011). En cualquier caso, la programación de redes neuronales se trata de un paradigma de programación relevante en la actualidad, por su carácter permeado por el cambio constante, difícilmente podrían establecerse lineamientos claros para realizar un estudio como el que se busca en esta investigación. Por ello, este texto se centra en la *programación antes de las redes neuronales*, específicamente la dedicada al *software de desarrollo* y la *Programación Orientada a Objetos*.

2.2.4 La Programación Orientada a Objetos y el *Software de Desarrollo*

Debido a su naturaleza emergente, difícilmente podría reducirse la clasificación del software en las categorías expuestas; lo mismo sucede con la programación. Pues,

bien, dado que se trata de un dispositivo en constante evolución su clasificación es cada vez es más grande. Con la intención de entender la idea de la Ontología del Software se toma como referencia el *Software de desarrollo*, que puede entenderse como un conjunto de herramientas que incluyen los lenguajes de programación y entornos de trabajo que buscan mantener y desarrollar aplicaciones, este tipo de software puede ser el resultado de una Programación Orientada a Objetos.

La Programación Orientada a Objetos o POO es un método de programación muy parecido a la forma como hacemos las cosas. El POO está formado por una colección de objetos que interactúan conjuntamente para resolver un problema (López, 2013). Este tipo de programación es muy cercano al estudio de los objetos de la sensibilidad realizado por Platón, pues, desde la Ingeniería del Software, se define el objeto como “entes, entidades, sujetos o cosas que encontramos en el dominio de nuestra realidad, entiéndase: en situaciones o problemas de nuestro mundo cotidiano empresarial, organizacional o institucional, que se requiere manejar o solucionar mediante una computadora” (López, 2013, pág., 223). Debido a sus características, este tipo de programación es el que suele usarse para la programación de videojuegos.

Con todo, para que un usuario digital pueda disfrutar de un videojuego se necesitan de diversos elementos que anteceden y definen su aparición fenoménica en pantalla. Algunos de estos elementos, que también pueden considerarse como *pasos*, se han mencionado a lo largo de este capítulo. Por ejemplo, el *código*, se trata de la implementación de un *algoritmo* o conjunto de instrucciones en un *lenguaje de programación* para que una computadora pueda ejecutarlo. El *algoritmo*, por su parte, es un conjunto de instrucciones que se usan para resolver un problema o tarea específica.

El *lenguaje de programación* puede entenderse como un conjunto de reglas y símbolos que escribe las instrucciones que una computadora puede ejecutar.

No obstante, la computadora no puede entender el lenguaje de programación por sí sola, por tanto, el *lenguaje de programación* se interpreta o compila en *lenguaje binario*, también llamado *lenguaje de máquina*. Este lenguaje es un sistema de representación de información que usa los dígitos 0 y 1, se usa para presentar todos los tipos de datos como números, letras, símbolos y comandos. Aun así, para que se pueda visualizar correctamente los objetos que un desarrollador programa para el funcionamiento de su videojuego, se necesita de la Unidad de Procesamiento Gráfico (GPU) y el diseño de la Interfaz de Usuario.

2.2.5 La Interfaz de Usuario (UI) y la Unidad de Procesamiento Gráfico (GPU)

Una vez ejecutado el *algoritmo*, mediante el *código* y gracias al *lenguaje de programación*, se necesita que todos estos procesos se lleven a cabo de forma visual para que los usuarios digitales puedan hacer uso de ellos, para esto se necesita de la Unidad de Procesamiento Gráfico o GPU por sus siglas en inglés.

La GPU es una tarjeta gráfica que tiene más de 10.000 núcleos, que son llamados CUDA. Estos núcleos están gestionados y distribuidos por un sistema que incluye memoria de acceso rápido, un módulo de regulación de energía y un sistema de enfriamiento,

conectado por interfaces y controladores para un lograr un rendimiento máximo en aplicaciones de gráficos y cálculos paralelos (Branch education, 2024). Para modelar una figura simple como un sombrero se necesitan cerca de 28.000 triángulos constituidos por 14.000 vértices, con unas coordenadas específicas X, Y y Z. Cada uno de los objetos modelados ocupa un espacio propio.

Para conseguir que sea visualizado se toma una instrucción que suma la posición del origen del sombrero en el espacio mundial de las coordenadas X, Y y Z, correspondientes de un único vértice en el espacio modelo. Pero la transformación de vértices es sólo uno de los pasos de un proceso de *renderización* de gráficos de videojuegos (Branch education, 2024). También se requiere de un sombreado de vértices, que se trata de tomar figuras geométricas y usar el campo de visión de la cámara para calcular dónde se visualiza un objeto específico. Para esto se hace uso de los cálculos de gráfico por fotograma, del sombreado de fragmentos y del cálculo de iluminación (Branch education, 2024).

En todo caso, se necesita de un proceso adicional para que la GPU pueda mostrar su capacidad de proyección y se ejecuten efectivamente los cálculos y los movimientos realizados. Este elemento es la Interfaz de Usuario o UI, por sus siglas en inglés, se trata de:

un sistema compuesto por diferentes elementos tanto visuales como interactivos e hipermediales que inciden directamente en el proceso de comunicación usuario-sistema y puede ser analizado desde diferentes dimensiones. Por demás las características de la interfaz, sus elementos y sistema de relaciones pueden determinar la experiencia del usuario, la funcionalidad del sistema informático y el cumplimiento de los objetivos para el que está diseñado (Morejón, 2020, pág., 147).

En síntesis, la Interfaz de Usuario proporciona la información y la estructura para la *renderización* de gráficos, es decir, para la visualización en pantalla; mientras que la GPU toma esa información y la convierte en una realidad visual en la pantalla gracias a las figuras geométricas, especialmente, el triángulo.

En conclusión, este capítulo tuvo la intención de ofrecer un acercamiento a los conceptos de: software, programación, código, algoritmo, lenguaje de máquina, interfaz de usuario y GPU. Para ello se ofreció un recorrido histórico que conduce al desarrollo de estos elementos y su comprensión en la actualidad. Por este motivo, este capítulo empieza por aludir a la Ingeniería del Software, sin dejar de lado la preocupación por ofrecer al lector una definición de los conceptos mencionados. En primera instancia se habló del software, el desarrollo histórico del mismo, así como la clasificación de este. Después, se mencionó la relación que tiene con la programación, su desarrollo histórico y sus paradigmas. Este capítulo se enfocó en la Programación Orientada a Objetos y sus elementos, esto es, código, algoritmo, lenguaje de máquina, UI y GPU. Con todo, se asume que la concepción de estos elementos puede resultar confusa para el lector, especialmente si no tiene nociones en el área, por ello, en el capítulo siguiente se articularán estas nociones con las expuestas en el capítulo precedente, de esta forma se espera obtener un entendimiento más claro de la ontología del software en su dimensión estructural, a la vez que se comprende cómo el pensamiento de Platón encuentra réplicas en la actualidad.

3 Ontología del software, una lectura desde Platón

Esta investigación se propuso alcanzar una articulación conceptual entre dos disciplinas que puede parecer ajenas entre ellas, por un lado, la disciplina que se encarga del estudio, desarrollo y evolución del software, y, por otro lado, la filosofía, particularmente, la platónica. El primer lugar de esta investigación estuvo dedicado al pensamiento del autor griego, y el segundo, a los elementos que llevan a la comprensión del software en su dimensión estructural. En este capítulo se encontrará una recapitulación conceptual en función de desarrollar en esta sección los elementos más importantes de las líneas precedentes y ofrecer una articulación más precisa.

3.1 La ontología de Platón, conceptos y nociones

En referencia a lo expuesto en la primera sección de esta investigación se empezó por hacer la distinción entre *eidos* y *devenir* como los dos extremos de la ya conocida línea ontoepistemológica. El primero de estos conceptos estuvo referido a la noción de *episteme*, es decir, se atribuye al conocimiento de las formas; mientras que el segundo se refiere a la noción de *doxa*, esto es, está relegado al conocimiento de la sensibilidad. Con la intención de comprender de manera más acertada estos estadios, se presentaron las nociones de:

Eikasia: también conocida como *conjetura*, se trata del primer lugar de la línea ontoepistemológica, esta noción se ubica en el terreno de la *doxa* y, por consiguiente, del *devenir*. Su descripción se presenta, en referencia a la imagen de la caverna, como el estado primigenio de los prisioneros en el cual se perciben, principalmente, sombras. Se basa únicamente en lo que se ve, sin un conocimiento de la realidad subyacente. En otras palabras,

se refiere al modo de conocimiento más básico y bajo, se basa enteramente en la percepción sensorial y la opinión; no en la razón o intuición.

Pistis: también conocida como *creencia*, se refiere al segundo lugar de la línea, acompaña a la *eikasia* en el terreno de la *doxa*. No obstante, es jerárquicamente superior, ya que, en alusión a la caverna, en este estado los prisioneros logran visualizar los cuerpos sensibles como tal, es decir, se abandona la visión de las sombras y se vislumbran los cuerpos de la sensibilidad. En otras palabras, se tiene una noción más profunda de los objetos visibles. Desde este concepto, entendemos que se trata de un nivel intermedio de conocimiento, se ubica entre la *eikasia* y la *dianoia*, aunque se trate de un estado más elevado, es asequible al hombre por medio de las imágenes, con todo, estas se emplean como creencias.

Dianoia: también conocida como *pensamiento discursivo*, se trata del primer estado correspondiente al segmento de la *episteme* y el tercero en la totalidad de la línea. En este los prisioneros de la caverna, pese a que se hace uso de las imágenes, parten de estas en búsqueda de los principios subyacentes, para lo cual también hacen uso de la matemática y la geometría. Es una forma de razonamiento estructurado que permite avanzar hacia la comprensión de principios abstractos, pero aún no llega a la intuición directa de las ideas, que Platón llama *noesis*. Con ello, se comprende que, en este estado se empieza a hacer uso de la razón, esta noción está más cercana a la *noesis*, lo que conlleva un alejamiento de los dos primeros estadios.

Noesis: también llamada *inteligencia*, se trata del estado superior de la *episteme*. En este estado el hombre tiene conocimiento pleno y consciente del mundo, es un estado de abstracción total, en el que ya no se parte de las imágenes sensoriales y las representaciones

materiales; sino de una comprensión directa y pura de las ideas. En ese estado, asegura el filósofo ateniense, el hombre está tras la búsqueda de las ideas mismas.

Es pertinente recordar que, para Platón, todos estos estados se unen mediante el uso de la matemática y geometría, es decir, en términos ontológicos es posible visualizar los objetos de la sensibilidad gracias a la articulación geométrico-matemática entregada por el demiurgo, específicamente, gracias a la unión y transformación de los poliedros regulares. Y, epistemológicamente hablando, se puede ascender en la búsqueda del conocimiento entre estos estados, solo en la medida que se tenga conocimientos en matemática y geometría. De esta forma el esquema antes retratado, enfocado en los conceptos de Platón es el siguiente:

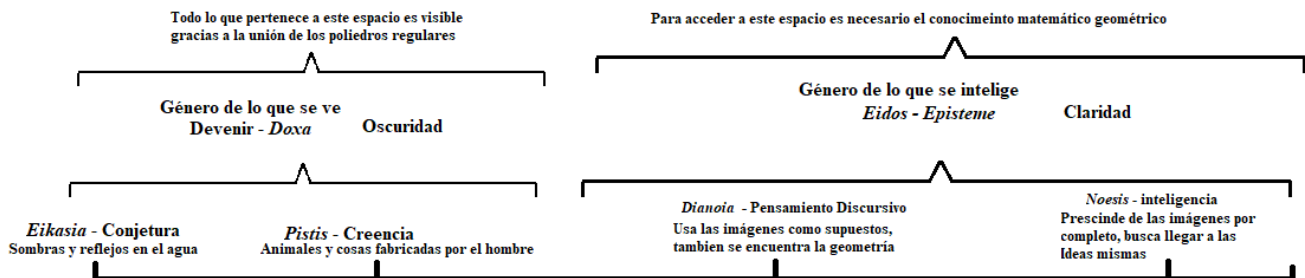


Ilustración 3. Tercer esquema de la Línea Ontoepistemológica

Esta ilustración no difiere mucho de la antes presentada, en esta se intenta retratar las nociones más importantes de la ontología del pensador ateniense, por un lado, se ubican los dos grandes elementos que dan lugar un desarrollo conceptual: *devenir* y *eidos*. A la vez que se aluden a sus equivalentes: *doxa* – *episteme*. Con esto da lugar la mención de los estados intermedios: *eikasia* – *pistis*, por el lado de la *doxa* y, *dianoia* – *noesis*, por el lado de la *episteme*. En esta ilustración se presenta el lugar que ocupa la matemática y geometría en cada estado. Por el lado de la *doxa* – *devenir*, la matemática y la geometría permiten que los

objetos de la sensibilidad sean visibles gracias a la configuración poliédrica instaurada por el demiurgo. Por el lado del *eidos – episteme*, el lugar de la matemática y la geometría se dedica al terreno epistemológico, esto es, garantiza la ascensión a los conceptos abstractos por la aplicación de la matemática y de la geometría.

3.2 El software, nociones y conceptos

Por su parte, en el segundo capítulo se mostraron las nociones más importantes y representativas del software. En primera instancia se examinaron las dimensiones en las cuales podría definirse y entenderse el software, se tomó como referencia, para esta investigación, la *dimensión estructural*, por tratarse de la estructura abstracta de todo software o programa, esto es, esencialmente, lo referido al funcionamiento del *código* que implica los elementos:

Algoritmo: conjunto de instrucciones paso a paso, bien definidas que buscan resolver un problema específico o realizar una tarea determinada; es la base de la programación. Se puede entender como una estructura organizada con el fin de resolver o dar respuesta a un problema. En el algoritmo se diseña y se visualiza el proceso, paso a paso, cuya culminación o resultado responde o da respuesta al *Input* o entrada inicial. Este elemento resulta imperceptible para el usuario digital, es decir, por ejemplo, el jugador del videojuego interactúa con la interfaz gráfica, pero no tiene acceso al conjunto de instrucciones lógicas que determinan el comportamiento del juego o algoritmo, sin embargo, este es necesario e imprescindible para la programación del mismo.

Entendíamos que el *Lenguaje de maquina* equivale a un sistema de representación que usa los dígitos 0 y 1, pero que sólo es entendido por la computadora. Es el proceso específico que llevan a cabo los circuitos y se puede representar como un flujo de ceros y unos, aunque también en otras representaciones matemáticas.

Código de programación: es un conjunto de instrucciones escritas en un lenguaje de programación específico que indican a una computadora qué acciones debe realizar en una tarea específica. El lenguaje de programación se concreta en un código (programa) que efectúa o realiza para ese lenguaje el diseño planteado por y en el algoritmo. Este código es un poco más cercano al lenguaje ordinario, pues, usa palabras que pueden ser comprendidas fácilmente por una persona que no sea programador.

Finalmente, el segundo capítulo cerró con señalar la función que tiene la GPU y la UI para la proyección de las imágenes y la experiencia del usuario en pantalla. Sin estas herramientas, que funcionan especialmente con un orden geométrico y matemático definido, no se podrían visualizar las figuras y formas en la pantalla. La UI es presentada como el punto de interacción entre un sistema informático y un usuario, que proporciona una experiencia de usuario por lo general intuitiva y eficiente, se presenta de una manera clara y sencilla. La *GPU* o Graphics Processing Unit, se presentó como un componente electrónico especializado en el procesamiento de gráficos y cálculos matemáticos intensivos que renderiza las imágenes y videos por medio de técnicas de gráficos 3d, presentados por figuras geométricas, particularmente, triángulos.

3.3 La dimensión estructural del software a la luz de la ontología de Platón

En analogía con la *dimensión estructural* del software, podría decirse que, por sus características, la noción *eikasia* concuerda con la imagen proyectada en la pantalla como resultado de una programación de POO, pues el usuario solo tiene acceso a una representación superficial del programa que aparece en pantalla, aunque si bien podría acceder directamente a su estructura más profunda, esta está limitada y requiere de un ejercicio epistemológico. Es decir, podría entenderse que la *eikasia* se refiere a la imagen del software que se presenta como resultado en pantalla, esta imagen que se visualiza tiene tras de sí una serie de pasos técnicos complejos que fueron necesarios para obtener este resultado, pasos que, en analogía con la línea ontoepistemológica, se refieren a los otros segmentos mencionados, *pistis*, *dianoia* y *noesis*.

Mientras que, la noción de *pistis* tiene un paralelo con el *código de programación*, pues este, pese a que es estructurado y lógico, sigue siendo una forma simbólica de una representación que se piensa desde el algoritmo. En otras palabras, el código permite que el programador interactúe con el sistema más allá de la imagen en pantalla, de la misma forma como la *pistis* permite al prisionero desprenderse de las sombras y visualizar los objetos de la sensibilidad directamente. Se trata de un conocimiento jerárquicamente más abstracto con respecto a la *eikasia*, o fenómeno en pantalla, pero comprensible, pues, el código de programación emplea palabras o expresiones cercanas al lenguaje ordinario como “hello world”.

Por lo que respecta al concepto de *dianoia*, se propone que esta noción es cercana al *lenguaje de máquina*. Al igual que la *dianoia* el *lenguaje de máquina* permite una comprensión más precisa de cómo funciona el sistema, en comparación con el *código* y la imagen en pantalla, pero sigue siendo un estado intermedio que depende de símbolos

binarios de 0 y 1 para la comprensión del *código de programación* por parte del dispositivo. Es decir, el *lenguaje de máquina* está determinado por el uso de símbolos matemáticos, de la misma como para estar en el estado de *dianoia* el prisionero debe conocer y emplear la matemática y geometría. Se trata de un lenguaje menos asequible al usuario digital, pues, en síntesis, este solo es entendido por la máquina, resulta, entonces, más abstracto que el código.

La noción cúspide del recorrido ofrecido por el pensador griego es la *noesis* que, analógicamente, es semejante al *algoritmo*, pues este representa la estructura lógica y abstracta que guía el comportamiento del software, de la misma forma que la *noesis* guía el conocimiento puro de las ideas. El algoritmo es el fundamento que define cómo debe funcionar el programa, va más allá de las representaciones visuales que el usuario experimenta en la interfaz. En el estado de *noesis* el prisionero también va más allá de las representaciones visuales ofrecidas en la caverna. Además, el acceso a la *noesis* y al *algoritmo* de programación es bastante limitado, solo se accede a este después de un proceso epistemológico aplicado.

Se puede afirmar que, de la misma forma, como en la caverna se proyectan las sombras (*eikasia*) como resultado de una figura que se muestra por encima del fuego con objetos reales (*pistis*), y esta a su vez, por encima de los objetos matemáticos geométricos (*dianoia*), y estos, a su vez, por encima del sol (*noesis*). En el software, el fenómeno que se percibe en pantalla (*eikasia*) es el resultado de un *código de programación* (*pistis*), este está escrito en lenguaje cercano a las expresiones ordinarias. Pero no puede ser entendido por la máquina directamente, entonces, es traducido por la máquina en un *lenguaje de máquina* (*dianoia*), que resulta más inasequible para el usuario ordinario, pues es comprendido por

la máquina, y este es el resultado de la aplicación de un *algoritmo* comprendido especialmente por el desarrollador y más lejano para el usuario digital (*noesis*). También, de la misma forma como todas las articulaciones conceptuales en el pensamiento de Platón se comprenden y son asequibles al hombre gracias a la geometría y a la matemática, particularmente, por la unión de los poliedros regulares. En el software es posible visualizar para el usuario digital las figuras gracias a la articulación de la *Interfaz de usuario* y la GPU que funcionan, esencialmente, como la unión y transformación de cientos de triángulos en unas coordenadas específicas indicadas desde unos cálculos matemáticos determinados.

De esta forma, con lo dicho hasta ahora, la línea que ha sido graficada desde el principio de esta investigación, y, en analogía con lo expuesto en este acápite, puede ser representada de la siguiente manera

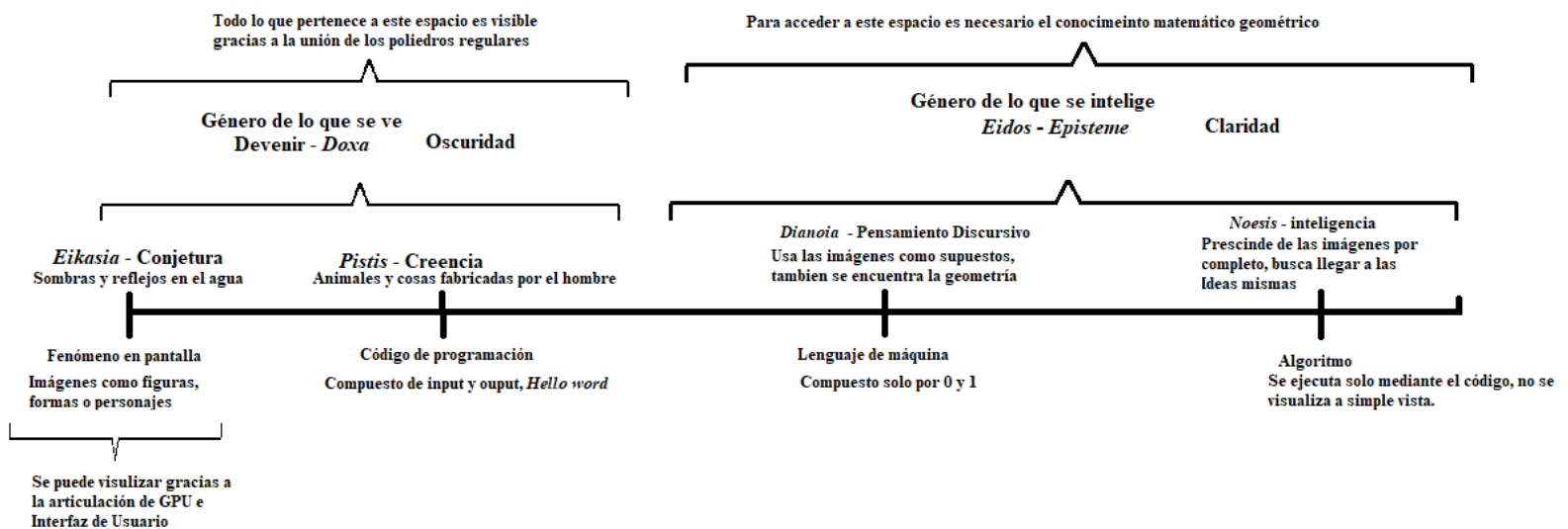


Ilustración 4. Esquema final de la Línea Ontoepistemológica en relación con la Ontología del Software

En esta ilustración se ubican las correspondencias antes mencionadas a saber: la *eikasia* equivale a los fenómenos percibidos en pantalla como resultado de una programación de POO, por ejemplo, figuras o personajes, en el caso de la *pistis*, se reconoce su semejanza con el código de programación, que usa expresiones cercanas al lenguaje ordinario, pero sigue siendo más abstracto que el fenómeno que se percibe en pantalla. Por lo que corresponde a la *dianoia*, se encuentra su semejanza con el lenguaje de máquina, es aún más ajeno al fenómeno en pantalla, pero no ocupa el lugar de la comprensión completa del proceso de programación, y, en lo que respecta a la *noesis*, encuentra su equivalente con el algoritmo, se trata un conocimiento completo y complejo del proceso de programación, más abstracto y de un acceso más limitado para el usuario digital. Finalmente, se ubica el lugar de la UI y la GPU en el primer segmento de la línea por indicar el uso de estos elementos en la percepción indirecta en pantalla, lo que quiere decir que, sin la intervención de UI y la GPU, la secuencia técnica de programación no podría ser visualizada por el usuario digital. La parte superior de la imagen se refiere, como se ha mencionado previamente, al esquema ontológico de Platón. Mientras que la parte inferior, en su conjunto, se refiere al software, resultado de una Programación Orientada a Objetos, por ello se hace referencia a los elementos, también comprendidos como pasos, para comprender su contenido total, a la luz de su dimensión estructural, es decir, su dimensión más abstracta orientada, principalmente, al código y su funcionamiento.

Ahora bien, el objetivo de esta investigación fue entender la Ontología del Software a la luz del pensamiento de Platón, sin ánimo de afirmar que sus elementos, nociones y pasos son exactamente los mismos que los realizados por el pensador griego en su proyecto filosófico. En este sentido, se identifican algunas diferencias precisas:

I. En primer lugar, secuencialmente hablando, el *lenguaje de máquina*, compuesto por ceros y unos, no es posterior al *código de programación*, tal como se retrata en la línea, pues, si se refiere al proceso, el algoritmo es la primera noción que desarrolla el programador, posteriormente, esta se ejecuta por medio de *código*, y el lenguaje de máquina es un elemento subyacente que ya está presente en el dispositivo que usa el programador para ejecutar el código, lo que quiere decir que el programador, por lo general, no interviene con el *lenguaje de máquina*, ya que los dispositivos electrónicos como la PC lo poseen.

Frente a esta diferencia se reconoce que, si bien, el lugar del *código de programación* es otro, la intención de este artículo, como se mencionó, es entender la Ontología del Software, esto es, sus elementos y composición a la luz de la Ontología de Platón, y para ello se usan las nociones propias del pensamiento del fundador de la Academia y, en esa correspondencia analógica, las características del *lenguaje de máquina* resultan más cercanas a la noción de *dianoia*, por tratarse del uso de expresiones matemáticas y por ser, jerárquicamente, de un acceso más limitado, de la misma forma que la *dianoia*. Además, el código de programación está compuesto de comandos que usan expresiones más cercanas al lenguaje ordinario, por ejemplo “Hello, World”, lo que quiere decir que es más comprensible para el usuario común, por este motivo está ubicado en el segundo paso de la línea o *pistis*.

II. En segundo lugar, para Platón la matemática y la proporción geométrica no sólo es importante en términos ontológicos, pues esta define y permite que los fenómenos sean perceptibles para el hombre; sino también, la comprensión de

esta ciencia garantiza que el hombre pueda ascender y superar los estados de la línea, es decir, sin una comprensión de la matemática y la geometría, el hombre no podría acceder al conocimiento de la *idea*. El equivalente matemático-geométrico de Platón, es, para este trabajo, la articulación entre la Interfaz de Usuario y la GPU. En este sentido, esta analogía tiene al menos dos implicaciones: por un lado, de la misma forma como la descripción de Platón del cosmos; en el software es necesaria la implementación de la geometría para la aparición de las formas en la pantalla, eso se garantiza por medio de la UI y la GPU. Pero, por otro lado, el programador podría prescindir, si lo desea, de la articulación de la UI y GPU crear un algoritmo. Dicho de otro modo, para Platón no es posible acceder a la *noesis* sin un conocimiento matemático-geométrico; el programador, por su parte, puede crear y formular un *algoritmo* (el equivalente de la *noesis*) prescindiendo de la IU y la GPU, pues estas herramientas se usan, especialmente, para verificar si el *algoritmo* y el *código de programación*, funcionan como el programador desea.

Conclusiones

La humanidad siempre se adapta a su entorno, que suele estar permeado y definido por las herramientas que se desarrollan. Desde las últimas décadas, las herramientas digitales han saturado la forma de realidad humana. Todas estas herramientas difieren entre sí, unas están enfocadas en la interconexión de los usuarios, otras en la economía, entretenimiento, etc. No obstante, todas ellas tienen algo en común, a saber, el software, comprendido como

un dispositivo tecnológico subyacente que garantiza el funcionamiento de programas. De ahí que, debido a su ubicuidad, esta noción haya sido el motivo de reflexión principal de esta investigación.

Para entender el software se planteó la idea de que este es un tipo dispositivo tecnológico que puede estudiarse desde la ontología y, además, se sostiene que una forma de entender su ontología puede ser bajo el estudio de su dimensión estructural, y, según esto, se sostiene que, la Ontología del Software, vista desde su dimensión estructural puede entenderse a la luz de la Ontología de Platón.

Para conseguir sostener esto, esta investigación empezó por examinar las nociones elementales de la Ontología del pensador griego, se determinó la relación entre el *devenir*, la *doxa*, la *aisthesis* y la *episteme*; a la vez que se mencionaron e ilustraron, por medio de una línea los estados intermedios de realidad y conocimiento llamados *eikasia*, *pistis*, *dianoia* y *noesis*. Finalmente, el primer capítulo de esta tesis señaló el lugar que ocupa la matemática y geometría en el proyecto ontológico del autor.

Posteriormente, el segundo capítulo de esta exploración busco entender el software como un dispositivo, y se enunció el motivo de tomar su estudio ontológico desde su dimensión estructural, después, se mostró la relación que tiene el software con la programación a la vez que se señalaron diversos paradigmas de programación y se hacía énfasis en el Software de Desarrollo y la Programación Orientada a Objetos, para entender en mejor medida estas nociones se exploraron los elementos o pasos fundamentales para llevar a cabo el acto de programar, por eso se definieron los elementos de: algoritmo, código lenguaje de máquina, Interfaz de Usuario y GPU. Para finalizar este estudio, el tercer capítulo estuvo dedicado en señalar cómo pueden entenderse las nociones básicas del software desde

las nociones elementales de la Ontología de Platón, esto consistió en mostrar las equivalencias de conceptos en ambos terrenos, además, se señalaron los puntos de distancia entre ambas ideas.

Con todo, este trabajo buscó comprender los elementos básicos del software con referencia a su dimensión estructural a la luz de la Ontología de Platón, pero este proyecto está lejos de terminar. Debido a su naturaleza emergente la renovación de los paradigmas de programación es cada vez más extensa, el diálogo que se puede establecer entre estos fenómenos y la filosofía clásica puede resultar sumamente valioso. Por ahora, puede sostenerse que, al menos por lo que respecta al filósofo fundador de la academia, sus contribuciones al estudio ontológico permiten ser estudiadas en diversos tipos de realidad, incluso, en la propia a la Ontología del Software.

Referencias bibliográficas

Aristóteles. (1994). *Metafísica*. (Calvo, M. T. trad.). Gredos.

Branch Education. (2024). *How do Graphics Cards Work? Exploring GPU Architecture*. [video].

YouTube.

https://www.youtube.com/watch?v=h9Z4oGN89MU&t=685s&ab_channel=BranchEducation.

Cornford, F. (2007). *La teoría platónica del conocimiento*. Paidós.

Contreras, L.C. (2022, 2 de abril). *Inteligencia Artificial - Clase 07: Redes Neuronales*. [video].

YouTube.

https://www.youtube.com/watch?v=ejrbMbz-F4g&t=280s&ab_channel=LuisCarlosContreras.

Cristiá, M. (2024). *Una teoría para el Diseño de Software*. Ingeniería del Software 2.
<https://www.fceia.unr.edu.ar/ingsoft/intro-diseno.pdf>

Denyer, N. (2007). Sun and Line: The Role of the Good. *The Cambridge Companion to Plato's Republic (Cambridge Companions to Philosophy)*. 284-309.
 Doi:10.1017/CCOL0521839637.011.

Gerena, L. (2009). La descripción platónica de la percepción Teeteto, 184 – 186. *Ideas y valores*, 139, 87 – 107. <https://revistas.unal.edu.co/index.php/idval/article/view/36029/38260>

Gómez, R. A (1993). *Platón: los seis grandes temas de su filosofía*. Fondo de cultura económica.

Lanza, G. H. (2015). Matemática y física en el Timeo de Platón. Poliedros regulares y elementos naturales. *Praxis filosófica*. 40, 85 – 112.
<https://www.redalyc.org/pdf/2090/209038528004.pdf>

López, R. L. (2013). *Metodología de la Programación Orientada a Objetos*. Alfaomega Grupo Editor.

Maldonado Serrano, J. F., Rodríguez Ramírez, D. A., Caceres, P. B., & Petit Suárez, J. F. (2020). An Ontology of Software: Series, Structure and Function. *Praxis Filosófica*, 51, 115-132.
<https://doi.org/10.25100/pfilosofica.v0i51.10114>

Migliori, M. (2009). El esquema ontoepistemológico de la Línea. *Areté*, XXI, 197 – 219.
https://www.researchgate.net/publication/262589585_El_esquema_ontoepistemologico_de_la_Linea/fulltext/57bca2a608aedf5f75eaa222/El-esquema-ontoepistemologico-de-la-Linea.pdf.

- Morejón, L. S. (2020). Principios del proceso de Diseño de Interfaz de Usuario. *Revista Cubana de Transformación Digital*, 3, 143 – 155. <https://rctd.uic.cu/rctd/article/view/96/33>.
- Palacios, L. M. (2020, 25 de agosto). *Clasificación de Software*. [video]. YouTube. https://www.youtube.com/watch?v=3fBuF6VBTk&t=52s&ab_channel=LuisM.Palacios.
- Pantaleo, G. (2012). *Calidad en el Desarrollo del Software*. Marcombo.
- Pantaleo, G. & Rinaudo, L. (2015). *Ingeniería del Software*. Alfaomega Grupo Editor.
- Pérez, H. F. M. (2007). La eliminación de la subjetividad de los fines. Platón y las matemáticas. *Eikasia. Revista de Filosofía*, 12, 219 – 252. <https://rua.ua.es/dspace/bitstream/10045/12471/1/Plat%C3%B3n%20matem%C3%A1tica.s.pdf>.
- Platón. (1988). *Diálogos III: Fedón, Banquete, Fedro*. (C. García Gual, M. Martínez Hernández, E. Lledo Iñigo, trad.). Gredos.
- Platón. (1987). *Diálogos II: Gorgias, Menéxeno, Eutidemo, Menón, Crátilo*. (J. Calonge Ruiz, E. Acosta Méndez, F. J. Olivieri, J. L. Calvo, trad.) Gredos.
- Platón. (1992). *Diálogos V. Parménides, Teeteto, Sofista, Político*. (M. I. Santa Cruz, A. Vallejo Campos, N.L. Cordero, trad.). Gredos.
- Platón. (1988). *Diálogos IV: República*. (L. Conrado Eggers, trad.). Gredos.
- Platón. (2010). *Timeo*. (J.M, Zamora Calvo, trad.). Abada.
- Santa, C. M. I. (1986). Eikós Lógos y diánoia en Platón. *Revista de filosofía y teoría política*, 26 – 27, 180 – 184. https://www.memoria.fahce.unlp.edu.ar/art_revistas/pr.1290/pr.1290.pdf

Tylor, A. E. (2005). *Platón*. Tecnos.

Valencia, C. (2018, 10 de marzo). *Historia y evolución de los lenguajes de programación*. [video].

YouTube.

https://www.youtube.com/watch?v=TrzZ7YQyXbs&t=65s&ab_channel=CamiloValencia.

Varela, E.A & Campbells, E.S. (2011). Redes Neuronales Artificiales: Una Revisión del Estado

del Arte, Aplicaciones Y Tendencias Futuras. *Investigación y desarrollo en TIC*, 2,

18

–

27.

<https://revistas.unisimon.edu.co/index.php/identific/article/download/2455/2348>.