Universidad Nacional Mayor de San Marcos

Universidad del Perú. Decana de América

Dirección General de Estudios de Posgrado

Facultad de Matemática

Unidad de Posgrado

Desarrollo de software para la optimización de funciones polinómicas de una variable aplicando criterio de la derivada enésima en el cálculo de máximos y mínimos

TESIS

Para optar el Grado Académico de Magíster en Computación e Informática

AUTOR

Br. Jaime Alfonso FERNANDEZ CAYCHO

ASESOR

Mg. Percy Edwin DE LA CRUZ VÉLEZ DE VILLA

Lima, Perú

2021

Referencia bibliográfica

Fernandez, J. (2021). *Desarrollo de software para la optimización de funciones polinómicas de una variable aplicando criterio de la derivada enésima en el cálculo de máximos y mínimos*. [Tesis de maestría, Universidad Nacional Mayor de San Marcos, Facultad de Matemática, Unidad de Posgrado]. Repositorio institucional Cybertesis UNMSM.

Metadatos complementarios: autor / asesor

Datos de autor	
Nombres y apellidos	Jaime Alfonso Fernandez Caycho
Tipo de documento de identidad	DNI
Número de documento de identidad	10025762
URL de ORCID	No aplica
Datos de asesor	
Nombres y apellidos	Percy Edwin De la Cruz Vélez de Villa
Tipo de documento de identidad	DNI
Número de documento de identidad	08583141
URL de ORCID	https://orcid.org/0000-0002-4943-7620
Datos del jurado	
Presidente del jurado	
Nombres y apellidos	Daniel Alfonso Quinto Pazce
Tipo de documento de identidad	DNI
Número de documento de identidad	10372726
Miembro del jurado 1	
Nombres y apellidos	José Carlos Oré Lujan
Tipo de documento de identidad	DNI
Número de documento de identidad	
Miembro del jurado 2	
Nombres y apellidos	Ronal Santos Paredes Vargas
Tipo de documento de identidad	DNI
Número de documento de identidad	09565844
Datos de investigación	
Línea de investigación	A.3.1.4. Matemática aplicada (Biomatemática, Economía, Optimización y otros)

Grupo de investigación	No aplica
Agencia de financiamiento	No aplica
Ubicación geográfica de la investigación	País: Perú Departamento: Lima Provincia: Lima Distrito: San Juan de Miraflores Latitud: -12.16979 Longitud: -76.96222
Año o rango de años en que se realizó la investigación	2015 - 2020
URL de disciplinas OCDE	Ciencias de la computación https://purl.org/pe-repo/ocde/ford#1.02.01

UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS
(Universidad del Perú, DECANA DE AMÉRICA)

FACULTAD DE CIENCIAS MATEMÁTICAS
VICEDECANATO DE INVESTIGACIÓN Y POSGRADO
UNIDAD DE POSGRADO

ACTA DE SUSTENTACIÓN VIRTUAL DE TESIS PARA OPTAR EL GRADO ACADÉMICO DE MAGÍSTER EN COMPUTACIÓN E INFORMÁTICA

Siendo las **18:55** horas del día viernes veinticuatro de setiembre de dos mil veintiuno, en la sala virtual meet.google.com/icg-tocy-rhv, el Jurado de Tesis conformado por los siguientes docentes:

PRESIDENTE	:	Mg. Daniel Alfonso Quinto Pazce
MIEMBRO :	.	Mg. José Carlos Oré Luján
MIEMBRO EXTERNO	:	Dr. Ronal Santos Paredes Vargas
MIEMBRO ASESOR	:	Mg. Percy Edwin De la Cruz Vélez de Villa

se reunieron para la sustentación de la tesis titulada: «DESARROLLO DE SOFTWARE PARA LA OPTIMIZACIÓN DE FUNCIONES POLINÓMICAS DE UNA VARIABLE APLICANDO CRITERIO DE LA DERIVADA ENÉSIMA EN EL CÁLCULO DE MÁXIMOS Y MÍNIMOS» presentada por el Bachiller Jaime Alfonso Fernández Caycho, para optar el Grado Académico de Magíster en Computación e Informática.

Concluida la exposición, los miembros del Jurado de Tesis procedieron a formular sus preguntas que fueron absueltas por el graduando; acto seguido se procedió a la evaluación correspondiente, según tabla adjunta, habiendo obtenido el Bachiller Jaime Alfonso Fernández Caycho, el calificativo de **B MUY BUENO (17)**

Habiendo sido aprobada la sustentación de la Tesis, el Jurado Evaluador recomienda para que el Consejo de Facultad apruebe el otorgamiento del **Grado Académico de Magíster en Computación e Informática al Bachiller Jaime Alfonso Fernández Caycho.**

Siendo las **20:05** horas, se levantó la sesión, firmando para constancia la presente Acta:

Mg. Daniel Alfonso Quinto Pazce
PRESIDENTE

Mg. José Carlos Oré Luján
MIEMBRO

Dr. Ronal Santos Paredes Vargas
MIEMBRO EXTERNO

Mg. Percy Edwin De la Cruz Vélez de Villa
MIEMBRO ASESOR

FICHA CATALOGRÁFICA

DESARROLLO DE SOFTWARE PARA LA OPTIMIZACIÓN DE FUNCIONES POLINÓMICAS DE UNA VARIABLE APLICANDO CRITERIO DE LA DERIVADA ENÉSIMA EN EL CÁLCULO DE MÁXIMOS Y MÍNIMOS

AUTOR: FERNANDEZ CAYCHO, JAIME ALFONSO

ASESOR: Mg. DE LA CRUZ VÉLEZ DE VILLA, PERCY EDWIN

LIMA - PERÚ, 2020

Grado Académico: Magister en Computación en Informática

Área/ Programa / Línea de Investigación
Ingenierias / Tecnologia de Informacion y Comunicación / Procesamiento de Datos

Posgrado: Universidad Nacional Mayor de San Marcos – Facultad de Ciencias Matemáticas – Unidad de Posgrado

Formato 28 x 20, Páginas XIX, 214

DEDICATORIA

A mis padres, Hilda y Luis por todo su amor y formación de vida.
A mi esposa e hijos por su comprensión por las horas que dejamos de compartir.

AGRADECIMIENTOS

A la memoria del Dr. José Tola Pasquel, a quien no tuve la oportunidad de conocer, pero cuya labor académica ha contribuido al desarrollo matemático del país e inspira el desarrollo de la presente investigación.

Al Dr. Pedro Celso Contreras Chamorro por sus valiosos aportes que han contribuido al desarrollo de la investigación, y en particular a una lección particular del dominio de las matemáticas.

Al Mg. Percy Edwin De la Cruz Vélez de Villa, asesor de la investigación, por su compromiso personal y sus consejos que me impulsaron a concluir la investigación.

Al estudiante de Ingeniería Informática Isaac Paliza Delgado por su apoyo incondicional en el desarrollo del software.

UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS
FACULTAD DE CIENCIAS MATEMÁTICAS
UNIDAD DE POSGRADO

DESARROLLO DE SOFTWARE PARA LA OPTIMIZACIÓN DE FUNCIONES POLINÓMICAS DE UNA VARIABLE APLICANDO EL CRITERIO DE LA DERIVADA ENÉSIMA EN EL CÁLCULO DE MÁXIMOS Y MÍNIMOS

Titulando : Fernandez Caycho, Jaime Alfonso
Asesor : De la Cruz Velez de Villa, Percy Edwin
Titulo : Desarrollo de software para la optimización de funciones polinómicas de una variable aplicando el criterio de la derivada enésima en el cálculo de máximos y mínimos
Fecha : Febrero 2021

RESUMEN

Un aporte de la derivada corresponde al cálculo de extremos relativos y puntos de inflexión al utilizar la primera y segunda derivada, sin embargo, existen situaciones donde los criterios no son concluyentes lo que se ha advertido en libros de autores, asi como al usar software en línea.
La investigación “Desarrollo de software para la optimización de funciones polinómicas de una variable aplicando el criterio de la derivada enésima en el cálculo de máximos y mínimos”, tiene como propósito el desarrollo de software de aplicación para determinar los máximos y mínimos relativos, así como los puntos de inflexión de una función polinómica aplicando el criterio de la derivada enésima presentando una alternativa diferente a los procedimientos conocidos.
El software DERIN se desarrolló con el lenguaje de programación Java, que luego fue sometido a prueba con una muestra aleatoria de funciones testigo seleccionadas de libros asi como en forma aleatoria de acuerdo con el grado del polinomio y comparadas con las soluciones proporcionadas por otros productos de software. Las pruebas permiten afirmar la confiabilidad del software desarrollado donde se obtiene igual o mejor resultado que los otros software en lo referente a la aproximacion de valores y la determinación de extremos relativos.

Palabras claves: aproximación de valores, software de aplicación, desarrollo de software, criterio de la derivada enésima, extremos relativos, métodos numéricos, optimización.

UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS
FACULTAD DE CIENCIAS MATEMÁTICAS
UNIDAD DE POSGRADO

DESARROLLO DE SOFTWARE PARA LA OPTIMIZACIÓN DE FUNCIONES POLINÓMICAS DE UNA VARIABLE APLICANDO EL CRITERIO DE LA DERIVADA ENÉSIMA EN EL CÁLCULO DE MÁXIMOS Y MÍNIMOS

Titulando : Fernandez Caycho, Jaime Alfonso
Asesor : De la Cruz Velez de Villa, Percy Edwin
Titulo : Desarrollo de software para la optimización de funciones polinómicas de una variable aplicando el criterio de la derivada enésima en el cálculo de máximos y mínimos
Fecha : Febrero 2021

ABSTRACT

A contribution of the derivative corresponds to the calculation of relative extremes and inflection points when using the first and second derivative, however, there are situations where the criteria are inconclusive what has been noted in author's books, as well as when using online software.
The research “Software development for the optimization of polynomial functions of a variable applying the criterion of the nth derivative in the calculation of maximums and minimums”, has the purpose of developing application software to determine the relative maximums and minimums, as well as the inflection points of a polynomial function applying the criterion of the nth derivative presenting a different alternative to the known procedures.
DERIN software was developed with the Java programming language, which was then tested with a random sample of selected control functions from books as well as randomly according to the degree of the polynomial and compared to the solutions provided by other products of software. The tests allow to affirm the reliability of the software developed where the same or better result is obtained than the other software in relation to the approximation of values and the determination of relative extremes.

Keywords: approximation of values, application software, software development, criterion of the umpteenth derivative, relative extremes, numerical methods, optimization.

ÍNDICE

CAPÍTULO IV: METODOLOGÍA Y RESULTADOS

CAPÍTULO V: CONCLUSIONES RECOMENDACIONES Y TRABAJOS FUTUROS

LISTA DE CUADROS

LISTA DE FIGURAS

INTRODUCCIÓN

La investigación “Desarrollo de software para la optimización de funciones polinómicas de una variable aplicando el criterio de la derivada enésima en el cálculo de máximos y mínimos” tiene como propósito el desarrollo de software que proporcione los extremos locales y las inflexiones que presenta una función polinómica, utilizando un criterio generalizado: la derivada enésima.

Las referencias bibliográficas provienen de libros de Cálculo Superior donde destaca el texto Análisis Real I del Dr. José Tola Pasquel, así como autores nacionales y extranjeros que han contribuido a dar la consistencia teórica y algorítmica del software desarrollado

La investigación se desarrolla en los siguientes capítulos:

El capítulo I describe el planteamiento de la investigación, la situación problemática, así como los objetivos, hipótesis, justificación, limitaciones y alcances de esta.

El capítulo II refiere el marco teórico considerando las bases conceptuales y la fundamentación matemática sobre el criterio de la derivada enésima que delinea la investigación.

El capítulo III expone el estado del arte considerando trabajos nacionales como extranjeros que aportan en forma directa o indirecta al desarrollo de la investigación, así como el estado del desarrollo de software referente con la investigación donde se presenta software de aplicación como Symbolab, GeoGebra, Wolframalpha.

El capítulo IV explica la metodología de la investigación, determinando el tipo, nivel, así como la población y muestra. Asimismo, se detalla el proceso de construcción y características del software y las pruebas referentes para el analisis de loss resultados

El capítulo V, aborda lo referente a las conclusiones de la investigación, así como las recomendaciones y bibliografía utilizada.

En síntesis el software DERIN es un software orientado a web a disposición como un recurso de búsqueda de extremos relativos y de puntos de inflexión para aproximación de funciones y como punto de partida para futuras investigaciones

CAPÍTULO I

PLANTEAMIENTO DEL PROBLEMA

CAPÍTULO I

PLANTEAMIENTO DEL PROBLEMA

1.1 Situación Problemática

La enseñanza de la matemática en el nivel superior requiere un análisis profundo en cuanto a la forma de cómo se enseña y para que se debe enseñar. Enseñar cálculo diferencial e integral representa uno de los mayores retos en el nivel superior en las especialidades de ciencias e ingeniería. En particular, el cálculo diferencial se aplica en situaciones como aproximaciones (límites), la razón de cambio, la teoría de errores, trazado de gráficas, la optimización.

En lo referente a trazado de gráficas y optimización se recurre a los criterios de la primera y segunda derivada como parte del procedimiento que nos lleva a localizar los valores extremos entendidos como máximos y/o mínimos relativos que pueden ser denominados globales en caso no exista otro valor para la función que sea considerado mayor o menor que los anteriores. Ambos criterios aportan lo necesario para la determinación de los puntos de inflexión, es decir, puntos donde la trayectoria de la curva cambia.

Por otra parte, la tecnología informática ha evolucionado y se ha insertado dentro del quehacer educativo. Los procesos de enseñanza en el área de matemáticas han incorporado tecnologías que van desde el uso del CD y los manuales multimedia (primeros esfuerzos en educación a distancia), presentaciones gráficas, hasta el uso de software como Geogebra , Derive, Winplot ,Matlab , Symbolab , WolframAlpha. Algunos de estos presentan versiones online (páginas web) desde las cuales también puede descargar versiones para Desktop, Tablet e inclusive para móviles (Apps).

En este contexto es donde un estudiante puede encontrar alguna dificultad para confrontar sus procedimientos y cálculos con el software existente. En la literatura sobre el tema de la aplicación de los criterios de la primera y segunda derivada para calcular máximos y mínimos se presenta algunas situaciones discutibles que se pone en evidencia en el libro de cálculo Matemáticas I Cálculo Diferencial de Denis Zill & Warren Wrigth (Zill & Wrigth , 2010) que refiere una falla en el criterio de la segunda derivada , así como en el libro Cálculo Tomo I de Piskunov (Piskunov, 1983) que refiere dificultades que presentan los criterios de la primera y segunda derivada y que también se ha evidenciado al usar software como WolframAlpha y Symbolab. En el caso de Symbolab se utilizó como elemento de discusión la función $f(x) = x^4$ la cual se ingresó en la opción Aplicaciones de la derivada: puntos extremos donde al utilizar la condición criterio de la segunda derivada nos ofrece como resultado que en el punto x=0 es **poco concluyente**

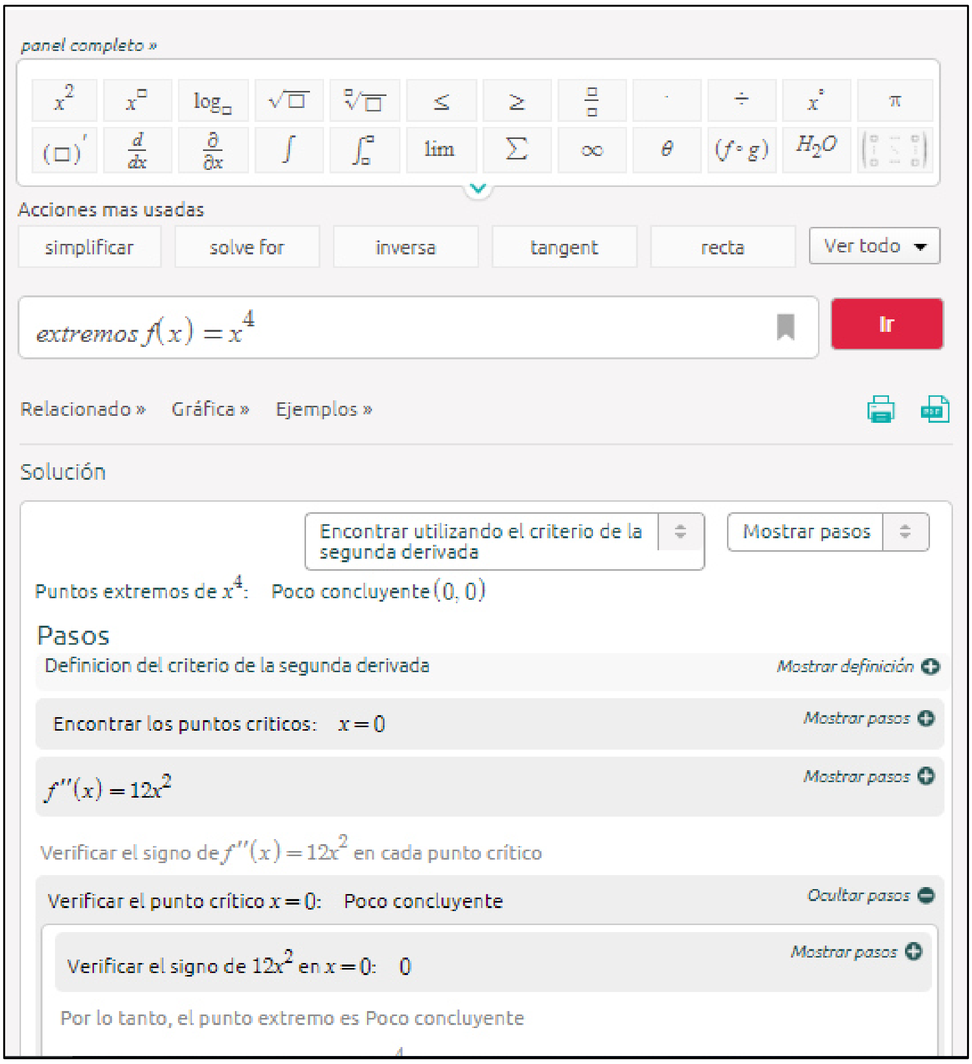
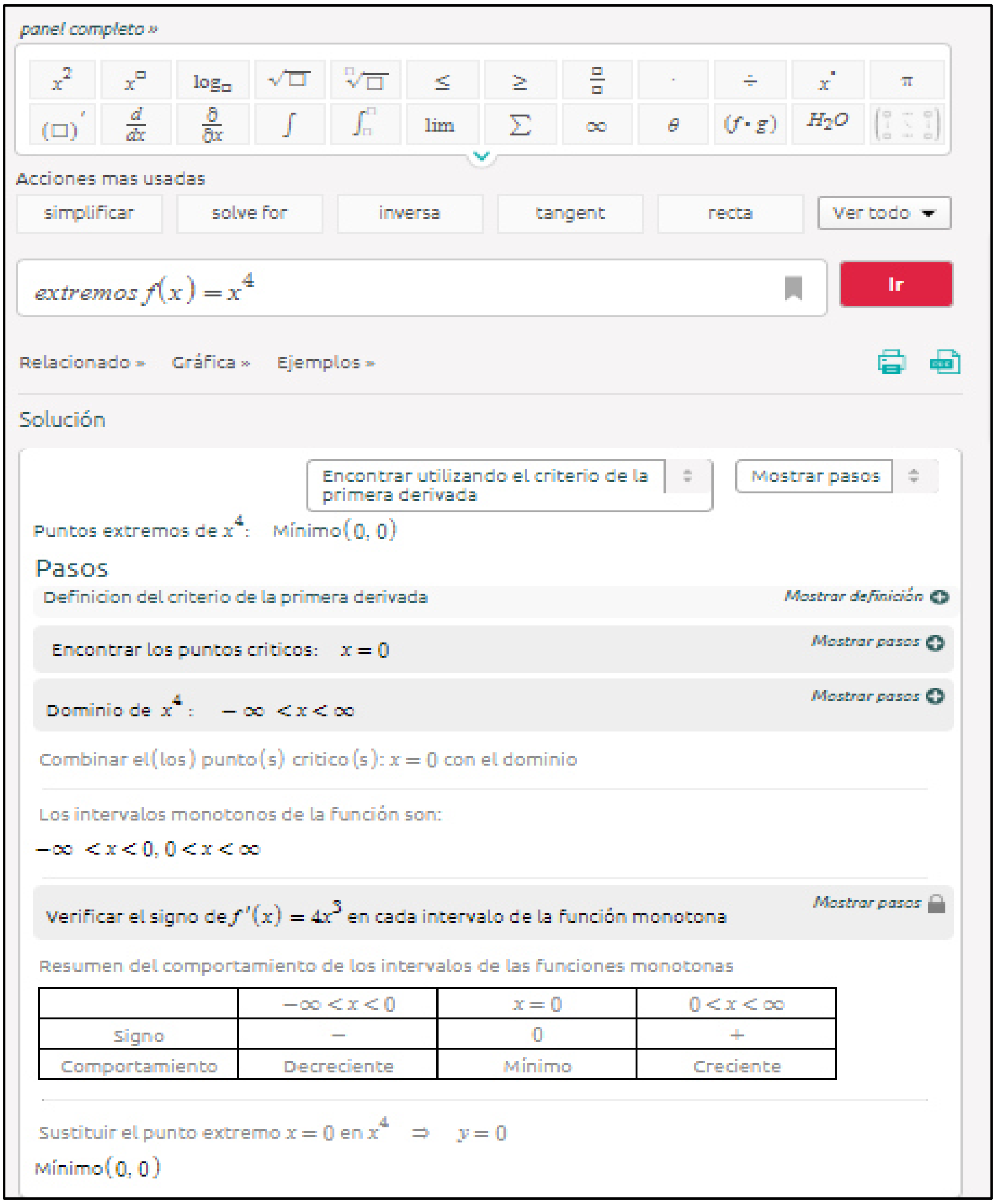


Figura N°1 : Puntos extremos de $f(x) = x^4$ con segunda derivada
Fuente: Symbolab

En su defecto Symbolab permite corregir esta situación al utilizar la condición criterio de la primera derivada tal como muestra la figura Nro. 2 donde el punto x=0 se muestra como un mínimo de la función.



	$-\infty < x < 0$	$x=0$	$0 < x < \infty$
Signo	$-$	0	$+$
Comportamiento	Decreciente	Mínimo	Creciente

Figura N° 2: Puntos extremos de $f(x) = x^4$ con la primera derivada
Fuente: Symbolab

Un caso diferente fue el análisis en la página de WolframAlpha donde se utilizó la función $g(x) = x^8 - x^6$ con mayor grado respecto al anterior. En la figura 3 se presenta los resultados que ofrece WolframAlpha

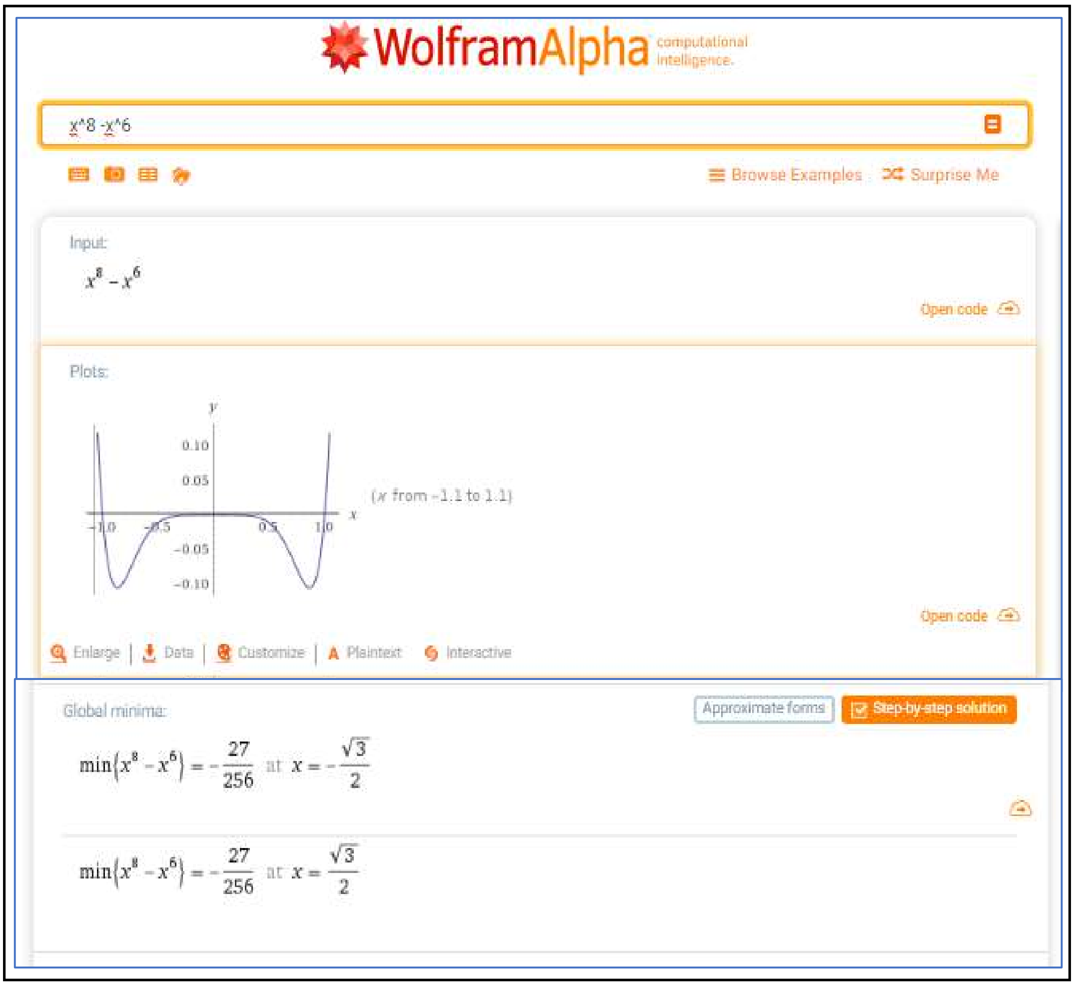


Figura N° 3: Puntos extremos de $f(x) = x^8 - x^6$

Fuente: software WolframAlpha

Solo se observa dos puntos extremos mínimos, pero en la gráfica se puede apreciar un punto máximo local en x=0. La situación descrita genera diferencia entre lo resuelto al aplicar los conocimientos teóricos y lo que se obtiene al utilizar software.

Luego de revisar libros de Cálculo diferencial para conocer el enfoque de resolución de problemas en la determinación de máximos y mínimos relativos, valores útiles en la teoría de optimización , se encontró en el libro Análisis I de José Tola Pasquel (Tola ,1970) la referencia del criterio de la derivada enésima ,el cual inspira un nuevo enfoque (un camino diferente), para alcanzar los extremos locales y puntos de inflexión de funciones siendo este enfoque el motivo de la investigación el cual será implementado en un software orientado a web constituyendo un nuevo camino para obtener los máximos y minimos relativos e inflexiones de funciones polinómicas.

1.2 Formulación del Problema

1.2.1 Problema general

¿El desarrollo del nuevo software DERIN, aplicando el criterio de la derivada enésima, constituirá una nueva alternativa que corregirá las debilidades que presenta el software existente al determinar los máximos y mínimos relativos y los puntos de inflexión de una función polinómica, favoreciendo su aplicación en la optimización de funciones, la enseñanza y transferencia de conocimientos en los cursos de cálculo diferencial?

1.2.2 Problemas específicos

Problema específico 1

¿El desarrollo del nuevo software DERIN, aplicando el criterio de la derivada enésima, constituirá una nueva alternativa que corregirá las debilidades que presenta el software existente para calcular los máximos y mínimos relátivos de una función polinómica?

Problema específico 2

¿El desarrollo del nuevo software DERIN, aplicando el criterio de la derivada enésima, constituirá una nueva alternativa que corregirá las debilidades que presenta el software existente para calcular puntos de inflexión de una función polinómica?

Problema específico 3

¿El nuevo software DERIN, aplicando el criterio de la derivada enésima, al corregir las debilidades del software existente entonces contribuirá en la optimización de funciones, la enseñanza y transferencia de conocimientos en los cursos de cálculo diferencial?

1.3 Justificación de la Investigación

"El cálculo diferencial e integral en una variable tiene como objeto de estudio las funciones, las cuales están presentes en todos los modelos matemáticos, lo cual hace indispensable su estudio en todas las ciencias básicas y sociales, así como en las ingenierías " (Cuevas y Pluvinage, 2009).

El desarrollo de ciencia y tecnología no habría sido posible sin el concurso de esta área del conocimiento. En particular, la derivada de una función es un tema con diversas aplicaciones dentro del desarrollo de nuestra sociedad. Podemos observar su inclusión en Economía, Ciencias Sociales, Física y otras disciplinas.

La investigación propone el desarrollo de nuevo software "DERIN" para determinar los extremos relativos (locales) y puntos de inflexión utilizando el criterio de la derivada enésima. Al respecto se debe indicar que existe una gran cantidad de software dedicado al cálculo de las derivadas sin embargo nuestro aporte consiste en ofrecer un elemento teórico adicional: la derivada enésima cuyo algoritmo implementado en un software pueda ser valorado como una nueva alternativa para ser usado en el cálculo de los máximos y mínimos relativos desde dos puntos hoy concatenados. El primero definido sobre la parte teórica, pues el criterio de la derivada enésima corrige las dificultades que pueden presentar los criterios de la primera y segunda derivada lo que influye en forma directa en la optimización de funciones, trazados de gráficas. El segundo punto es referente al uso de herramientas tecnologícas (software) como apoyo en los cursos de cálculo, puesto que la enseñanza a nivel universitario está inmersa en un proceso de cambios que obedece a diversos factores tales como la variabilidad de enfoques de conocimiento, modelos pedagógicos, libros de texto, materiales audiovisuales y software educativo, todo lo cual se incrementa cada año, además de abundante material disponible en forma gratuita a través de la Internet.

1.4 Objetivos de la Investigación

1.4.1 Objetivo General

Desarrollar nuevo software, DERIN, aplicando el criterio de la derivada enésima como una nueva alternativa para corregir las debilidades que presenta el software existente al determinar los máximos y mínimos relativos y los puntos de inflexión de una función polinómica, favoreciendo su aplicación en la optimización de funciones, la enseñanza y transferencia de conocimientos en los cursos de cálculo diferencial.

1.4.2 Objetivos Específicos

a) Desarrollar el nuevo software DERIN aplicando el criterio de la derivada enésima para aproximar los máximos y mínimos relátivos de una función polinómica

b) Desarrollar el nuevo software DERIN aplicando el criterio de la derivada enésima para calcular los puntos de inflexión de una función polinómica

c) Utilizar el nuevo software DERIN como recurso de apoyo para la optimización de funciones, trazado de gráficas, así como la enseñanza y transferencia de conocimientos en los cursos de cálculo diferencial.

1.5 Hipótesis

1.5.1 Hipótesis general

El desarrollo de nuevo software, DERIN, aplicando el criterio de la derivada enésima es una nueva alternativa para corregir las debilidades que presenta el software existente al determinar los máximos y mínimos relativos y los puntos de inflexión de una función polinómica, siendo aplicable en la optimización de funciones, la enseñanza y transferencia de conocimientos en los cursos de cálculo diferencial.

1.5.2 Hipótesis especificas

a) El desarrollo de nuevo software DERIN aplicando el criterio de la derivada enésima representa una nueva alternativa para obtener los máximos y mínimos relativos de una función polinómica.

b) El desarrollo de nuevo software DERIN aplicando el criterio de la derivada enésima representa una nueva alternativa para obtener los puntos de inflexión de una función polinómica.

c) El nuevo software DERIN aplicando el criterio de la derivada enésima constituirá un recurso de apoyo para la optimización de funciones, asi como para enseñanza y transferencia de conocmientos en los cursos de cálculo diferencial.

1.6 Variables

Variable independiente	: Desarrollo de software
Variable Dependiente	: Cálculo de máximos, mínimos y puntos de Inflexión de una función polinómica
Variable Interviniente	: Criterio de la derivada enésima, Optimización de funciones, enseñanza , transfencia de conocmientos

1.7 Operacionalización de variables:

1.7.1 Variable Independiente.

La Variable Independiente es el desarrollo de software DERIN con sus indicadores los cuales son:

- Implementación de nuevos algoritmos
  Algoritmo de la derivada enésima
  Algoritmo para aplicar Método de Graeffe en el cálculo de raíces de polinomios
- Desarrollo de software orientado a web
- Transferencia e intercambio de conocimiento.

1.7.2 Variable Dependiente

La Variable dependiente es el cálculo de máximos, mínimos y puntos de inflexión de funciones polinómicas de una variable propuesto con sus indicadores:

- Pruebas de software vs problemas de texto
- Pruebas de software vs software existente

1.7.3 Variable Interviniente

Dentro del desarrollo de software contempla el uso de algoritmo para el criterio de la derivada enésima, influencia en procesos de optimización, enseñanza.

En el cuadro N° 1 se muestra la matriz de consistencia sobre la investigación y relacionando sus parámetros : problema, objetivo , hipótesis , variables

Cuadro N°1 Matriz de consistencia: Desarrollo de nuevo software

Problema general	Objetivo General	Hipótesis General	Metodología de la investigación	Variables	Dimensiones
¿El desarrollo del nuevo software DERIN, aplicando el criterio de la derivada enésima, constituirá una nueva alternativa que corregirá las debilidades que presenta el software existente para calcular los máximos y mínimos relativos, así como los puntos de inflexión de una función polinómica, favoreciendo su aplicación en la optimización de funciones, la enseñanza y transferencia de conocimientos en los cursos de cálculo diferencial?	Desarrollar nuevo software, DERIN, aplicando el criterio de la derivada enésima como una nueva alternativa para corregir las debilidades que presenta el software existente al calcular los máximos y mínimos relativos, así como los puntos de inflexión de una función polinómica, favoreciendo su aplicación en la optimización de funciones, la enseñanza y transferencia de conocimientos en los cursos de cálculo diferencial.	El desarrollo de nuevo software, DERIN, aplicando el criterio de la derivada enésima es una nueva alternativa para corregir las debilidades que presenta el software existente al calcular los máximos y mínimos relativos, así como los puntos de inflexión de una función polinómica, siendo aplicable en la optimización de funciones, la enseñanza y transferencia de conocimientos en los cursos de cálculo diferencial. .	Tipo Explicativa Tecnológica Nivel Aplicada Diseño Pre experimental Método Prueba de software	**Independiente** Desarrollo de software **Dependiente** Cálculo de máximos, mínimos y puntos de inflexión **Interviniente** Criterio de la derivada enésima, optimización, enseñanza	**Tecnológica** Creación de software de aplicación orientado a web Implementación de nuevos algoritmos **Innovación** Búsqueda de nuevos caminos, enfoques para la solución de problemas
Problemas específicos	**Objetivos específicos**	**Hipótesis específicos**			
¿El desarrollo del nuevo software DERIN, aplicando el criterio de la derivada enésima, constituirá una nueva alternativa que corregirá las debilidades que presenta el software existente para calcular los máximos y mínimos relátivos de una función polinómica?	Desarrollar el nuevo software DERIN aplicando el criterio de la derivada enésima para calcular los máximos y mínimos relátivos de una función polinómica	El desarrollo de nuevo software DERIN aplicando el criterio de la derivada enésima representa una nueva alternativa para obtener los máximos y mínimos relativos de una función polinómica.			
¿El desarrollo del nuevo software DERIN, aplicando el criterio de la derivada enésima, constituirá una nueva alternativa que corregirá las debilidades que presenta el software existente para calcular puntos de inflexión de una función polinómica?	Desarrollar el nuevo software DERIN aplicando el criterio de la derivada enésima para calcular los puntos de inflexión de una función polinómica	El desarrollo de nuevo software DERIN aplicando el criterio de la derivada enésima representa una nueva alternativa para obtener los puntos de inflexión de una función polinómica.			
¿El nuevo software DERIN, aplicando el criterio de la derivada enésima, al corregir las debilidades del software existente entonces contribuirá en la optimización de funciones, la enseñanza y transferencia de conocimientos en los cursos de cálculo diferencial?	Utilizar el nuevo software DERIN como recurso de apoyo para la optimización de funciones, trazado de gráficas, así como la enseñanza y transferencia de conocimientos en los cursos de cálculo diferencial.	El nuevo software DERIN aplicando el criterio de la derivada enésima constituirá un recurso de apoyo para la optimización de funciones, asi como para enseñanza y transferencia de conocmientos en los cursos de cálculo diferencial			

Fuente: Elaboración propia

1.8 Limitaciones

La investigación pretende el desarrollo del software DERIN como un recurso tecnologico que ayude en el cálculo de los extremos relativos y los puntosde inflexión de una función polinómica, de modo que el software presente las soluciones con su gráfica asociada en dos dimensiones. La aplicación ha sido implementada para funciones polinómicas de una variable cuyo grado sea menor a seis, exceptuando otros tipos de funciones como las exponenciales, logarítmicas y trigonométricas.

Los métodos numéricos para el cálculo de raíces de polinomios presentan debilidades dado que se analizan sobre ciertos intervalos, además de responder a criterios de convergencia. Otros factores presentes son la división por cero y el cálculo de logaritmos, considerándose para tal efecto funciones polinómicas cuyo grado sea menor a seis lo que no desvirtúa el uso con otras funciones polinómicas donde encontraremos algunas dificultades por ausencia de métodos generales para calcular raíces de polinomios.

1.9 Alcances

La investigación está enmarcada dentro del área de matemáticas, específicamente en el cálculo diferencial, dirigido a la comunidad de personas vinculadas con la ciencia. La esencia del software se fundamenta en la mejora del cálculo de los puntos máximos y mínimos relativos de modo que en forma paralela apoyará a otras disciplinas donde se utiliza el modelamiento de funciones y en las cuales sea necesario la optimización de la misma.

1.10 Viabilidad

La investigación es viable desde el punto de vista tecnológico a nivel de hardware y software. La programación está garantizada en un entorno de programación NET BEANS que da soporte al lenguaje Java considerando requisitos mínimos de trabajo y con el apoyo del personal encargado de la programación quienes dividen esfuerzos para generar el software de aplicación y la documentación. Desde el punto de vista teórico los fundamentos del algoritmo se referencian en libros de cálculo y con demostración pertinente que aporta confiabilidad al software. La compatibilidad no representa dificultad pues al estar orientado a web solo requiere conexion a internet desde cualquier dispositivo.

En lo referente a la parte operativa estará al alcance de toda la comunidad de estudiantes, quienes de forma sencilla podrán hacer uso del software.

CAPÍTULO II

MARCO TEÓRICO

CAPÍTULO II

MARCO TEÓRICO

Bases Teóricas

2.1. Situación en la Enseñanza del Cálculo

La enseñanza del cálculo es uno de los desafíos de la educación actual porque es de carácter formativo en los planes curriculares de instituciones técnicas y universitarias. En forma paralela su aprendizaje presenta dificultades donde están presentes las capacidades de abstracción, análisis, visualización, demostración y argumentación. Sin embargo el fracaso de los estudiantes por carecer de la preparación adecuada en álgebra, artimetica, geometría termina siendo una creencia o un mito, pues los y las estudiantes pueden tener todos estos conocimientos, capacidades y herramientas tecnológicas inclusive y aun así fracasar en el estudio del cálculo diferencial e integral.

María del Mar Moreno Moreno en El papel de la didáctica en la enseñanza del cálculo (2005) señala lo siguiente:

> "Los intentos por reformar la enseñanza del cálculo han sido muchos y distintos. La mayoría de los programas de renovación han compartido los mismos criterios, y creen que los cambios deben afectar a los aspectos de currículo vigente, desarrollo profesional de la universidad, a la utilización sistemática de la tecnología y a la formación didáctica y científica de los docentes. Los métodos tradicionales de enseñanza de las matemáticas en el nivel superior se centralizan en la parte algorítmica y algebraica del cálculo, que acaba siendo rutinaria (Artigue, 1995). Muchas de las innovaciones curriculares son reformas orientadas al campo tecnológico, siendo un caso es el australiano que se centralizó en el desarrollo de software que facilita las representaciones gráficas y algebraicas" (Moreno, 2005: pag 82,84)

En síntesis, la búsqueda de mejoras en la forma de enseñar el cálculo que conlleve a un mejor aprender impulsa a reflexionar sobre las características de los estudiantes (son muy diferentes los de hace veinte años a los actuales), la sociedad en la que están inmersos y la tecnología que modifica sus patrones socio- culturales que influyen en los entornos de aprendizaje. (Cuevas y Pluvinage, 2009)

2.2 Raíces de un polinomio

Eduardo Espinoza en su libro Números Complejos y ecuaciones Polínómicas (2000) señala:

> " De acuerdo al teorema del factor se conoce que dado un polinomio P8x) con grado $n \geq 1$, un número r se llama raíz o cero del polinomio P(x) si P(r)=0 " (Espinoza, 2000: pag 100).

Por ejemplo, dado el polinomio:

$P(x) = x^2 - 7x + 12$

Planteando y resolviendo la ecuación:

$$x^2 - 7x + 12 = 0$$

Se tiene que 3 y 4 son raíces ya que *P*(3) = 0 y *P*(4) = 0.

2.3 Método Numérico

"Los métodos numéricos son técnicas mediante las cuales es posible formular problemas de tal forma que sean resueltas con operaciones aritméticas, Aunque hay muchos tipos de métodos numéricos todos comparten una característica común, llevan cabo un buen número de tediosos cálculos aritméticos" (Chapra, Steven C. & Canale, Raymond, 2011, pag 3).

"Los métodos numéricos nos vuelven aptos para entender esquemas numéricos a fin de resolver problemas matemáticos, de ingeniería y científicos en una computadora, sintetizar esquemas numéricos básicos, escribir programas y resolverlos en una computadora y usar correctamente el software existente para dichos métodos y no solo aumenta nuestra habilidad para el uso de computadoras, sino que también amplia la pericia matemática y la comprensión de los principios científicos básicos" (Nakamura, Schoichiro, 1992: Prefacio).

2.4 Método numérico para calcular las raíces de un polinomio

Desde el siglo XX, el crecimiento tecnológico ha contribuido a una gran bonanza en el uso y desarrollo de métodos numéricos siendo uno de sus primeros retos el cálculo de raíces de ecuaciones. El desarrollo de métodos numéricos en el tema de las raíces es amplio disponiendo de métodos cerrados y abiertos.

En el caso de los métodos cerrados presentan como dificultad la lenta convergencia y la necesidad de dos o más valores iniciales. En este grupo podemos mencionar la falsa posición y la bisección.

“Los métodos abiertos se fundamentan en fórmulas que necesitan un valor inicial que no necesariamente contiene a las raíces. Éstos, algunas veces divergen o se alejan de la raíz verdadera a medida que se avanza en el cálculo, pero cuando convergen lo hacen más rápido que los métodos cerrados” (Chapra, Steven C. & Canale, Raymond, 2011, pag 130). Podemos mencionar al método de punto fijo; de Newton-Raphson, de la secante, de Müller.

Un método interesante es el de Graeffe, que ha sido escogido para la búsqueda de raíces dentro del desarrollo de software, como parte de la solución de un problema. Es importante recalcar que la solución de problemas tiene un antes (a) y un después (b) del avance tecnológico como lo explica Chapra, Steven C. & Canale, Raymond en su libro sobre Métodos Numéricos que se reproduce en la figura N° 4

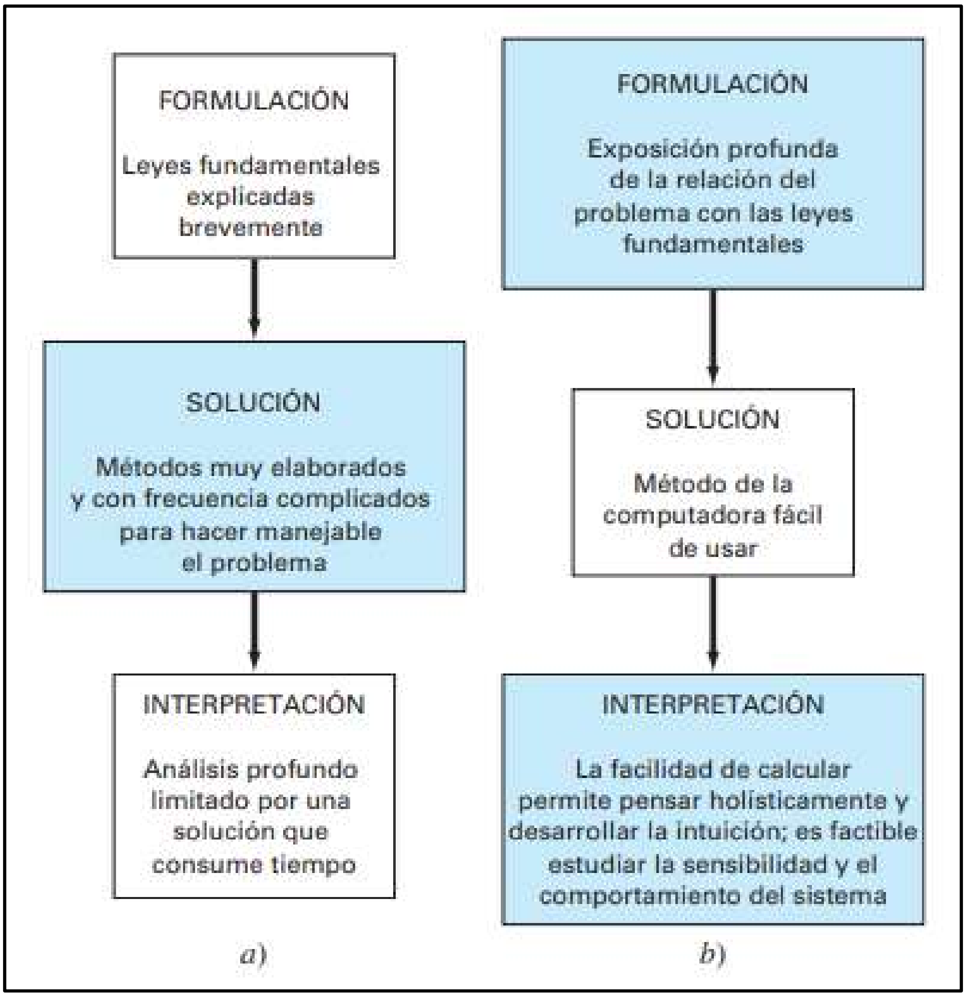


Figura N° 4: Fases en la solución de problemas
Fuente: Extraido de Métodos Numéricos Steven Chapra & Raymond Canale pág. 4.

En atención a los textos refereridos de Steven CHapra y Raymond Canale así como de Efracio Asís López (Métodos numéricos con MATLAB) se expone algunos métodos numéricos utilizados para aproximación de raíces

2.4.1 Método de Bisección

Es un método basado en el Teorema de Bolzano, es un tipo de búsqueda incremental en el que el intervalo se divide siempre a la mitad, tratándose de ubicar un cambio de signo de la función. La raíz se ubica en el punto medio del subintervalo. La figura N° 5 ilustra el método de bisección

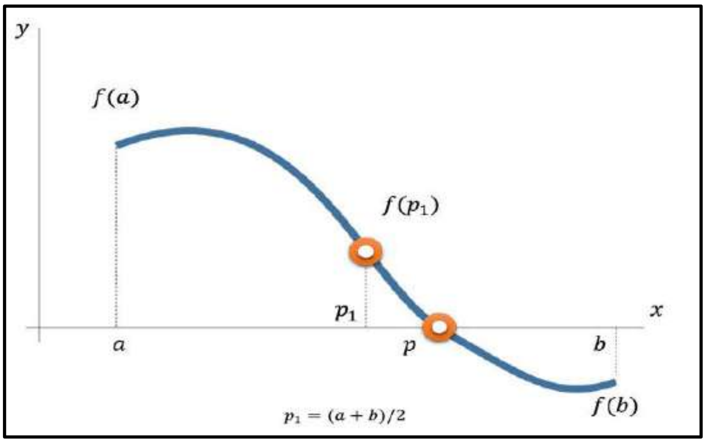


Figura N° 5 Método de Bisección
Fuente: Trucos y Cursos (2016) Método numérico Bisección en Excel
Recuperado de. http://trucosycursos.es/metodo-numerico-biseccion-en-excel/

2.4.2 Método de la Falsa Posición

Conocido como Regula Falsi proviene de la frase latina que significa regla inclinada. Es similar al método de bisección generando los subintervalos que encierran a la raíz, pero esta vez no es el punto medio del intervalo, sino el punto de intersección de la recta que pasa por los puntos $(a_n, f(a_n))$ y $(b_n, f(b_n))$ con el eje X. La figura N° 6 ilustra el método de la falsa posición

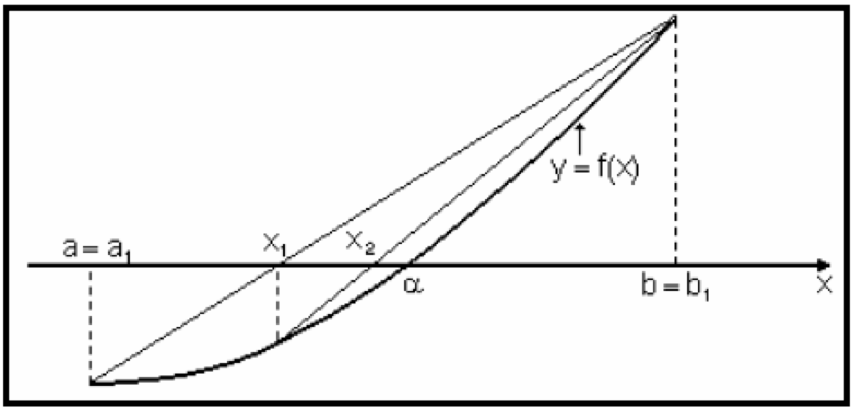


Figura N°6 Método Regula Falsi
Fuente: Pacheco, Diego (2011) Método de la Regla Falsa
Recuperado de http://esimecu-anumerico.blogspot.com/2011/06/metodo-de-la-regla-falsa.html

2.4.3 Método de Punto fijo

Para encontrar una raíz debemos generar una ecuación de la forma $x = g(x)$ y mediante la iteración $x_{n+1} = g(x_n)$ con n=1;2;3;… y un x_0 dado . La figura N°7 ilustra el método de punto fijo

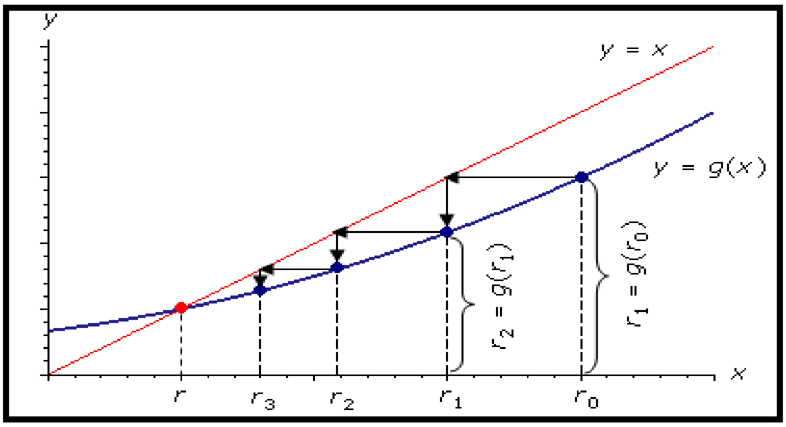


Figura N° 7 Método Punto Fijo
Fuente: Jaramillo, Daniel (2014) Punto fijo
Recuperado de : https://sites.google.com/site/danielmalteymariaj/home/integrantes/punto-fijo

2.4.4 Método de Newton- Raphson

Es muy utilizado por la cantidad de aplicaciones directas y cantidad de generalizaciones, modificaciones y aplicaciones derivados del mismo. Se basa en la fórmula de iteración $x_n = x_{n-1} - \frac{f(x_{n-1})}{f'(x_{n-1})}$ con n=1; 2;… y escogiendo un x_0 “cercano” a la raíz. La figura N° 8 ilustra el método descrito

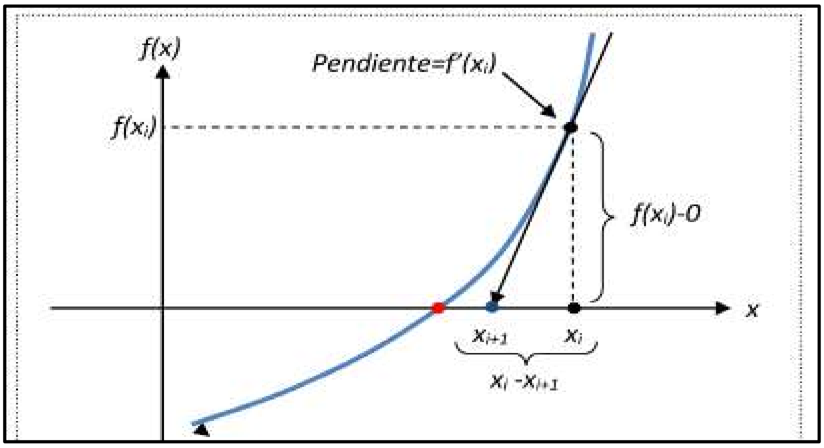


Figura N° 8 Método Newton - Raphson
Fuente: Flores, Miguel (2016) Método de Newton Raphson en Matlab
Recuperado de https://medium.com/@hdezfloresmiguelangel/m%C3%A9todo-de-newton- raphson-en-matlab-ef86f1972e4

2.4.5 Método de la secante

Este método trata de no utilizar la derivada como lo requiere el método de Newton, para ello lo aproxima mediante una diferencia finita dividida hacia atrás mediante la fórmula de iteración $x_{i+1} = x_i - \frac{f(x_i)(x_{i-1}-x_i)}{f(x_{i-1})-f(x_i)}$. La figura N° 9 ilustra la idea del uso de este método.

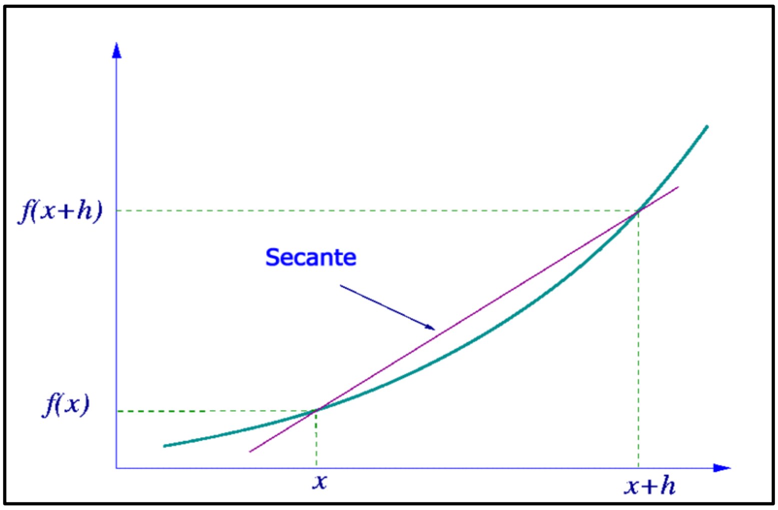


Figura N°9 Método de la Secante
Fuente: Manske, Magnus (2011)
Recuperado de https://es.wikipedia.org/wiki/Archivo:Secante-calculo.gif

2.4.6 Método de Graeffe para calcular las raices de un polinomio

Este método fue descubierto independientemente por Graeffe (o Gräffe), Dandelin, y Lobachevsky. La idea general es, tomar un polinomio $P_1(x)$ y crear un polinomio $P_2(x)$ cuyas raíces sean los cuadrados de las raíces de $P_1(x)$. Es decir, si las raíces de $P_1(x)$ son $r_1, r_2, r_3 \ldots r_n$ entonces las raíces de $P_2(x)$ serán ${r_1}^2, {r_2}^2, {r_3}^2 \ldots {r_n}^2$. Se repite el proceso "n" veces, así se obtiene un polinomio $P_m(x)$ cuyas raíces son ${r_1}^{2^m}, {r_2}^{2^m}, {r_3}^{2^m} \ldots {r_n}^{2^m}$. Este procedimiento hace que las raíces que antes estaban juntas se separen, y permite usar las fórmulas de Vieta (relaciones entre coeficientes y raíces) para aproximarlas.

Dado P(x) un polinomio de grado n, tratamos de que el coeficiente principal sea 1 (si no lo es, simplemente dividimos el polinomio entre el coeficiente principal). Entonces el polinomio P(x) se puede escribir en forma factorizada como: $P(x) = (x - r_1)(x - r_2)(x - r_3) \ldots . (x - r_n)$ siendo $r_1, r_2, r_3 \ldots r_n$ sus raíces. Se considera $P(-x) = (-x - r_1)(-x - r_2)(-x - r_3) \ldots . (-x - r_n)$ entonces se multiplica $P(x).P(-x) = (x - r_1)(x - r_2)(x - r_3) \ldots . (x - r_n)(-x - r_1)(-x - r_2)(-x - r_3) \ldots . (-x - r_n)$

Se obtiene

$$P(x).P(-x) = (x - r_1)(-x - r_1)(x - r_2)(-x - r_2)(x - r_3)(-x - r_3) \ldots (x - r_n)(-x - r_n)$$

$= (x^2 - r_1{}^2)(x^2 - r_2{}^2)(x^2 - r_3{}^2)\ldots.(x^2 - r_n{}^2)$
Este nuevo polinomio, cuando se mira en función de x^2 , tiene raíces $r_1{}^2, r_2{}^2, r_3{}^2 \ldots r_n{}^2$, entonces ahora hacemos la sustitución $y = x^2$ y al polinomio producto lo cambiamos por $Q(y)$ se obtendría: $Q(y) = (x^2 - r_1{}^2)(x^2 - r_2{}^2)(x^2 - r_3{}^2)\ldots.(x^2 - r_n{}^2)$.
Repetir el procedimiento varias veces, nos permite llegar a un polinomio: $P_m(x). = x^n + a_1x^{n-1} + a_2x^{n-2} + a_3x^{n-3} + \ldots + a_n$ con raíces $r_1{}^{2^m}, r_2{}^{2^m}, r_3{}^{2^m} \ldots r_n{}^{2^m}$.
Sea $r_1{}^{2^m} = b_i$ (una de las raíces). Ahora ordenemos las raíces de mayor a menor, como están separadas por elevarlas al exponente 2^m, idealmente tendrán gran diferencia en su magnitud, por lo que si $r_1 > r_2 > r_3 > \ldots > r_n$ se puede escribir $b_1 >> b_2 >> b_3 >> \ldots >> b_n$.
Las fórmulas de relaciones entre coeficientes y raíces se deducen de la fórmula de Vieta:
$a_1 = -(b_1 + b_2 + \ldots b_n)$ (suma de raíces)
$a_2 = b_1b_2 + b_1b_3 + \ldots b_{n-1}b_n$ suma de productos binarios
$a_3 = -(b_1b_2b_3 + b_1b_2b_4 + \ldots b_{n-2}b_{n-1}b_n)$ (suma de productos por ternas)
$\ldots$
$a_n = (-1)^n b_1b_2b_3b_n$ Producto de las raíces
Pero al observar que $b_1 >> b_2 >> b_3 >> \ldots >> b_n$, se afirma que se aproxima bien tomando el primer término (ya que al estar ordenado y por los cambios efectuados es de lejos el valor mayor), por lo tanto:
$a_1 = -b_1$
$a_2 = b_1b_2$
$a_3 = -b_1b_2b_3$
$\ldots$
$$a_n = (-1)^n b_1b_2b_3b_n$$
Deshaciendo el cambio, obtenemos:
$$|r_1| = \sqrt[2^m]{a_1}$$
$$|r_2| = \sqrt[2^m]{\frac{a_2}{a_1}}$$
$$|r_1| = \sqrt[2^m]{\frac{a_3}{a_2}}$$
En general (se considera $a_0 = 1$):
$$|r_j| = \sqrt[2^m]{\frac{a_j}{a_{j-1}}}$$

Aproximando al valor absoluto de las raíces del polinomio original, ahora solo falta evaluar el polinomio original en $\pm|r_j|$ para ver el signo, o si $\pm|r_j|$ no es raíz del polinomio debido a que no tiene n raíces reales, porque el método falló, o porque necesitamos mejor aproximación.

2.5 Máximos y Mínimos de una función

Piskunov en su libro Cálculo Diferencial e Integral define máximo y mínimo de una función

"Se dice que una función $f(x)$ tiene un máximo en el punto x_1, si su valor es aquí mayor que en cualquier otro punto de cierto intervalo que comprende el punto x_1 . Es decir, la función tiene un máximo en $x = x_1$, si $f(x_1 + \Delta x) < f(x_1)$ para todo valor de Δx (positivo o negativo) suficientemente pequeño en valor absoluto (Piskunov,1983, pag 169)

"Del mismo modo una función $f(x)$ tiene un mínimo para $x = x_2$, si $f(x_2 + \Delta x) > f(x_2)$ para cualquier valor de Δx (positivo o negativo) suficientemente pequeño en valor absoluto" (Piskunov,1983, pag 169).

La imagen presentada en la figura N° 10 representa una función que posee un punto máximo y un mínimo además de otro punto particular que no es ni máximo ni mínimo al que se denomina punto de inflexión .

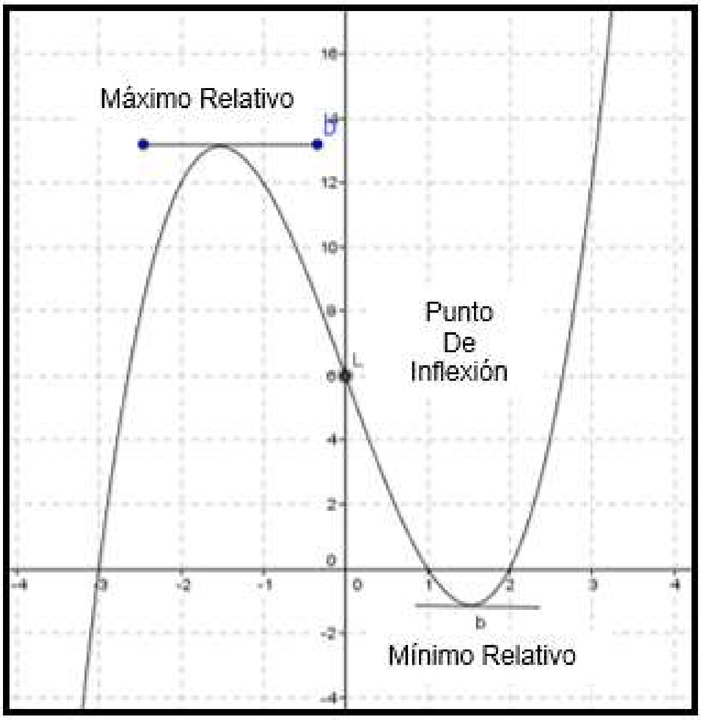


Figura N° 10: Máximos y Mínimos de una función
Fuente: GeoGebra

2.6 Cálculo de Máximos y Mínimos de una función

En el libro Análisis Real I su autor, José Tola Pasquel (1970) “considera que muchos problemas de aplicación práctica se reducen a la determinación del máximo o del mínimo de una función. La resolución de tales problemas consiste en obtener primero la función y luego en determinar el máximo o el mínimo utilizando criterios básicos sobre la derivación” (Tola, 1970: 303).

Los problemas utilizados ilustran ideas teóricas de modo que su solución, con arreglo a las técnicas que se deprenden de la teoría, no ofrecen dificultades invencibles. En la práctica la situación es generalmente otra y los procedimientos que se usen deben acomodarse a las circunstancias particulares. En algunos casos será necesario recurrir a métodos aproximados, o de aproximaciones sucesivas. Así por ejemplo si se desea obtener los valores extremos de la función $f(x) = x^8 + 2x^7 + 0{,}065x^5 + 2{,}5x^4 + 3{,}781x + 5$ sería necesario resolver la ecuación $8x^7 + 14x^6 + 0{,}325x^4 + 10x^3 + 3{,}781 = 0$ lo que no será posible por procedimientos algebraicos. Si la intención es definir valores extremos Piskunov (1983) sustenta su existencia con los siguientes teoremas:

- “Teorema 1 (Condición necesaria para la existencia de un valor extremo)
  Si la función derivable $y = f(x)$ tiene un máximo o un mínimo en el punto $x = x_1$, su derivada en ese punto se reduce a cero, es decir $f'(x_1) = 0$. Los valores en los que la derivada se reduce a cero se llaman valores o puntos críticos” (Piskunov 1983: 167).
- “Teorema 2 (Condiciones suficientes para la existencia de un valor extremo)
  Si una función $y = f(x)$ es continua sobre un cierto intervalo, al cual pertenece el punto crítico x_1 , y es derivable en cada punto del mismo (excepto, posiblemente, el mismo). Si al pasar por este punto x_1 de izquierda a derecha, el signo de la derivada cambia de “más” a “menos”, entonces la función admite máximo en $x = x_1$. Si al pasar por el punto x_1 , de izquierda a derecha, el signo de la derivada cambia de “menos a “más”) la función admite un mínimo en $x = x_1$” (Piskunov, 1983: 173).

2.6.1 Criterio de la Primera derivada

En base al análisis previo en lo expuesto en el ítem anterior se puede elaborar un procedimiento para el análisis de extremos relativos de una función derivable $y = f(x)$, al respecto Piskunov (1983) señala lo siguiente:

- "Calcule la primera derivada $f^{\prime}(x)$ de la función
- Calcule los valores críticos del argumento x, para lo cual es necesario:
    a) Igualar a cero la primera derivada y encontrar las raíces reales de la ecuación $f^{\prime}(x) = 0$
    b) Determinar los valores de x para los cuales la derivada $f^{\prime}(x)$ es discontinua.
- Analizar el signo de la derivada a la izquierda y la derecha del punto crítico
- Calcule los valores de la función $f(x)$ para cada valor crítico" (1983:175).

El siguiente cuadro extraído del texto de Piskunov (1983) resume los casos posibles.

Cuadro N°2 Criterio de la Primera derivada

Signo de la derivada $f^{\prime}(x)$ al pasar por el punto crítico x_1			Naturaleza del punto crítico
$x < x_1$	$x = x_1$	$x > x_1$	
+	$f^{\prime}(x) = 0$ o es discontinua	-	Punto máximo
-	$f^{\prime}(x) = 0$ o es discontinua	+	Punto mínimo
+	$f^{\prime}(x) = 0$ o es discontinua	+	No hay máximo ni mínimo (la función crece)
-	$f^{\prime}(x) = 0$ o es discontinua	-	No hay máximo ni mínimo (la función decrece)

Fuente: Extraido de Cálculo Diferencial e Integral Tomo I Piskunov N. pág. 176

Como ejemplo se analiza el máximo y el mínimo de la función $f(x) = \frac{x^3}{3} - 2x^2 + 3x + 1$

Se aplica el criterio de la primera derivada.

Solución

Se determina la primera derivada y se iguala a cero

$$x^2 - 4x + 3 = 0$$

Se factoriza y se obtiene como valores críticos $x_1 = 1$, $x_2 = 3$.

Se analiza los valores críticos siguiendo el análisis de signos

Como $x^2 - 4x + 3 = (x - 1)(x - 3)$

En x=1

Para x<1 se tiene $f^{\prime}(x) = (-).(-) > 0$

Para x>1 se tiene $f^{\prime}(x) = (+).(-) < 0$

Se confirma la existencia de un valor máximo

En x=3

Para x<3 se tiene $f'(x) = (+).(-) < 0$

Para x>3 se tiene $f'(x) = (+).(+) > 0$

Se confirma la existencia de un valor mínimo

2.6.2 Criterio de la Segunda derivada

Piskunov (1983) argumenta sobre el tema:

"Si en $x = x_1$ la derivada de la función $y = f(x)$ se reduce a cero, es decir $f'(x_1) = 0$, y se admite, además que existe la segunda derivada $f''(x)$ continua sobre cierta vecindad del punto x_1. Para este caso es válido el siguiente teorema.

Teorema. Si $f'(x_1) = 0$ entonces en $x = x_1$ la función tiene máximo cuando $f''(x_1) < 0$, y, un mínimo cuando $f''(x_1) > 0$" (Piskunov, 1983: 178).

El resultado del análisis de los valores extremos, mediante la segunda derivada, se representa en el cuadro extraido del texto de Piskunov (1983)

Cuadro N° 3 Criterio de la segunda derivada

$f'(x_1)$	$f''(x_1)$	Naturaleza del punto critico
0	-	Punto máximo
0	+	Punto mínimo
0	0	Desconocido

Fuente: Extraido de Cálculo Diferencial e Integral Tomo I Piskunov N. pág. 179

Se analiza el siguiente problema donde se debe determinar los extremos relativos de la función $f(x) = \frac{x^3}{3} - 3x^2 + 8x + 1$ aplicando el criterio de la segunda derivada

Solución

Se deriva e iguala a cero $f'(x) = 0$

$$x^2 - 6x + 8 = 0$$

Se factoriza y resuelve $x_1 = 2$, $x_2 = 4$.

Se deriva por segunda vez

Se obtiene $f''(x) = 2x - 6$

Se evalúa cada punto crítico

Si $x_1 = 2$ entonces $f''(x_1) = 2(2) - 6 = -2 \Rightarrow f''(x_1) < 0$ y por consiguiente en este punto se confirma la existencia de un máximo

Si $x_2 = 4$ entonces $f^{''}(x_2) = 2(4) - 6 = 2 \Rightarrow f^{''}(x_2) > 0$, por lo tanto en este punto se confirma la existencia de un mínimo

2.7 Optimización

La Real Academia de la Lengua española define la optimización como "la acción y efecto de optimizar" que a su vez significa "buscar la mejor manera de realizar una actividad". En diversas disciplinas, la optimización consiste en la selección del mejor elemento (con respecto a algún criterio) de un conjunto de elementos disponibles.

Daniel Alberto Riveros Vásquez en su tesis "Aplicación de la investigación de operaciones al problema de la distribución a una empresa de logística" nos proporciona una conceptualización de optimización

> "Una jerarquía de problemas ha emergido, junto con una correspondiente colección de técnicas para su solución. Estos problemas, son conocidos como el problema de la programación matemática, donde se encuentra el valor de x , tal que:
>
> $\min f(x_1, x_2, \ldots, x_n)$
>
> sujeto a
>
> $g_i(x_1, x_2, \ldots, x_n) \geq b \ \ , i = 1, \ldots, m$
>
> $h_j(x) = 0 \ , \qquad\qquad j = 1, \ldots, p$
>
> Si la restricción no existe, o es una restricción de igualdad, con menor o igual número de variables que la función objetivo entonces, el cálculo diferencial, da la respuesta, ya que solo se trata de buscar los valores extremos de la función" (Riveros Vásquez 2015:12).

La optmización en resumen consiste en maximizar o minimizar una determinada función objetivo seleccionando valores de entrada de un conjunto posible y admisible. En esta búsqueda donde existirán restricciones se intenta obtener el "mejor valor".

.

Esta teoría sobre extremos relativos permite resolver problemas de geometría, mecánica, economía, entre otras ciencias. Se presenta un ejemplo sobre optimización extraído del libro Cálculo Diferencial e Integral tomo I de Piskunov (Piskunov, 1983)

“La distancia que cubre un proyectil sobre la horizontal al ser lanzado con una velocidad inicial V_0 desde una pieza de artillería que tiene un ángulo de elevación θ respecto al horizonte se determina según la fórmula $R = \frac{{V_0}^2 sen2\theta}{g}$ donde g es la aceleración de la gravedad. Determine el ángulo con el cual la distancia R resultará máxima dada la velocidad inicial” (Piskunov 1983: 184).

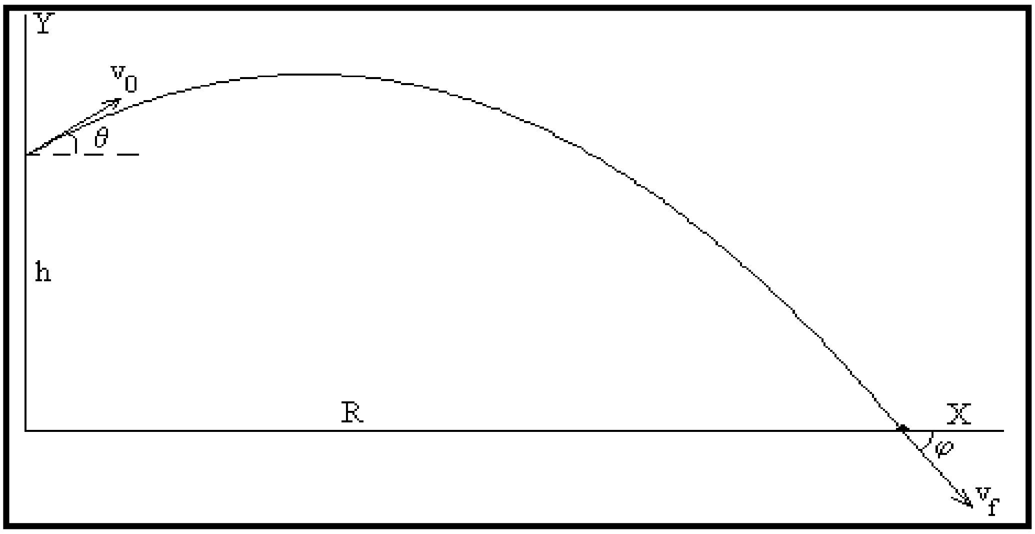


Figura N°11: Problema sobre Máximos y Mínimos de una función
Fuente Recuperado de http://www.sc.ehu.es/sbweb/fisica/cinematica/parabolico/alcance/alcance.htm

Diversos autores como Armando Venero en su texto Análisis Matemático I , Eduardo Espinoza Ramos en Análisis Matemático I , Moisés Lázaro Carrión en su obra Cálculo Diferencial , William Anthony Granville en Cálculo Diferencial e Integral , Louis Leithold en su libro Cálculo, definen problemas sobre extremos relativos aplicando los criterios de la primera y segunda derivada que luego complementan en situaciones de optimización.

2.8 Punto de Inflexión

George Thomas en su libro Cálculo de una variable expresa lo siguiente:
“Un punto donde la gráfica de una función tiene recta tangente y la concavidad cambia es un punto de inflexión”.
“El punto de una curva donde es positiva en un lado y negativa en el otro, es un punto de inflexión. En tal punto, es cero (porque las derivadas tienen la propiedad del valor intermedio), o no está definida. Si y es una función dos veces diferenciable, en un punto de inflexión, entonces tiene un máximo o mínimo local” (Thomas G, 2012).
Al respecto José Tola en Analisis Real I indica:

"Los puntos de inflexión pueden ser de primera especie, si sobre el punto particular x_0 su concavidad se dirige hacia abajo a la izquierda de x_0 y hacia arriba a su derecha; y de segunda especie si su concavidad se dirige hacia arriba a la izquierda de x_0 y hacia abajo a su derecha" (Tola 1970: 292).

2.9 Derivada Enésima

El proceso de derivación de funciones reales de variable real puede iterarse cierto número de veces así puede obtenerse la segunda, tercera y sucesivas derivadas de una función. En consecuencia para $n \in N$, la definición de la derivada enésima de una función se obtiene aplicando las fórmulas de derivación las veces que se requiera.

Sea la función $y = f(x)$ se puede obtener las derivadas sucesivas $f^{´}(x); f^{´´}(x); f^{´´´}(x); \ldots; f^{n}(x)$

Ejemplo

Dada la función $f(x) = 2x^4 - 3x^2 + 4x - 5$

Se obtiene las derivadas sucesivas

$$f^{´}(x) = 8x^3 - 6x + 4$$
$$f^{´´}(x) = 24x^2 - 6$$
$$f^{´´´}(x) = 48x$$
$$f^{iv}(x) = 48$$
$$f^{v}(x) = 0$$

Siendo la última iteración la que representa a la derivada de enésimo orden.

2.10 Argumentación matemática del algoritmo

El siguiente problema extraído del libro Cálculo Diferencial e Integral de Piskunov se utiliza para sostener el argumento del criterio de la derivada enésima aplicado a la obtención de extremos locales.

Sea la función $f(x) = -x^5$ determine los valores máximo y mínimo que pudiera presentar la función

Se aplica la primera derivada

Se obtiene $f^{´}(x) = -5x^4$

Se iguala a cero, $-5x^4 = 0$ al resolver la ecuación se obtiene el valor critico $x_1 = 0$

Se analiza con el criterio de la primera derivada

Para x<0 se tiene $x^4 > 0 \Rightarrow -5x^4 < 0 \Rightarrow f^{´}(x) < 0$

Para x>0 se tiene $x^4 > 0 \Rightarrow -5x^4 < 0 \Rightarrow f^{´}(x) < 0$

La conclusión es que en x =0 no existe ni máximo ni mínimo solamente que la función es decreciente.

Ahora se aplica el criterio de la segunda derivada

De la primera derivada $f'(x) = -5x^4$ volvemos a derivar y obtenemos

$$f''(x) = -20x^3$$

Se evalúa la segunda derivada en el valor crítico

$$f''(0) = -20(0)^3 = 0$$

Por lo tanto, la conclusión en x =0 es de un punto que no es máximo ni mínimo.

A continuación, se presenta la gráfica de la función generado con GeoGebra

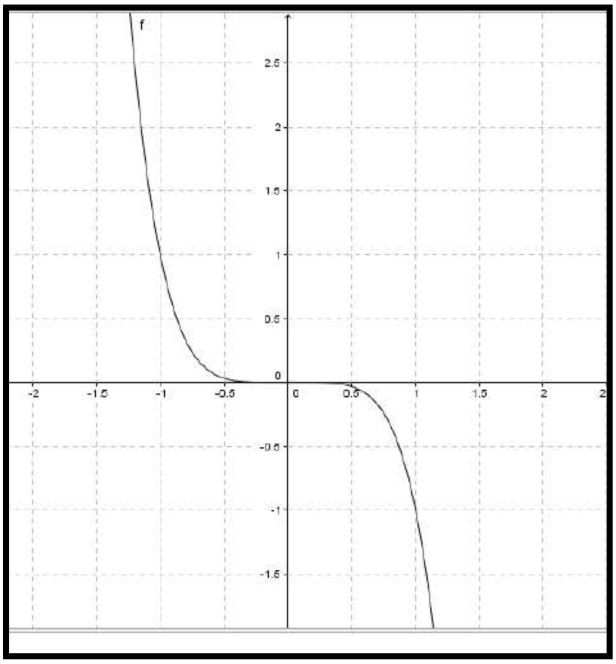


Figura N°12 Gráfica de la función $f(x) = -x^5$
Fuente: GeoGebra

2.11 La fórmula de Taylor

En el tomo I del libro Cálculo Diferencial e Integral de Piskunov (1983) detalla una explicación de como obtener extremos realtivos a partir de la fórmula de Taylor:

“Si en algún punto x=a se tiene que $f'(a) = 0 \; y \; f''(a) = 0$ se deduce que en dicho punto puede existir un máximo o un mínimo, o quizás ni máximo ni mínimo. Se puede utilizar la fórmula de Taylor para analizar esta situación.

Con el fin de generalizar el estudio suponemos que no solo $f'(x) = 0$, sino todas las derivadas de la función $f(x)$, hasta el orden enésimo inclusive, se reducen a cero cuando x=a:

$f'(a) = f''(a) = f'''(a) = \ldots = f^n(a) = 0$ mientras que $f^{n+1}(a) \neq 0$, suponiendo además que $f(x)$ tiene derivadas continuas hasta el orden (n+1) inclusive, en el entorno o vecindad del punto x=a.

De la fórmula de Taylor

$f(x) = f(a) + \frac{(x-a)}{1!} f'(a) + \frac{(x-a)^2}{2!} f''(a) + \frac{(x-a)^3}{3!} f'''(a) + \dots + \frac{(x-a)^{n+1}}{(n+1)!} f^{n+1}(a)$ se obtiene la forma reducida $f(x) = f(a) + \frac{(x-a)^{n+1}}{(n+1)!} f^{n+1}(a)$ donde a es un número comprendido entre a y x entonces despejamos y obtenemos

$f(x) - f(a) = \frac{(x-a)^{n+1}}{(n+1)!} f^{n+1}(a)$ Luego se analiza la segunda parte de la igualdad

Primer caso

Si n es un número impar entonces n+1 es par y por tanto $\frac{(x-a)^{n+1}}{(n+1)!}$ es positivo

Si $f^{n+1}(a) < 0$ entonces $f(x) - f(a) < 0$ lo que indica que la función tiene un máximo en x =a

Si $f^{n+1}(a) > 0$ entonces $f(x) - f(a) > 0$ lo que indica que la función tiene un mínimo en x=a

Segundo caso:

Si n es un número par entonces n+1 es impar entonces $\frac{(x-a)^{n+1}}{(n+1)!}$ tendrá signos opuestos cuando x>a y cuando x<a

Si $f^{n+1}(a) > 0$ entonces $f(x) - f(a) < 0$ para x<a y $f(x) - f(a) > 0$ para x>a

Si $f^{n+1}(a) < 0$ entonces $f(x) - f(a) > 0$ para x<a y $f(x) - f(a) < 0$ para x>a

Esto nos lleva a afirmar que no existe máximo, ni mínimo" (Piskunov, 1970: pag 185-187).

2.12 Criterio de la derivada enésima

Luego de efectuar una revisión de textos de nivel universitario de autores nacionales y extranjeros se evidencia que aplican el criterio de la primera y segunda derivada para la optimización, solamente en el texto Cálculo Diferencial e Integral Tomo I de Piskunov (Piskunov, 1983) se hace un análisis de la derivada enésima, así como en el texto Análisis I de José Tola (Tola, 1970). En dicho libro del reconocido matemático José Tola Pasquel se detalla un criterio que generaliza el análisis de valores extremos y de los puntos donde la concavidad cambia: el criterio de la derivada enésima cuya justificación tiene antecedente en la fórmula de Taylor.

Se deduce que los valores extremos están ligados a la derivada de orden n+1, es decir a la derivada enésima que no se anule.

Proposición

Sea f una función cuyas n ($n \geq 2$) primeras derivadas son continuas en un intervalo abierto I que contiene al punto x_0. La condición necesaria y suficiente para que

$f(x_0)$ sea una valor extremo de la función es que exista el número n tal que $f^p(x_0) = 0$ para p= 2,3,4,5,…,n-1 y $f^n(x_0) \neq 0$ entonces se cumpla lo siguiente :

1. Si n es par

   si $f^n(x_0) > 0$ el valor de $f(x_0)$ es un mínimo de la función en $(x; f(x_0))$ y si $f^n(x_0) < 0$ el valor de $f(x_0)$ es un máximo de la función

2. Si n es impar

   si $f^n(x_0) > 0$ la función $f(x)$ presenta un punto de inflexión de primera clase en $(x; f(x_0))$ es decir su concavidad se dirige hacia abajo a la izquierda de x_0 y hacia arriba a su derecha

   si $f^n(x_0) < 0$ la función $f(x)$ presenta un punto de inflexión de segunda clase en $(x; f(x_0))$ es decir su concavidad se dirige hacia arriba a la izquierda de x_0 y hacia abajo a su derecha

2.13 Algoritmo de la derivada enésima

El criterio de la derivada enésima plantea lo siguiente

Ingresar una función $f(x)$

Obtener $f^{´}(x)\ y\ f^{´´}(x)$

Resolver las ecuaciones $f^{´}(x) = 0\ y\ f^{´´}(x) = 0$

Obtener los puntos críticos x_o

Mientras que las n $(n \geq 2)$ primeras derivadas son continuas en un intervalo $\langle a, b \rangle$ y $f^{(p)}(x_o) = 0$ para p=2,3,4,5,…,n-1 hacer

1° Si n es par y $f^n(x_0) > 0$ entonces existe un mínimo en $(x_0, f(x_0))$

Sino existe un máximo en $(x_0, f(x_0))$

2° Si n es impar y $f^n(x_0) > 0$ entonces existe un punto de inflexión de primera especie o clase en $(x_0, f(x_0))$

Sino

existe un punto de inflexión de segunda clase en $(x_0, f(x_0))$

2.14 Desarrollo de Software

En la ingeniería del software el desarrollo de un nuevo producto se enmarca en un conjunto de fases que explica cómo evoluciona o avanza dicho software y cuánto tiempo puede requerir. Es común escuchar sobre versiones alfa (donde se añade características nuevas) y beta o de depuración de errores. Lo cierto es que para alcanzar los estadíos de desarrollo se necesita seguir alguna metodología como XP(programación extrema) , RUP (Proceso dunificado de desarrollo) , experiencia del usuario ,entre otras .

2.15 Software orientado a web

En la actualidad es común la referencia a la "nube" donde **cloud computing** es una de las referencias técnicas (en idioma inglés) lo que significa que muchas de las actividades se desarrollan sobre la nube.

La tendencia actual se orienta a que las aplicaciones (software de aplicación) se desarrollen sobre la web donde será un servidor el encargado de realizar la funcionalidad del sistema que hemos implementado a través de un programa que manejará el usuario con el navegador web (Internet Explorer, Firefox, Chrome) de su ordenador.

Dámaso Velásquez en su artículo publicado opina sobre las aplicaciones en web:

> *"La* ***principal*** *ventaja será la disponibilidad de la aplicación a través de dispositivos que tengan un navegador web: ordenadores, teléfonos móviles, tablets, etc. De esta forma un escenario posible podría ser un comercial de una empresa que cierra un pedido en el domicilio de su cliente y a través de una tablet deja realizado el mismo y confirmado con el cliente un plazo de entrega. En ese caso el equipo que tramite los pedidos ubicado en la empresa tendrá constancia del pedido en el momento y podrá tramitarlo rápidamente.*
> *Otra ventaja muy importante será la gestión de actualizaciones que, con actualizar la aplicación del servidor, todos los usuarios la tendrán en el momento. Sólo será necesario poner la aplicación en modo mantenimiento para que no haya ningún usuario conectado en ese momento (y no pierda datos) y realizar la mejora. Este tipo de actualizaciones puede hacerse en un horario fuera del horario de oficina de la empresa"*
> Dámaso Velázquez Álvarez (30-11-2012). Recuperado de https://webprogramacion.com/356/blog-informatica-tecnologia/aplicaciones-web-vs-aplicaciones-de-escritorio.aspx.

2.16 Calidad del Software

Una definición muy reconocida es la de Robert Pressman en su libro "Concordancia con los requisitos funcionales explícitamente establecidos con los estándares de desarrollo explícitamente documentados y con las características implícitas que se espera de todo software desarrollado profesionalmente" (Pressman R.S., 2012)

2.17 Lenguaje de Programación Java

Es un lenguaje formal diseñado para expresar procesos que pueden ser llevados a cabo por máquinas como las computadoras para lo cual dispone de un conjunto de instrucciones y sentencias donde cada lenguaje define sus propias reglas de sintaxis. En la evolución de los lenguajes de programación debemos destacar la programación, en particular ha cambiado de la forma estructurada a la programación basada en objetos propiciando que los propios lenguajes como C, C++, Visual Basic y Java entre otros busquen generar nuevas versiones. No se descarta el mundo web y la programación móvil que actualmente también requiere de lenguajes tales como PHP, HTML y Android; esto solamente nos da un panorama de la variedad de lenguajes y el entorno para el cual son diseñados.

En particular el lenguaje Java se vislumbra como el más importante no solo por la cantidad de usuarios y/o programadores sino por la potencialidad de su uso en distintas plataformas bajo el slogan "escríbelo una vez, ejecútalo en cualquier lugar", además que su sintaxis proviene en gran parte de lenguajes como C y C++ Pueden usarse para crear programas que controlen el comportamiento físico y lógico de una máquina, para expresar algoritmos con precisión, o como modo de comunicación humana

2.18 Entorno de programación Java

Java al igual que otros lenguajes requiere de un entorno de desarrollo integrado (IDE) que facilite el desarrollo de software al programador, donde destaca NetBeans

> "NetBeans es un entorno de desarrollo integrado libre, hecho principalmente para el lenguaje de programación Java, PHP. Existe además un número importante de módulos para extenderlo. NetBeans IDE es un producto libre y gratuito sin restricciones de uso.
>
> Es un proyecto de código abierto de gran éxito con una gran base de usuarios, una comunidad en constante crecimiento. Sun MicroSystems fundó el proyecto de código abierto NetBeans en junio de 2000 y continúa siendo el patrocinador principal de los proyectos"
>
> ¿Que es NetBeans? (s.f) en https://netbeans.org/index_es.html Recuperado el 20 de enero de 2017.

Asimismo

> "La plataforma NetBeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados *módulos*. Un módulo es un archivo Java que contiene clases de java escritas para

interactuar con las APIs de NetBeans y un archivo especial (manifest file) que lo identifica como módulo. Las aplicaciones construidas a partir de módulos pueden ser extendidas agregándole nuevos módulos. Debido a que los módulos pueden ser desarrollados independientemente, las aplicaciones basadas en la plataforma NetBeans pueden ser extendidas fácilmente por otros desarrolladores de software”

NetBeans (s.f) en Wikipedia Recuperado el 15 de Enero de 2017 de https://es.wikipedia.org/wiki/NetBeans .

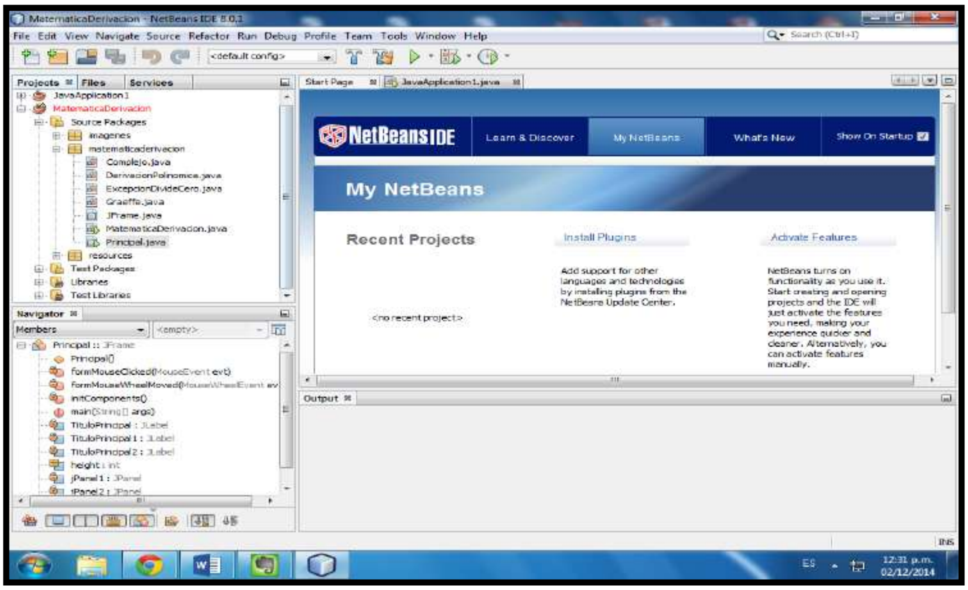


Figura 13: Entorno NetBeans
Fuente: Net Beans.org

.

CAPÍTULO III

ESTADO DEL ARTE

CAPÍTULO III

ESTADO DEL ARTE

La aplicación de la derivada de una función a diversas situaciones del quehacer humano es un tema que despertó el interés en el mundo de la programación pues se busca implementar los algoritmos existentes, teorías y/o conceptos matemáticos en productos de software de aplicación.

3.1 Antecedentes del Problema

La bibliografía revisada no encuentra temas precedentes por lo que constituye una investigación inédita. Al respecto, existen referencias de trabajos que presentan un esfuerzo por abordar los conceptos de la derivada en la optimización desde un punto de vista educativo.

3.1.1 Antecedentes Internacionales

- **Autor**: José Cortés Zavala (Cortez, 2002)
  Título: "Software para la enseñanza de la derivada"
  Tipo: Artículos del Seminario Nacional de Reflexiones sobre la Enseñanza del Precálculo y del Cálculo –México

  Resumen

  José Cortés Zavala (Cortez, 2002) expone la parte inicial de una alternativa, que está en proceso de desarrollo, para la enseñanza del concepto de la derivada a partir de analizar la razón de cambio que se obtiene al tomar dos puntos de una función dada. Para ello se construyó un software educativo en el que se presentan diferentes registros semióticos de representación.

  Como punto de arranque se inicia con el registro de representación numérico, a través de introducir ideas relacionadas con progresiones aritméticas y posteriormente se intercala el registro semiótico de representación gráfico.

  Aporte: La búsqueda de un nuevo enfoque para enseñanza de la derivada puede trasladarse hacia la construcción de un software. Es cierto que ya existe sobre el particular, pero la idea de un nuevo enfoque enriquece y produce la mejora del tratamiento de información

- **Autor**: María Teresa Dávila Araiza (Dávila, 2010)
  Título: “La Derivada a Partir de Problemas de Optimización en Ambientes Dinámicos creados con GeoGebra”.
  Tipo: Tesis de Maestría. Universidad de Sonora. Hermosillo –México

  Resumen
  María Teresa Dávila Araiza (Dávila,2010) formula una propuesta de desarrollo docente, que consiste en el diseño de una serie de actividades didácticas, que a su vez tiene como objetivo “ impulsar en los estudiantes de ingeniería la construcción del significado de la derivada a partir de la pendiente de la recta tangente, así como de otros objetos matemáticos del cálculo diferencial, en ambientes dinámicos virtuales y a partir de problemas de optimización” (Dávila,2010 : 175). Este objetivo constituye un esfuerzo relevante pero dista de nuestra propuesta en el sentido de la producción de software.

  Aporte
  El uso del software orientado e intencionado para generar en los estudiantes el significado de la derivada en la resolución de problemas de optimización constituye un punto interesante en la búsqueda de nuevos caminos para la obtención de los puntos críticos de una función.

3.1.2 Antecedentes Nacionales

- **Autor**: Edgar Cruz Ruiz Lizama (Ruiz, 2006)
  Título: “IntegraLAB: Un software para integración de funciones y solución de ecuaciones diferenciales por métodos numéricos*”*
  Tipo*:* Tesis de Maestría. Pontificia Universidad Católica del Perú

  Resumen
  Edgar Cruz Ruiz Lizama (Ruiz,2006) expone “El diseño e implementación del software IntegraLAB perfilado como una herramienta para resolver problemas de integración de funciones y solución de ecuaciones diferenciales ordinarias aplicando métodos numéricos. Para su construcción ha tenido en cuenta los principios de la ingeniería de software y utiliza el lenguaje Java para su codificación implementando con él, herramientas y técnicas proporcionadas por la teoría de compiladores, los algoritmos, las estructuras de datos y los métodos numéricos. Asimismo, ha escrito un ambiente que permite la gráfica o ploteo de funciones. El software ha sido probado en relación a sus resultados usando

funciones cuyos resultados son conocidos y presentados en la literatura sobre métodos numéricos y contrastados con los obtenidos o devueltos por IntegraLAB, estableciendo las comparaciones del caso se encuentra precisión y exactitud con los resultados esperados” (Ruiz Lizama, 2006: 1).

Aporte : Sin duda el documento más cercano a la propuesta de creación de nuevo software, aunque está orientado a integrales y ecuaciones diferenciales , enfatiza la prueba de software usando funciones referentes o funciones testigo lo cual ha sido considerado en la presente investigación.

3.2 Taxonomía

Título de investigación: Desarrollo de software para la optimización de funciones polinómicas de una variable aplicando el criterio de la derivada enésima en el cálculo de máximos y mínimos

En conconcordancia con ACM (Association for Computing Machinery – Digital Library) la taxonomía del presente proyecto es la siguiente:

Técnica: Algoritmo derivada enésima

Theory of computation/Design and analysis of algorithms/Approximation algorithms analysis. La figura N° 14 ilustra la taxonomía en lo referente a la técnica

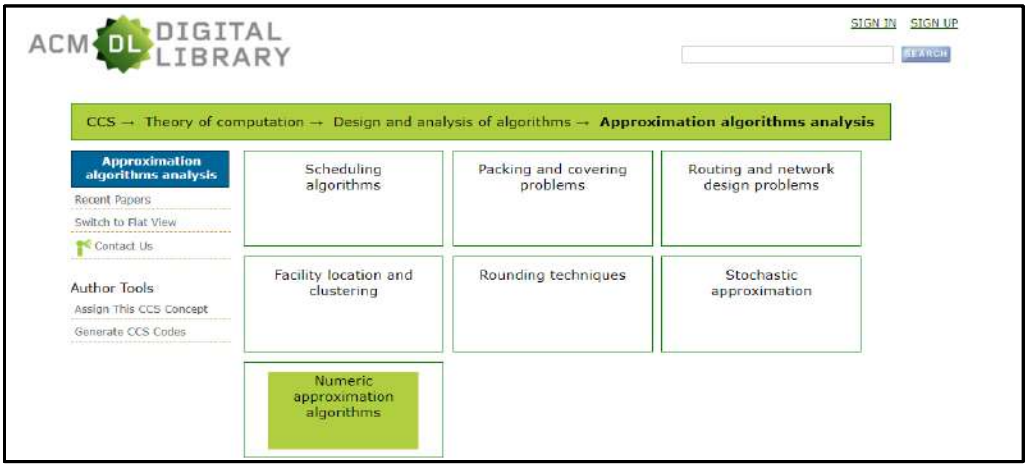


Figura 14 Taxonomía ACM 1
Fuente: ACM Digital Library. Recuperado de
https://dl.acm.org/ccs/ccs.cfm?id=10003636&lid=0.10003752.10003809.10003636

Herramienta: Software orientado a web

CCS/Software and its engineering/Software creation and management/Software development techniques.

La fiigura N° 15 muestra lo referente a la herramienta

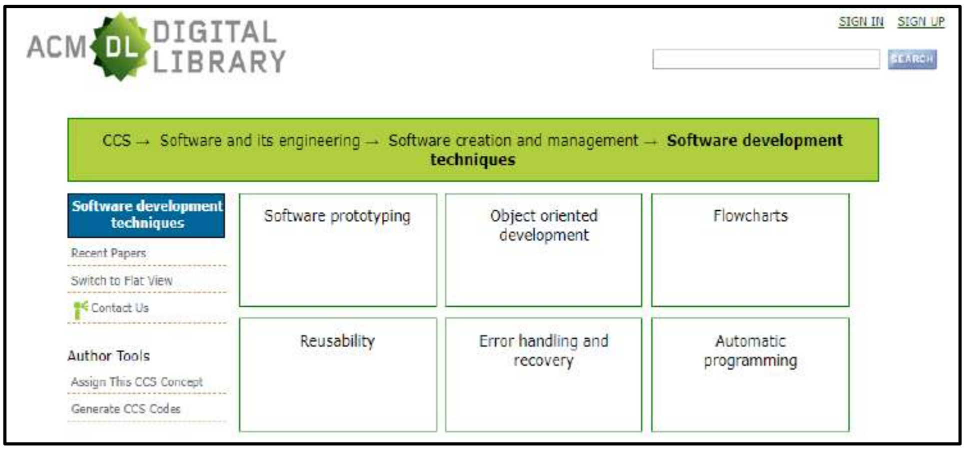


Figura 15 Taxonomía ACM 2
Fuente: ACM Digital Library Recuperado de
https://dl.acm.org/ccs/ccs.cfm?id=10011092&lid=0.10011007.10011074.10011092

Medio: Internet

CCS/Networks/Network types/

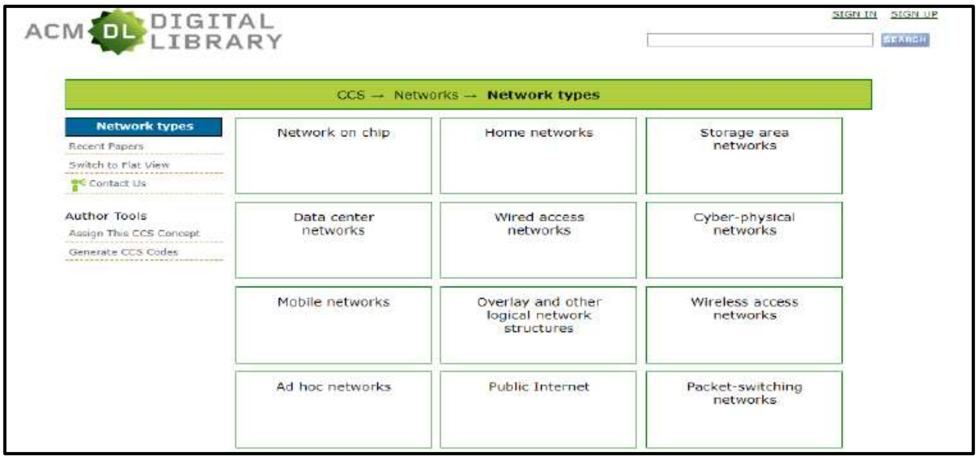


Figura 16 Taxonomía ACM 3
Fuente: ACM Digital Library Recuperado de
https://dl.acm.org/ccs/ccs.cfm?id=10003106&lid=0.10003033.10003106

En relación con IEEE 2017 Versión 1.0 el proyecto corresponde a una taxonomía definida por:

Técnica: **Algoritmo derivada enésima**

Computational and artificial intelligence/ Computers and information processing/ Computer applications/ Scientific computing

Herramienta: **software orientado a web**

Computational and artificial intelligence/ Computers and information processing/ software/ Application software

3.3 Metodología de la revisión bibliográfica

a) Repositorios de tesis

Las fuentes de investigación (Tesis) se realizó a través del buscador de Google Académico tomando en cuenta la búsqueda con palabras clave: derivada enésima, criterio de la derivada enésima, método de Graeffe, teniendo que ser tesis internacionales y nacionales; el año de publicación y la similitud con el problema, en consecuencia, servirán como referencia o modelo para la investigación:

- Google Académico
- Repositorio digital Universidad Técnica del Norte
- Repositorio digital Universidad de Sonora – México
- Repositorio PIRHUA Universidad de Piura
- Repositorio digital de tesis UNMSM
- Repositorio digital de tesis PUCP

Motor de Búsqueda

Se utilizó el buscador de Google Académico (Google Scholar) considerando como parámetros:

- **Inclusión**
  - Tesis publicadas en los años 2000 al 2018.
  - Tesis incorporadas en repositorios académicos.
  - Tesis que contengan palabras claves en su título.
- **Exclusión**
  - Tesis publicada en más de 20 años de antigüedad.
  - Tesis no incorporadas en repositorios académicos.
  - Tesis que no contengan palabras claves en su título.

b) Repositorios de artículos científicos

Adquirir información de artículos científicos usando Google académico permitió explorar webs que puedan otorgar información con alta credibilidad en revistas indexadas considerando la búsqueda con palabras claves de “derivada”, “criterio de la derivada enésima” y “optimización”, “software derivada” dentro del bloque revistas como:

- Scielo
- Redalyc
- DBSpace

- IEEE

Motor de Búsqueda

El buscador de Google Académico ayudó a documentar la tesis usando como parámetros:

- **Inclusión**

- Información publicada en los años 2000 al 2018.
- Corresponda a artículos en revistas indexadas (alta calidad).
- Información que contenga palabras claves en su título.

- **Exclusión**

- Información que no contenga palabras claves en su título.

c) Libros de consulta sobre derivada

Respecto a este punto se seleccionó una variada literatura de autores nacionales y extranjeros donde la antigüedad del texto no constituye un parámetro de exclusión siendo uno de ellos impulsor de la tesis. Por otra parte, han cumplido el propósito de validar el software al contrastarse ejercicios y problemas seleccionados.

3.4 Software

A continuación, una referencia a productos de software en línea que guardan relación con la investigación.

3.4.1 Derive

Derive es un programa para el cálculo matemático avanzado: variables, expresiones algebraicas, ecuaciones, funciones, vectores, matrices, trigonometría. Se utiliza como calculadora científica, y puede representar funciones gráficas en dos y tres dimensiones en varios sistemas coordenados, En la actualidad Texas Instruments ya no comercializa Derive, cuyo desarrollo paso ahora como un nuevo producto el TI-Nspire CAS".

Derive.(s.f) En Matemáticas3con14 Recuperado el 16 de diciembre de 2018 de http://3con14.com/104-%C3%BAtiles/software/68-i-derive-software.html

Derive es un programa matemático que aparte de realizar cálculos numéricos permite que los ususarios puedan generar sus propias funciones ejerciendo principios de programación en su entorno.

En su tesis Tratamiento didáctico de la derivada : Aplicación del programa Derive, Quintana Sanchez (2010) destaca : “El programa incluye algunos archivos llamados de aplicación de gran utilidad para resolver Ecuaciones Diferenciales, Transformada de Laplace, Integrales Elípticas y para Graficar Curvas y Superficies Paramétricas en tres dimensiones” (Quintana Sánchez, 2010, pág. 7). La figura 17 muestra la pantalla inicial o escritorio de Derive 6.0

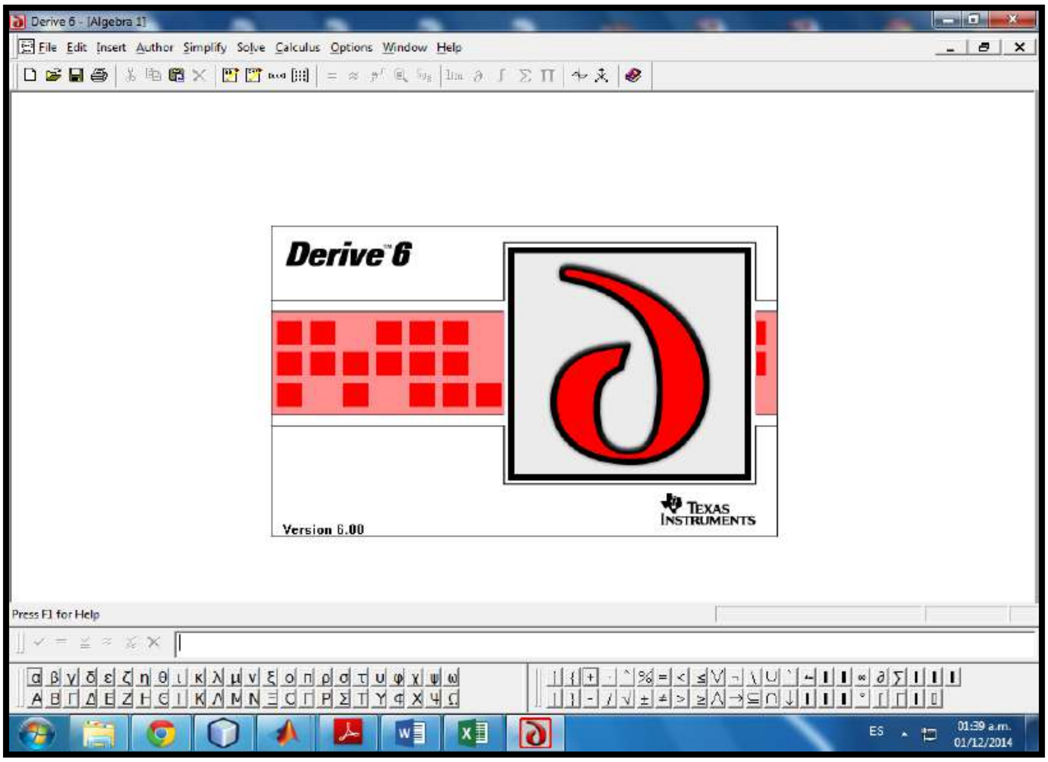


Figura N° 17 Escritorio de Derive
Fuente : Derive

3.4.2 Wolfram Alpha

En su investigación relacionada con el uso del software WolframAlpha en ambientes educativos, Gálvez, Julca y Tineo (2017) describen a WolframAlpha como:
“El primer programa de cálculo simbólico capaz de ejecutarse en diversos sistemas operativos que se escribió en lenguaje C en 1988 y desde entonces, se usa en numerosos campos de la ciencia y la técnica y también ha tenido una buena acogida entre los estudiantes de carreras en las que la matemática es básica para su formación. En el 2009 WolframAlpha fue anunciado por Stephen Wolfram y entró en funcionamiento en mayo del mismo año” (Gálvez, Julca y Tineo, 2017: 36)   . La figura 18 nos ilustra la página oficial de WolframAlpha

Figura N° 18. Presentación de WolframAlpha
Fuente: WolframAlpha

Asimismo Gálvez, Julca y Tineo (2017) reconocen en WolframAlpha las siguientes características:

"Una **calculadora de tipo numérico**

porque tiene alrededor de 800 funciones predefinidas y alta preciso en sus cálculos.

Un paquete de subrutinas para cálculo matemático

Se pueden usar funciones o de procedimientos.

Una calculadora simbólica

Permite el uso de expresiones simbólicas.

Una herramienta para cálculo simbólico y gráficos

Puedes derivar e integrar funciones, resolver ecuaciones diferenciales, calcular límites, manipular series de potencias, graficar en 2D y 3D, realizar todo tipo de animación.

Un lenguaje de programación, se puede realizar programación

Un sistema de apoyo a otros programas. Puedes comunicar con *Wolfram* desde otros programas y pedirles tareas que realizará y después enviará los resultados" (Gálvez, Julca y Tineo, 2017: 37)

3.4.3 MATLAB

MATLAB, abreviatura de MATrix LABoratory es un software matemático que tiene un lenguaje de programación propio (M) y se adapta a plataformas Unix, Windows, Mac OS X y GNU/Linux.

"Según, (Little & Moler, 1 994) en la aplicación de la informática educativa [www.mathwoks.com] dice: "MATLAB es un lenguaje de alto nivel y un entorno interactivo para cálculo numérico, visualización y programación. Usando MATLAB, puede analizar los datos, desarrollo de algoritmos, y crear modelos y aplicaciones. El lenguaje, las herramientas y funciones incorporadas de matemáticas le permiten explorar múltiples enfoques y llegar a una solución más rápida que con las hojas de cálculo o lenguajes de programación tradicionales, tales como C / C + + o Java". (Como se cita en Carapaz, José, 2014).

En los últimos años ha aumentado el número de prestaciones, como la de programar directamente procesadores digitales de señal o crear código VHDL"

La figura 19 presenta la pantalla inicial de MATLAB

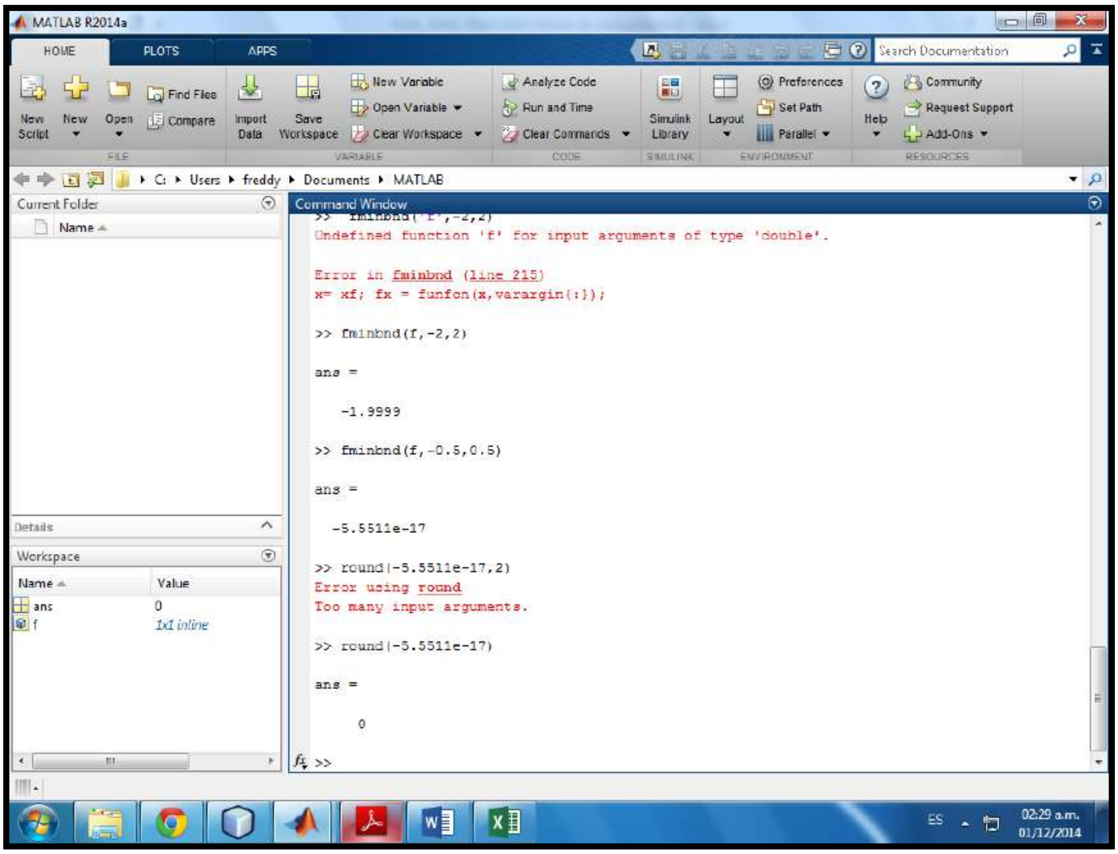


Figura N° 19. Presentación de MATLAB
Fuente: software MATLAB

En el libro Método numéricos con MATLAB su autor Efracio Asís López (2010) señala: “Matlab cuenta con paquetes especailizados orientados a ingenieros, científicos y otros profesionales técnicos. Entre estos paquetes se destaca sobre Splines, redes neuronales, lógica difusa, optimización” (Asís López, 2010: 261).

3.4.4 GeoGebra

Bello (2013) indica que el software GeoGebra

“Es un software de geometría dinámica aplicado en todos los niveles de educación y dirigido tanto para profesores como para alumnos; este programa fue creado por los esposos Markus y Judith Hohenwarter, quienes trabajaron con este software desde el año 2001 en la Universidad de Salzburgo y posteriormente en la Universidad de Atlantic, Florida, Estados Unidos. (p. 30)” (como es citado en Bermeo, 2017)

En Geogebra.org se indica que “GeoGebra es un software de matemáticas para todo nivel educativo. Reúne dinámicamente geometría, álgebra, estadística y cálculo en registros gráficos, de análisis y de organización en hojas de cálculo. GeoGebra, con su libre agilidad de uso, congrega a una comunidad vital y en crecimiento. En todo el mundo, millones de entusiastas lo adoptan y comparten diseños y aplicaciones de GeoGebra. Dinamiza el estudio. Armonizando lo experimental y lo conceptual para experimentar una organización didáctica y disciplinar que cruza matemática, ciencias, ingeniería y tecnología (STEM: Science Technology Engineering & Mathematics)”

¿Qué es GeoGebra? (s.f.) En Geogebra.org recuperado el 20 de enero de 2018 de https://www.geogebra.org/about

Castellanos (2010) añade sobre lo que permite GeoGebra

“Construir en forma precisa y rápida usando los componentes básicos de la geometría. Razonar y comprender a cerca de las relaciones geométricas entre diferentes objetos. Controlar el aspecto gráfico de una figura, usando simplemente el mouse. Ejecutar cálculos de medida. Manipular las figuras geométricas y observar las semejanzas y diferencias entre ellas. Repetir las construcciones las veces que ellos necesiten hacer, es decir observar los pasos que se siguieron para realizarlas. Hacer las conjeturas respectivas de las construcciones realizadas. Imprimir las construcciones. (p. 46)” (como es citado en Bermeo, 2017)

La pantalla incial o escritrio de GeoGebra se observa en la figura 20

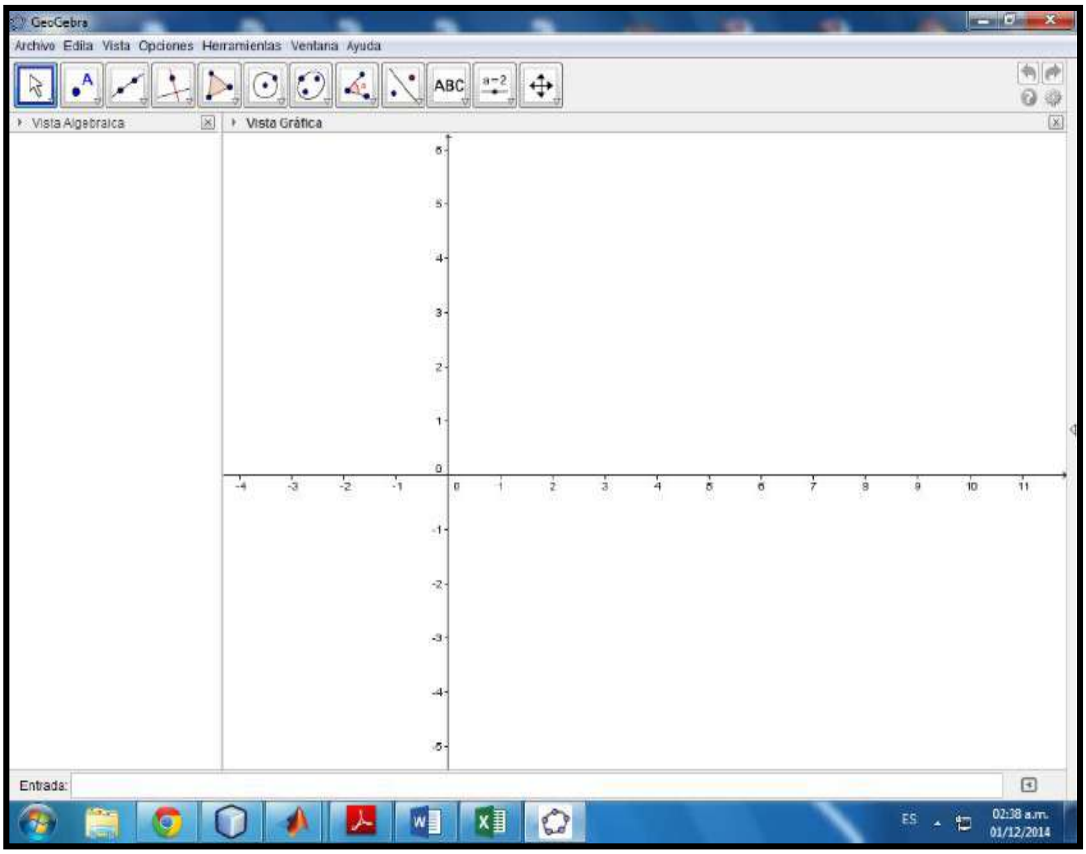


Figura N° 20: Presentación de GeoGebra
Fuente: software GeoGebra

3.4.5 Symbolab

"Symbolab es un tutor online que resuelve problemas matemáticos. Por defecto, da tanto el resultado como una explicación detallada del proceso. No se trata de una sola calculadora, sino una página con acceso a varias aplicaciones web para resolver problemas distintos: uno para cálculos algebraicos, otro para ecuaciones diferenciales, series de Taylor"

8 Calculadoras matemáticas online gratuitas (09-06-2016) En ROBOLOGS Recuperado el 28 de noviembre de 2018 de https://robologs.net/2016/06/09/8-calculadoras-matematicas-on-line-gratuitas/

Symbolab fue fundada a fines de 2011 por tres israelíes , Michal Avny (CEO), Adam Arnon (Jefe Científico) y Lev Alyshayev (CTO). Crearon un motor que permite al usuario ingresar una ecuación y encontrar la respuesta si existe. Luego agregaron la capacidad de realizar el cálculo y enumerar todos los pasos para que el usuario entienda cómo se llega a la respuesta.

Eqsquest comenzó en 2011, como un motor de búsqueda semántica matemática. El énfasis se desvió gradualmente hacia soluciones integrales, paso a paso, para problemas matemáticos de nivel secundario y universitario. Se basa en algoritmos de aprendizaje automático para realizar los aspectos de búsqueda y solución del motor.

Symbolab. (s.f.) En Wikipedia Recuperado el 28 de noviembre de 2018 de https://en.wikipedia.org/wiki/Symbolab

La figura N° 21 muestra la pagina oficial de Symbolab

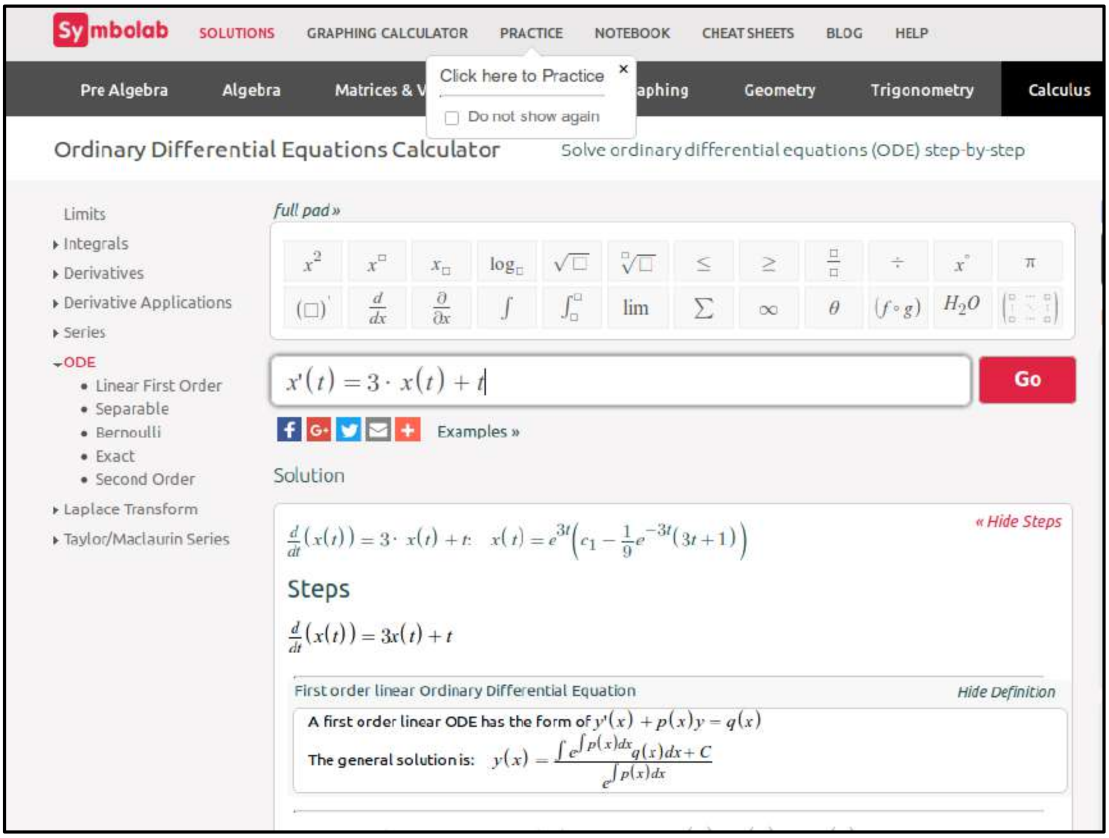


Figura N° 21: Presentación de Symbolab
Fuente: Symbolab

CAPÍTULO IV

METODOLOGÍA   Y RESULTADOS

CAPÍTULO IV

METODOLOGÍA   Y RESULTADOS

4.1. Tipo de Investigación

Hernández, R., Fernández, C. y Baptista, M. coautores del libro Metodología de la investigación formulan una clasificación de los tipos de investigación. Luego de analizar las posibles formas de investigar se argumenta que será **explicativa** debido a que se busca la relación causa-efecto que se genera al utilizar un proceso manual y el uso de un software para alcanzar extremos relativos y puntos de inflexión de una función.

La investigación es de tipo **tecnológica** pues tiene como propósito el desarrollo de software basado en una nueva forma de plantear la obtención de los extremos locales y de los puntos de inflexión lo que constituye una reconfiguración de métodos matemáticos concibiendo la idea de mejora respecto a lo existente

4.2 Nivel y diseño de la investigación

El nivel de la investigación es **aplicado**. El desarrollo de nuevo software propone una alternativa para calcular máximos y mínimos, así como un nuevo camino para la enseñanza de las derivadas y sus aplicaciones con el empleo de herramientas tecnológicas, siendo el desafío mayor la mejora del producto frente a los que ya existen en el medio.

El diseño seleccionado es de tipo **preexperimental** porque representa un estudio de caso con una sola medición. Una función polinómica es insertada dentro del software y se espera observar el resultado de su acción.

Este diseño podría diagramarse de la siguiente manera:

G X 0

4.3 Método y Técnicas

El método es el camino   que guia la investigación. Ell **método cientifico** es el elegido, porque el sustento de la investigación se define sobre una teoria, demostrada con la rigurosidad de la ciencia matemática merced a procesos inductivo -deductivo que se describen en el criterio de la derivada enesima y que ahora se debe traducir en el desarrollo de software

La técnica es el conjunto de instrumentos y medios que asisten al método. En el desarrollo del software se ha utilizado una **metodologia ágil** donde el programador y el experto en el tema han avanzado en forma conjunta en el desarrollo , realizando mejoras , haciendo pruebas preliminares y lo principal si el software está encaminado en el objetivo trazado.

4.4 Población y Muestra

El universo que conforma la población está conformado por todas las funciones polinómicas de segundo, tercer, cuarto y quinto grado. No se consideran las funciones lineales dado que su forma gráfica advierte una función estrictamente creciente o decreciente sin curvaturas.

Para calcular el tamaño de la muestra se utilizará el tipo de muestreo estratificado con igual posibilidad de elección para las funciones polinómicas de segundo, tercer, cuarto y quinto grado.

En la investigación se aborda una población infinita por lo que para calcular el tamaño de la muestra se usa la fórmula para cálculo de la muestra con población infinita, con un nivel de confianza (Z) del 90% y un margen de error (d) del 10%.

En el libro Metodología de la Investigación (Hernandez R., Fernandez C. y Baptista M) en el capítulo 8 página 175 los autores definen:

> *“Cuando no tenemos marcos de muestreo previos, usamos un porcentaje estimado de 50% (que es la opción automática que brinda STATS®, es decir, asumimos que “p” y “q” serán de 50% — igual probabilidad — ó 0,50 — en términos de proporciones—, y que resulta lo más común, particularmente cuando seleccionamos por vez primera una muestra en una población)”.*

En consecuencia, la probabilidad de que el software aplicado a una función devuelva el valor correcto será de 50%.

La fórmula aplicada es la siguiente:

$$n = \frac{Z_{\alpha}^{2} \times p \times q}{d^{2}}$$

En donde

Z = nivel de confianza 90% =1,645 (obtenido de la tabla de distribución normal)

p= probabilidad de éxito = 0,5

q = probabilidad de fracaso = 0,5

d = precisión (error máximo admisible en términos de proporción) = 0,10

Reemplazando en la fórmula $n = \frac{1{,}645^2 \times 0{,}5 \times 0{,}5}{0{,}1^2}$

De donde n= 67,65 equivalente a 68 para el tamaño muestral

Para corroborar nuestro cálculo utilizamos una calculadora de tamaño de muestra online de la pagina web https://es.surveymonkey.com/mp/sample-size-calculator/ donde se obtiene 68 unidades para el tamaño de muestra. La figura 22 nos muestra el cálculo del tamaño de muestra

Figura 22 Cálculo de tamaño de muestra
Fuente: surveymonkey

Por tanto, el tamaño de muestra es igual 68 que se distribuirá de la siguiente forma un 20% será analizado con funciones seleccionadas de libros de consulta sobre el tema lo que equivale a 14 funciones testigo a las que se denomina casos. El 80% restante equivalente a 54 será distribuido con funciones aleatorias en base a las formas posibles (completa e incompleta) que se ha identificado para el grado de cada polinomio referente de la función. Por ejemplo, una función cuadrática tiene cuatro formas posibles: $f(x) = ax^2$; $f(x) = ax^2 + bx$; $f(x) = ax^2 + c$; $f(x) = ax^2 + bx + c$. Una función cúbica tiene 8 formas posibles, las funciones de cuarto grado tienen 15 formas posibles y las funciones de quinto grado presenta 27 posibilidades lo que nos indica un total de 54 funciones los que serán contrastados con otros software online: WolframAlpha, GeoGebra y Symbolab. En el cuadro N° 4 se muestra con detalle la distribución para la muestra

Cuadro N° 4 Distribución de la muestra

Estratos	Formas posibles (P)	Cantidad de Funciones
Casos	Funciones consultadas de libros	14
Funciones testigo	Función polinómica segundo grado	4
	Función polinómica tercer grado	8
	Función polinómica cuarto grado	15
	Función polinómica quinto grado	27
Total		68

Fuente: Elaboración propia

4.5 Unidad de Análisis

La unidad de análisis se refiere al objeto de estudio en la investigación sobre la cual se realiza las observaciones. En la presente investigación se analiza una función polinómica de segundo, tercer, cuarto y de quinto grado

4.6 El software DERIN

4.6.1 Características del usuario

DERIN es un software orientado a web con una finalidad específica, que está enfocado para dar soporte en la resolución de problemas de cálculo referido a la obtención de extremos relativos: máximos y mínimo y de puntos de inflexión. Por tal motivo los usuarios del software deberán tener conocimiento sobre la derivada y sus aplicaciones, así como en la formulación adecuada de la función que resuelve un problema y la capacidad para analizar e interpretar en forma correcta la solución que entregue el software.

4.6.2 Descripción del software

El software denominado "DERIN" permite calcular los extremos locales y puntos de inflexión utilizando el criterio de la enésima derivada y está desarrollado en lenguaje de programación Java en un entorno de trabajo IDE Net Beans.

Luego de su ejecución se observa la siguiente interfaz donde en la parte superior izquierda se selecciona el grado de la función

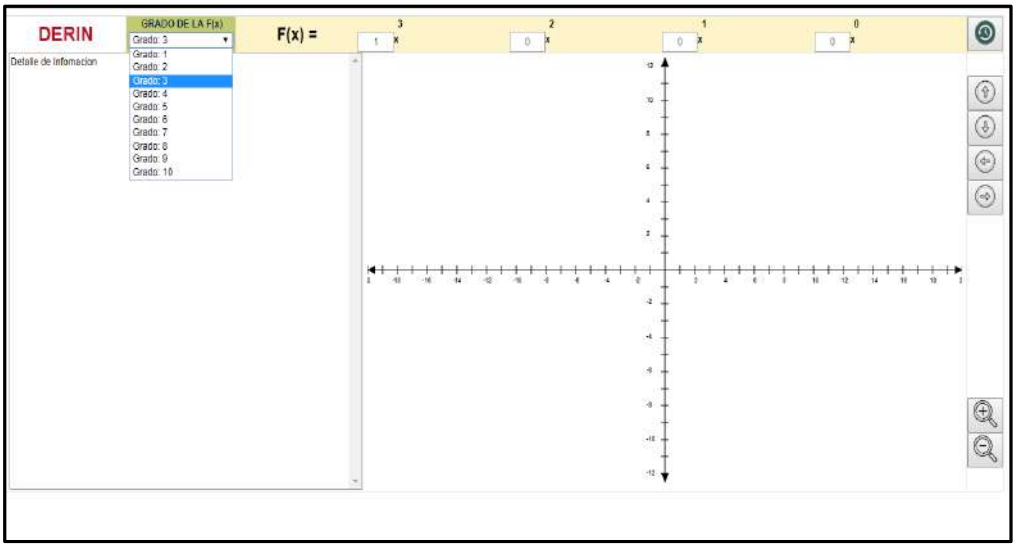


Figura 23 Pantalla Principal del software DERIN
Fuente: Elaboración propia

A continuación, ingresa los coeficientes de la función polinómica ordenado y que por defecto tiene coeficientes cero lo cual permite al usuario modificar los coeficientes.

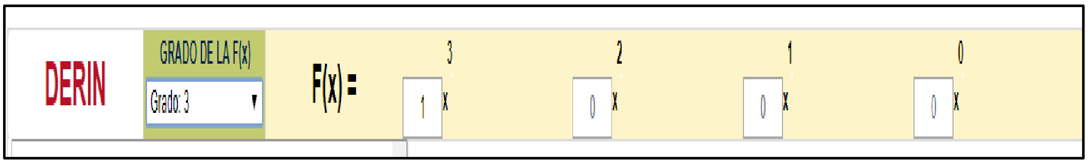


Figura 24: Ingreso de la función polinómica 1
Fuente: Elaboración propia

Luego hace clic en el botón procesar ubicado en la parte superior derecha

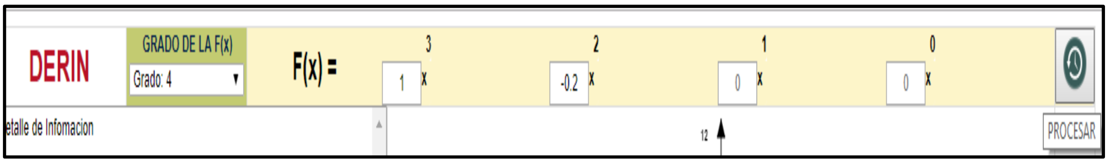


Figura 25: Ingreso de la función polinómica 2
Fuente: Elaboración propia

El resultado de la ejecución se muestra en la siguiente figura donde se observa en la parte izquierda los resultados de las iteraciones y los máximos, mínimos y puntos de inflexión y a su derecha se observa la gráfica respectiva.

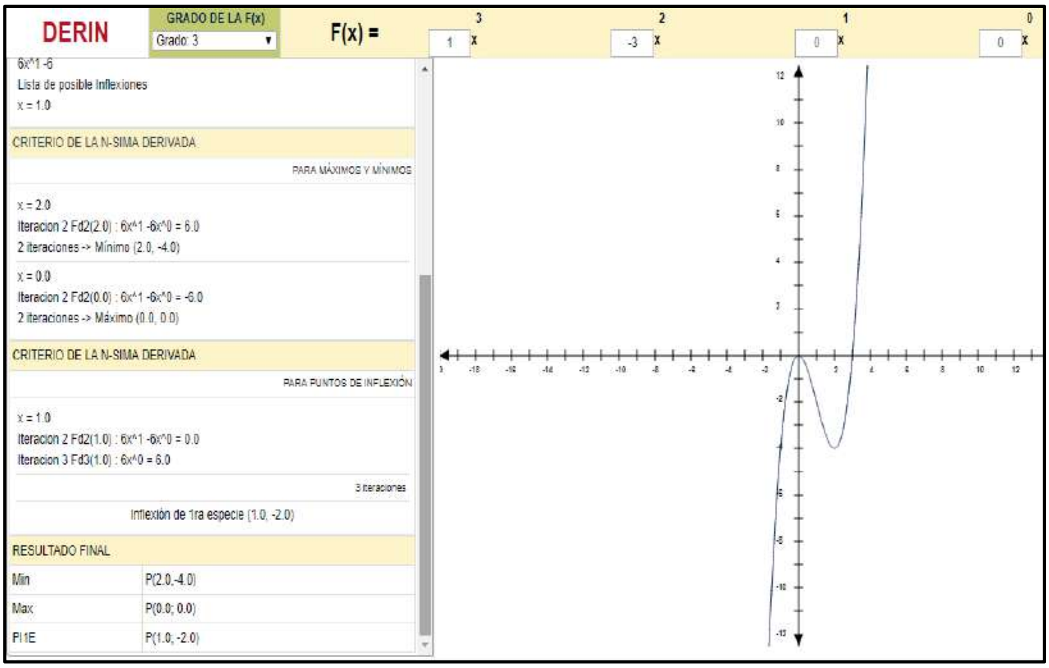


Figura 26: Ejecución del software
Fuente: DERIN

En la parte gráfica se tiene los **botones**, ubicados a la derecha de la gráfica, que permiten modificar el tamaño (zoom) de la gráfica, así como desplazarlo en forma horizontal y vertical

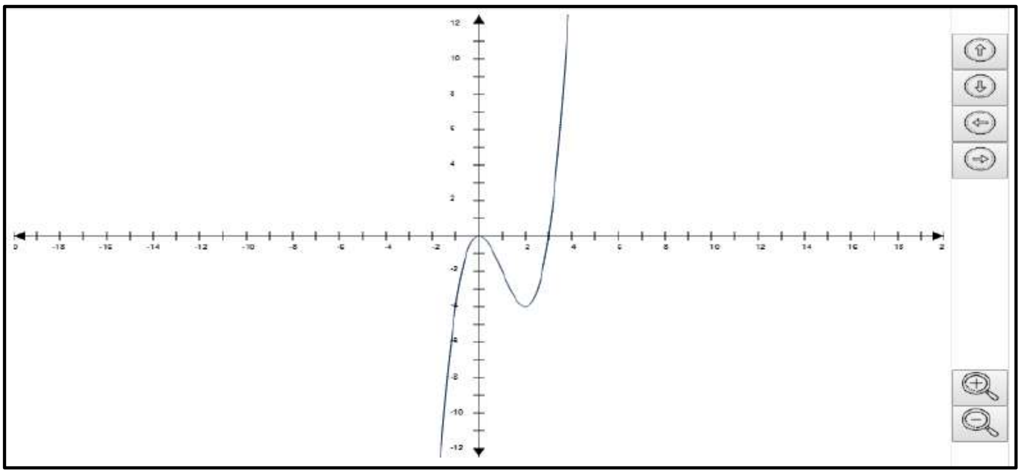


Figura 27: Botones para desplazamiento
Fuente: DERIN

En lo referente al desarrollo tiene seis módulos de los cuales dos están orientados al cálculo de raíces de un polinomio (Graeffe.java y Complejo.java). Un módulo está orientado a la derivación (derivacionpolinomica.java), el módulo Matematicaderivacion está orientado al cálculo de los extremos relativos para lo cual vuelve a utilizar en forma iterativa el módulo derivacionpolinomica.

4.7 Pruebas del software

"La prueba de un programa sólo puede mostrar la presencia de defectos, no su ausencia"

Edsger Wybe Dijkstra

El software DERIN se ha utilizado para calcular los máximos, mínimos y puntos de inflexión de una función polinómica incluyendo su gráfica siendo comparado con las soluciones que ofrecen software como GeoGebra, WolframAlpha y Symbolab. Asimismo, se ha utilizado en casos de optimización seleccionados de textos de reconocidos autores comprarando los resultados de los textos y lo obtenido con DERIN.

4.7.1 Prueba para casos de comparación

Caso 1

Edwin Purcell, Dale Varberg y Steven Rigdon en coautoría del libro Cálculo Diferencial e Integral (Purcell, Varberg y Rigdon, 2007) en el capítulo 3 referente a las aplicaciones de la derivada en el ejemplo 1 de la página 164 proponen:

"Encuentre los valores extremos locales de la función $f(x) = x^2 - 6x + 5$ en $< -\infty; \infty >$"

En el proceso de resolución se concluye que existe un valor mínimo cuando x=3 haciendo notar que corresponde a un mínimo global.

La figura 28 muestra la ejecución del software DERIN respecto a la función $f(x) = x^2 - 6x + 5$ donde se obtiene un mínimo en el punto (3;-4)

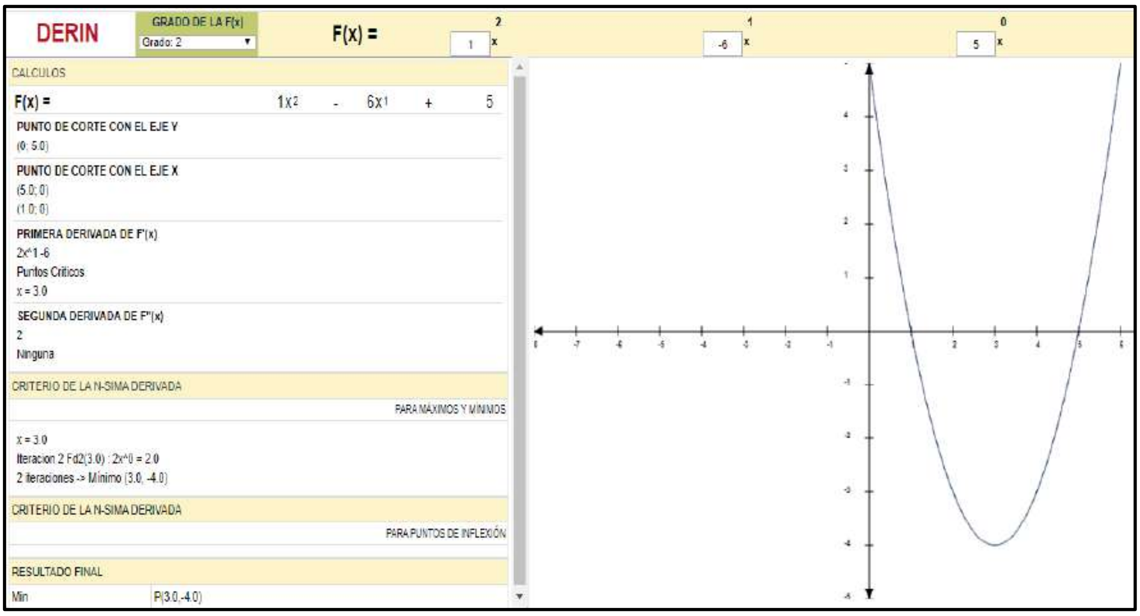


Figura 28: Problema_Purcell
Fuente: DERIN

Caso 2

En la página 230 del libro Matemáticas 1 Cálculo Diferencial (Zill & Wright, 2010) se solicita graficar la función $f(x) = x^3 - 3x^2 - 9x + 2$. En el marco resolutivo el autor deduce los puntos críticos x= -1 y x= 3. Luego se concluye que la función presenta el máximo en x= -1 y el mínimo en x=3. La figura 29 muestra la ejecución del software DERIN donde se corrobora en (3;-25) el mínimo, en (-1;7) el máximo y un punto de inflexión en (1;-9)

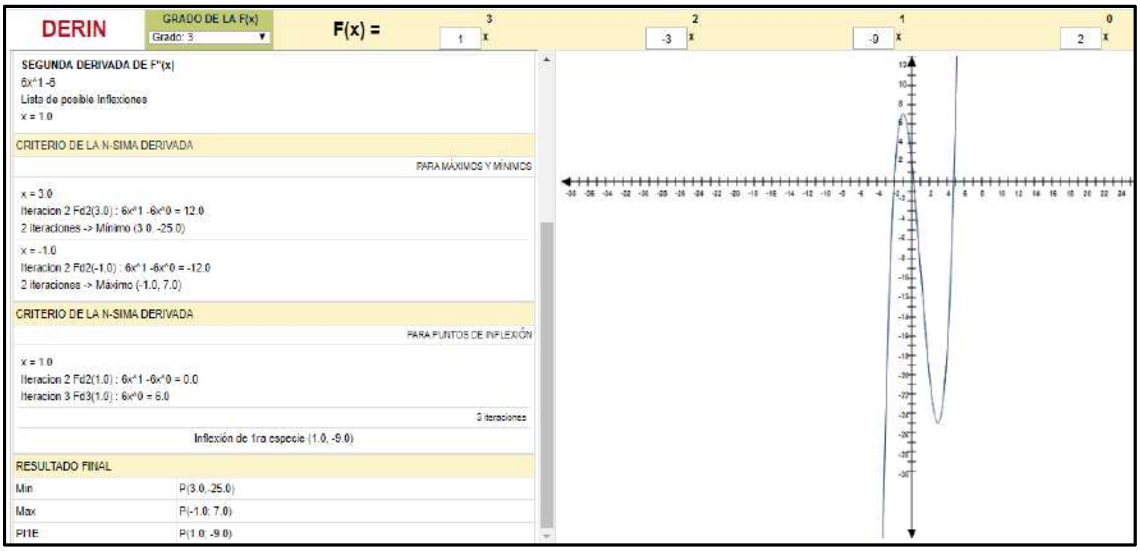


Figura 29: Problema_Zill
Fuente: DERIN

Caso 3

George Thomas B. Jr en la página 270 del libro Cálculo de una variable (Thomas , 2012) presenta el ejemplo del uso de f´ y f´´ para graficar la función $f(x) = x^4 - 4x^3 + 10$ donde el marco resolutivo indica la existencia de un mínimo en el punto (3;-17) y dos puntos de inflexión en (0;10) y (2;-6) cuya grafica se muestra en la figura 30

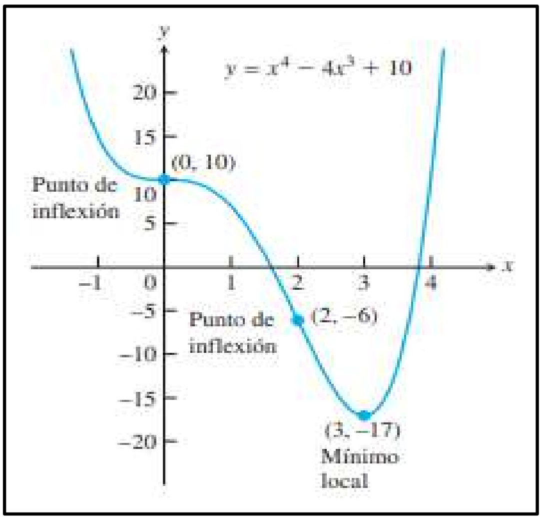


Figura 30: Gráfica de la función Thomas
Fuente: Cálculo de una Variable. George Thomas Capitulo 4 página 271

Al ejecutar el software DERIN se obtuvo como resultado un mínimo en el punto (3,-17) y dos puntos de inflexión en (2;-6) y en (0;10) tal como muestra la figura 31.

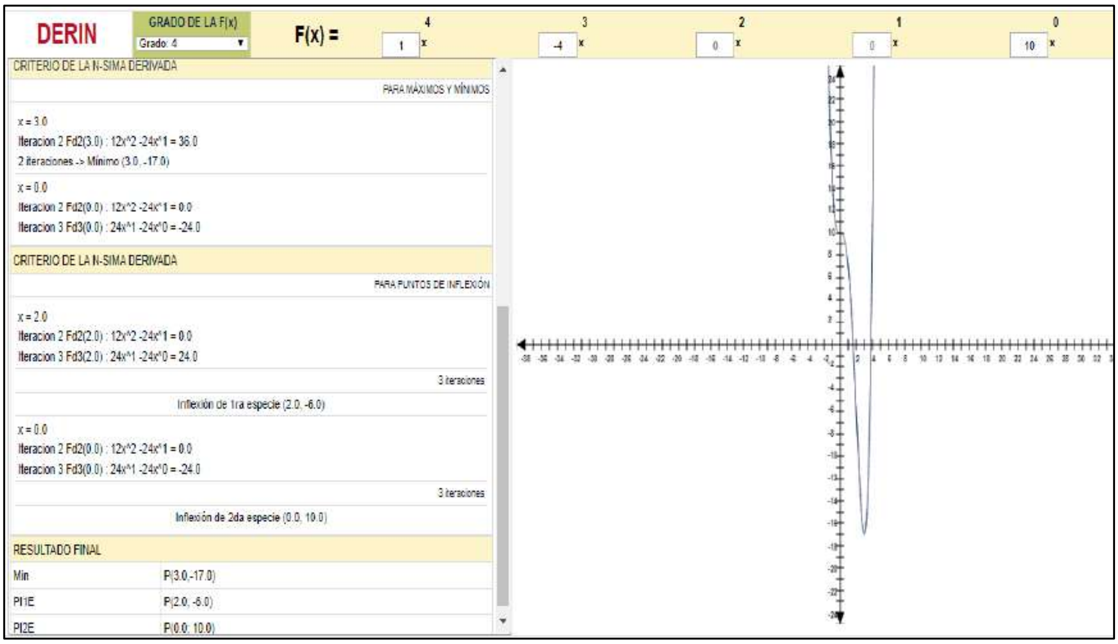


Figura 31: Problema Thomas
Fuente: software DERIN

Caso 4

Nikolai Piskunov en la página 181 del libro Cálculo Tomo I (Piskunov, 1983) en el segundo ejercicio propone analizar los extremos locales de la función $f(x) = 1 - x^4$. La resolución del problema aplica los criterios de la primera y segunda derivada donde x= 0 es un punto crítico, sin embargo se considera necesario retornar a usar el criterio de la primera en forma exclusiva para determinar el crecimiento o decrecimiento de la función con lo que se concluye que x= 0 representa un máximo para la función

Luego de ejecutar el software DERIN (figura 32) se obtuvo como resultado un máximo en el punto (0; 1)

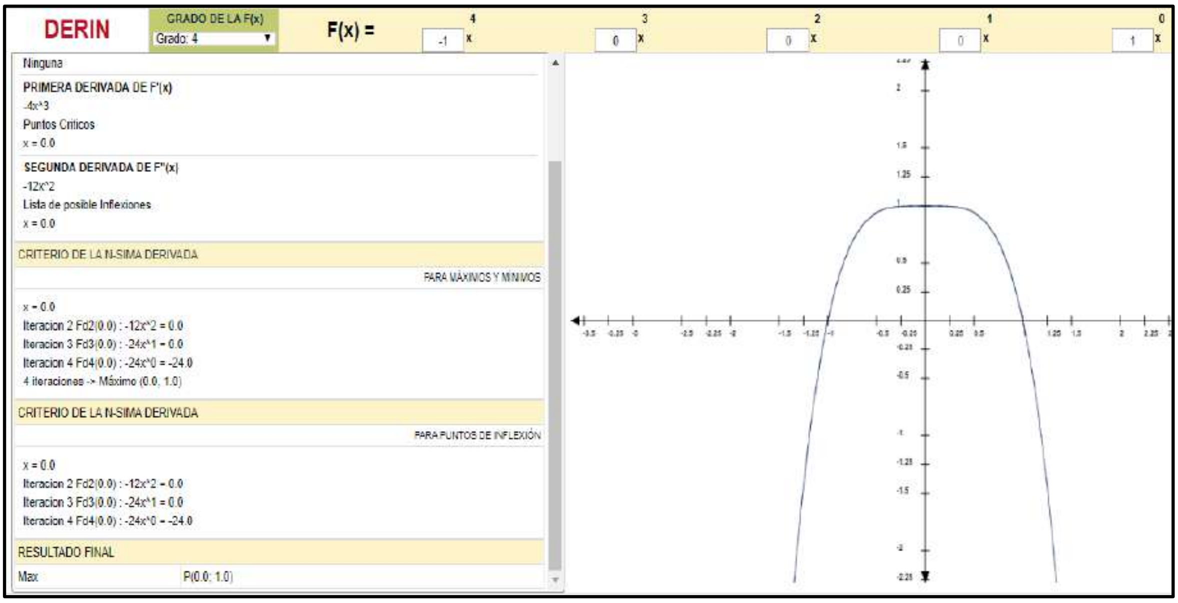


Figura 32: Problema Piskunov
Fuente: DERIN

Caso 5

Armando Venero en su libro Análisis Matemático I en el capitulo 6 página 565 propone "bosquejar la gráfica de la funcion $f(x) = \frac{1}{8}(3x^5 - 20x^3 + 16)$"

La resolucion del problema por el autor detemina un máximo local en un punto donde la abscisa x= -2 , un mínimo local en x=2 , asi como puntos de inflexión cuando $x = \{\sqrt{2}; 0; -\sqrt{2}\}$ tal como muestra la figura 33

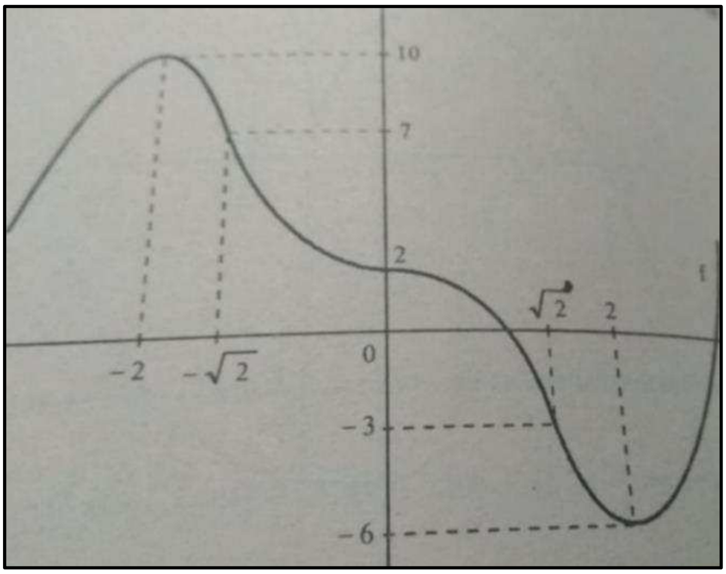


Figura 33: Gráfica de la función Venero
Fuente: Análisis Matemático I. A.Venero , Capitulo 6 página 566

Al ejecutar el software DERIN se ubica en (2;-6) el mínimo , en (2;-10) el máximo así como dos puntos de inflexión de primera especie en (1,4142;-2,9497) y (-1,4142 ; 6,9497) y una inflexión de segunda especie en (0;2) lo cual se muestra en las figuras 34 y 35

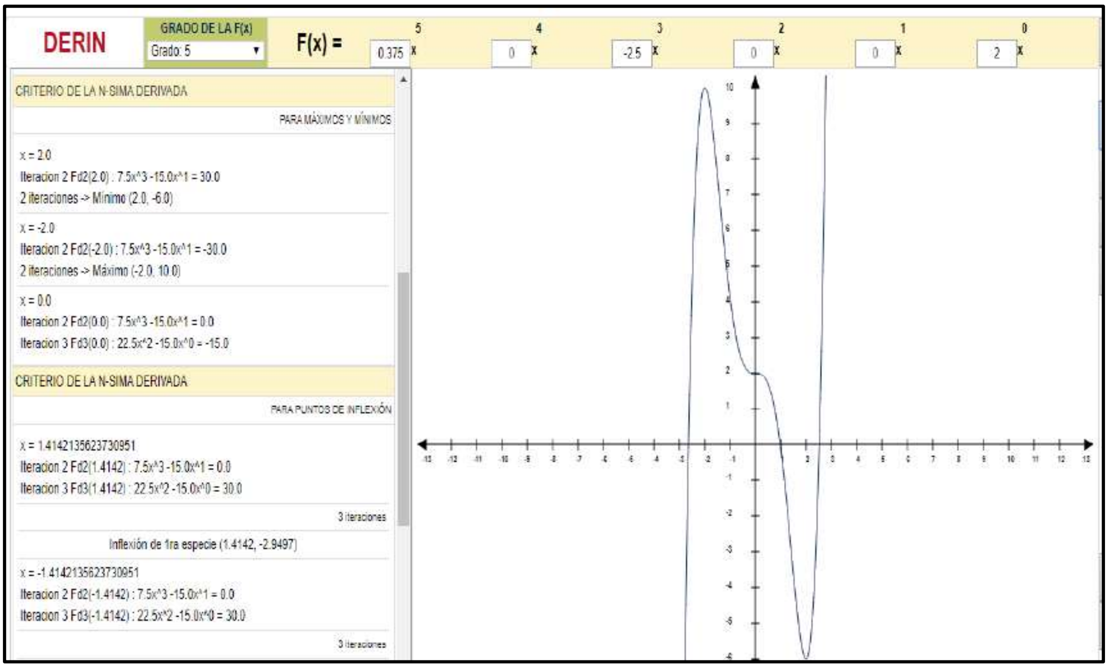


Figura 34: Problema Venero A
Fuente: DERIN

CRITERIO DE LA N-SIMA DERIVADA

PARA PUNTOS DE INFLEXIÓN

x = 1.4142135623730951
Iteracion 2 Fd2(1.4142) : 7.5x^3 -15.0x^1 = 0.0
Iteracion 3 Fd3(1.4142) : 22.5x^2 -15.0x^0 = 30.0

3 iteraciones

Inflexión de 1ra especie (1.4142, -2.9497)

x = -1.4142135623730951
Iteracion 2 Fd2(-1.4142) : 7.5x^3 -15.0x^1 = 0.0
Iteracion 3 Fd3(-1.4142) : 22.5x^2 -15.0x^0 = 30.0

3 iteraciones

Inflexión de 1ra especie (-1.4142, 6.9497)

x = 0.0
Iteracion 2 Fd2(0.0) : 7.5x^3 -15.0x^1 = 0.0
Iteracion 3 Fd3(0.0) : 22.5x^2 -15.0x^0 = -15.0

3 iteraciones

Inflexión de 2da especie (0.0, 2.0)

RESULTADO FINAL	
Min	P(2.0,-6.0)
Max	P(-2.0; 10.0)
PI1E	P(1.4142; -2.9497)
PI1E	P(-1.4142; 6.9497)
PI2E	P(0.0; 2.0)

Figura 35: Problema Venero B
Fuente: DERIN

Caso 6

Eduardo Espinoza Ramos en su libro Análisis Matemático I (Espinoza, 2012) en la pagina 587 propone “construir la gráfica determinando los puntos criticos, .., puntos de inflexión y la dirección de la concavidad de la funcion ”hallar los máximos y mínimos de la función $y = x^3 - 3x^2$

En la resolución del problema el autor concluye que existe un máximo en el punto (0;0) y un mínimo en el punto (2,-4) así como una inflexión en (1;-2)

Al ejecutar el software DERIN se determina igual resultado que lo deducido por el autor como se observa en la figura 36

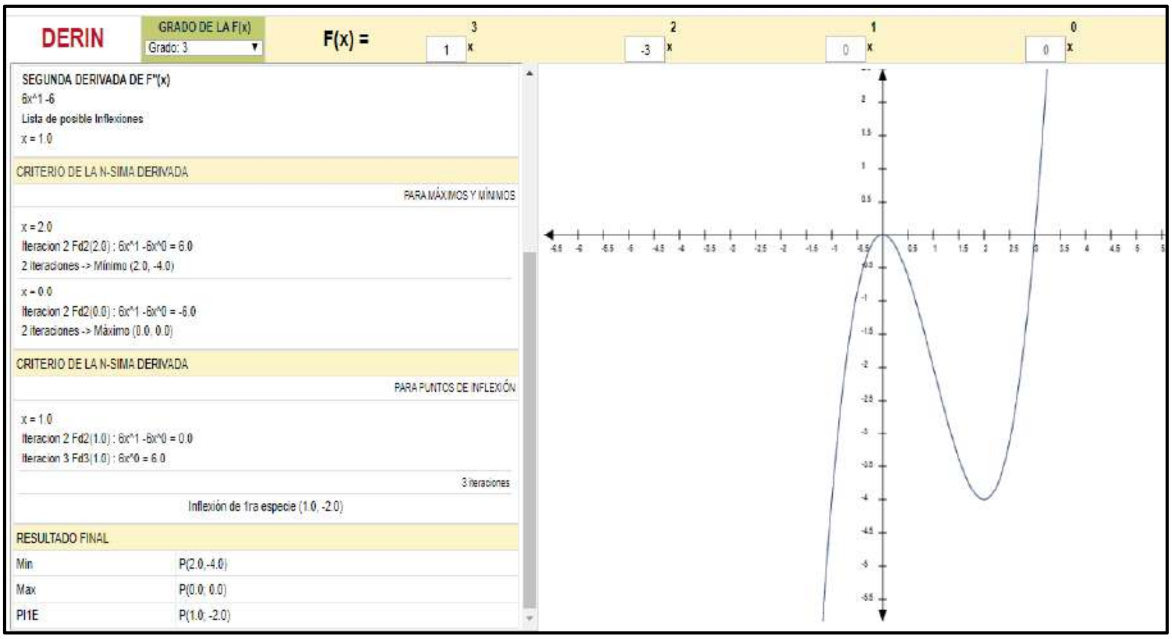


Figura 36: Problema Espinoza
Fuente: DERIN

Caso 7

En el libro de consulta El Cálculo con Geometría Analítica de Louis Leilthold (Leithold, 2012) en la página 227 el ejemplo 1 propone trazar la gráfica de la función $f(x) = x^3 - 6x^2 + 9x + 1$. En el marco de resolucion del problema el autor concluye que la función presenta un máximo en x=1 y un mínimo en x= 3 .Al ejecutar el software DERIN se obtiene un mínimo en el punto (3;1) , un máximo en el punto (1;5) y un punto de inflexión en (2;3) tal como muestra la figura 37

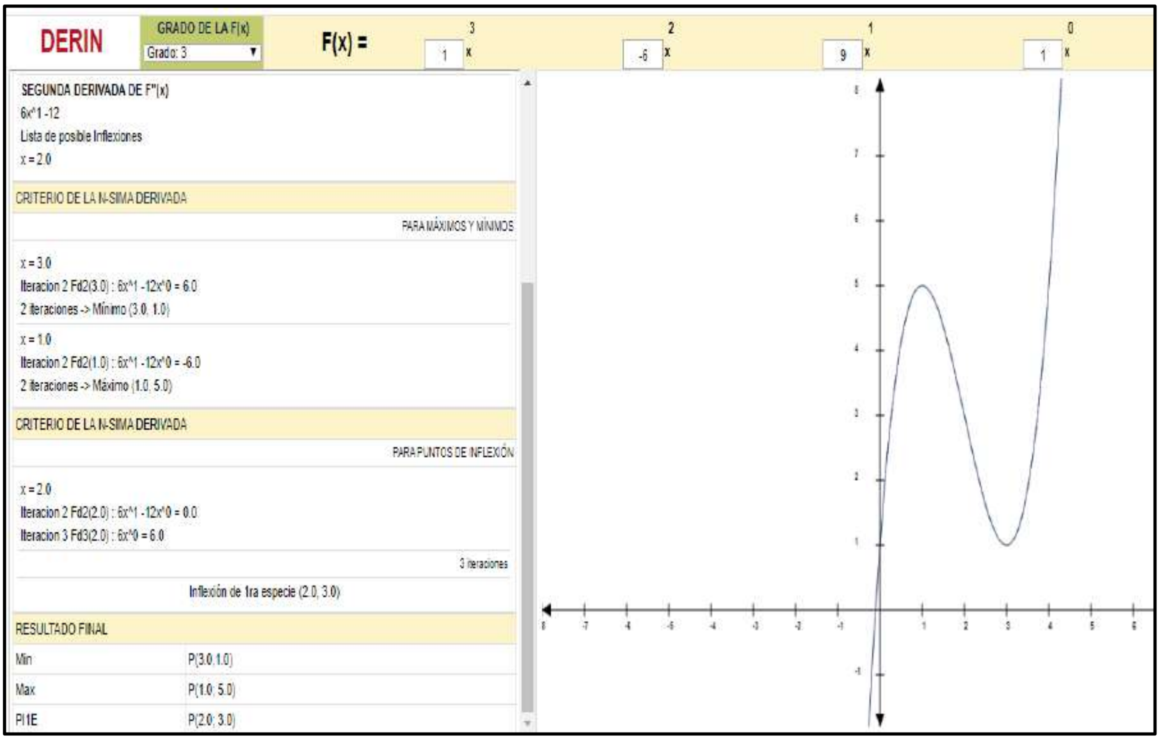


Figura 37: Problema Leithold
Fuente: DERIN

Caso 8

En el texto Cálculo Diferencial e Integral de William Granville (Granville, 2009) en la página 65 considera como ejemplo la función $f(x) = 2x^3 - 9x^2 + 12x - 3$

En la resolución del ejercicio el autor refiere en la página 66 de dicho libro que la función presenta un máximo en x = 1 y un mínimo en x= 2

Al ejecutar el software DERIN se obtiene como resultado un mínimo en el punto (2;1), un máximo en el punto (1;2) y un punto de inflexión en (1,5; 1,5) tal como muestra la figura 38

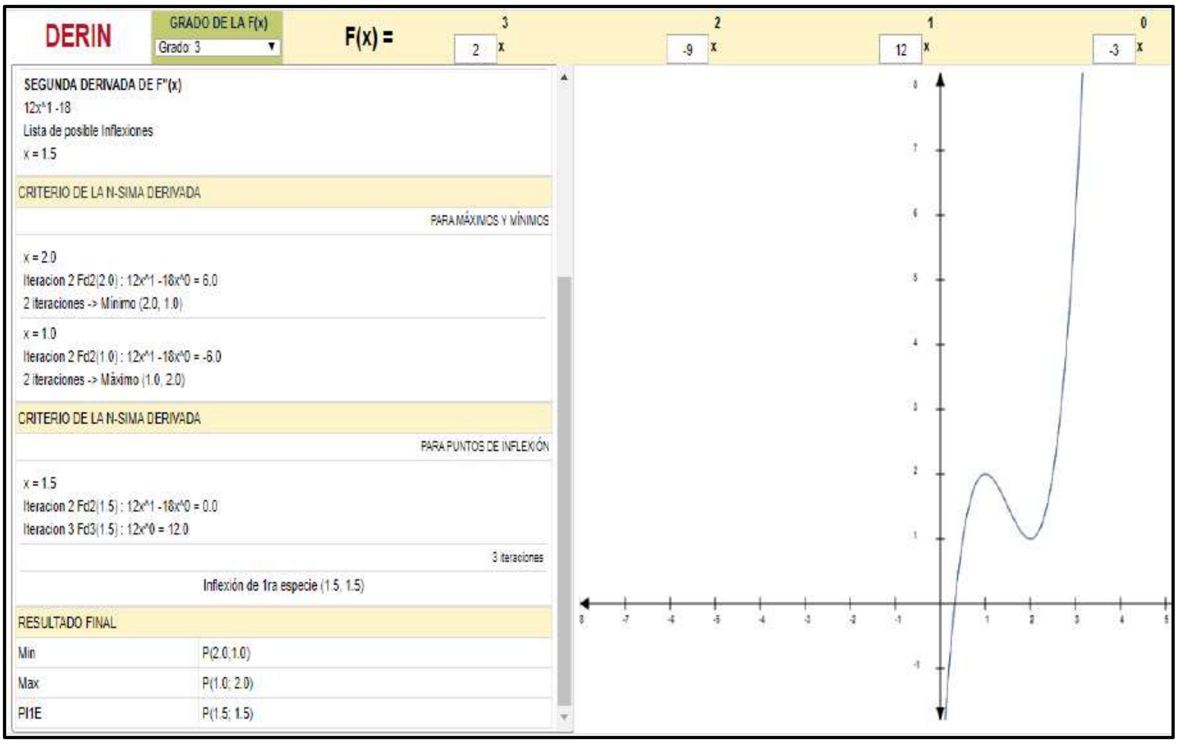


Figura 38: Problema_Granville
Fuente: software DERIN

Caso 9.

James Stewart en su libro Cálculo de una variable (Stewart, 2014) en el capítulo 4: Aplicaciones de la derivada, página 287 , propone el ejemplo :"Encuentre en qué puntos la función $f(x) = 3x^4 - 4x^3 - 12x^2 + 5$ es creciente o decreciente" y en la página 288 propone "hallar los máximos y mínimos locales de la función f"

En el marco de resolución del problema el autor concluye que existe un máximo en x= 0 y dos mínimos locales en x= -1 y x=2

Al ejecutar DERIN encuentra un máximo en (0;5) y dos mínimos en (-1;0) y (2;-27) así como una inflexión de primera especie en (1,2153;-13,3578) y de segunda especie en (-0,5486;2,3207) tal como muestra la figura 39

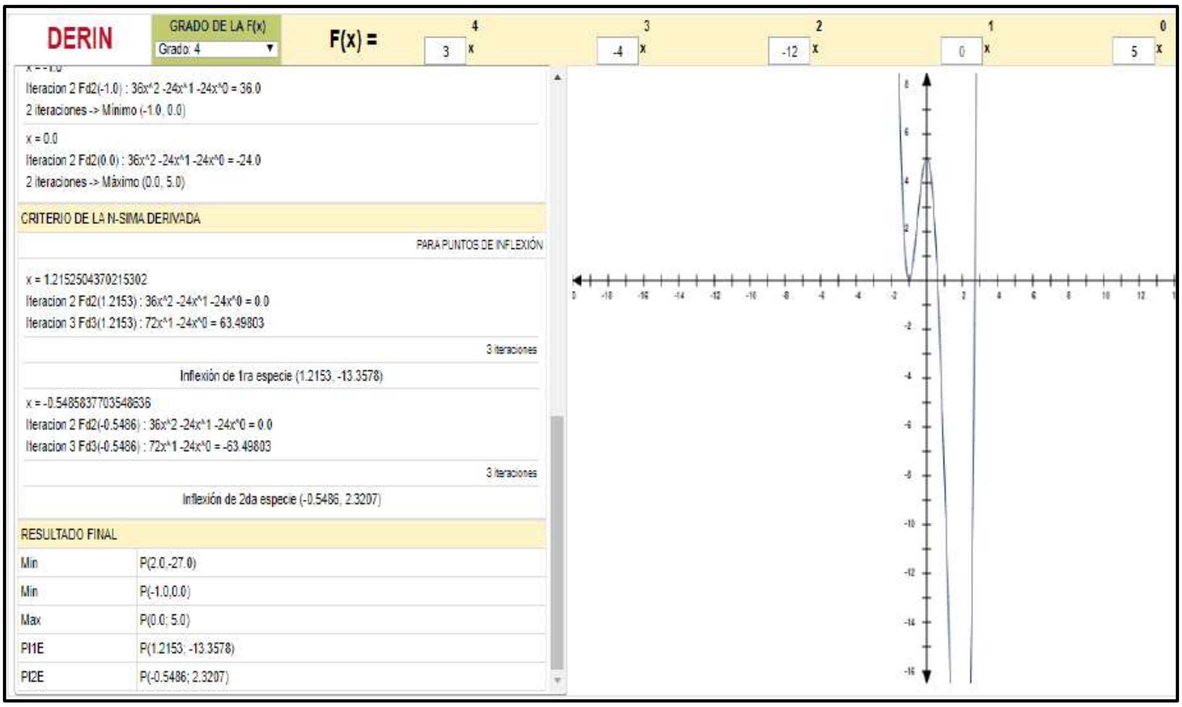


Figura 39: Problema Stewart
Fuente: DERIN

Caso 10

En el texto de consulta Cálculo Diferencial de Maynard Kong (Kong, 2001) en el capítulo 9: Aplicaciones de la derivada, en la página 396 se propone encontrar los extremos locales de la función $f(x) = x^3 - 12x + 1$

El marco resolutivo del problema usa los criterios de la primera y segunda derivada luego concluye que en x= - 2 se obtiene un máximo relativo y en x= 2 se alcanza un mínimo relativo

Al ejecutar el software DERIN se obtiene el mínimo en (2; -15) y el máximo en (-2; 17) así como un punto de inflexión de primera especie en (0;1) tal como inidica la figura 40

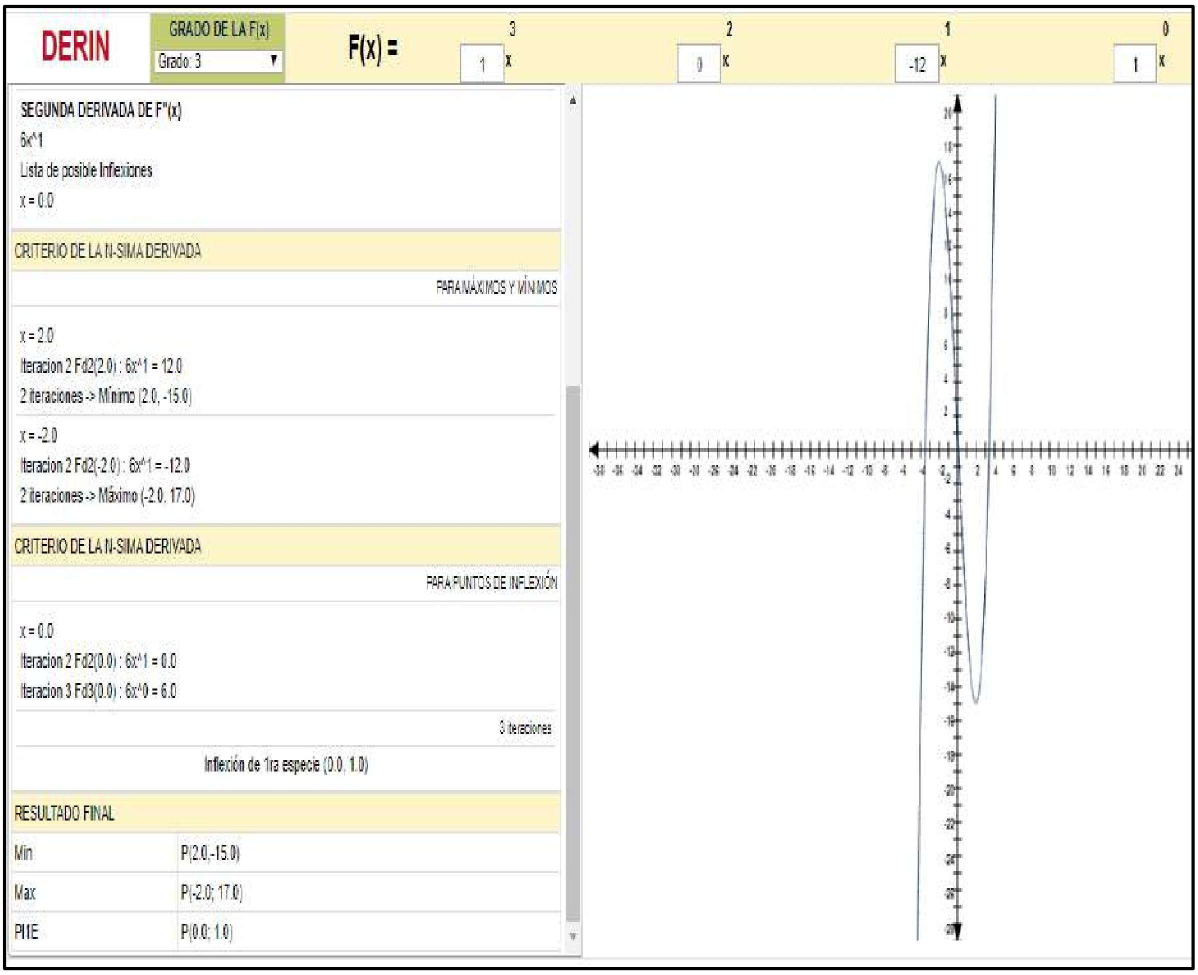


Figura 40: Problema_Kong
Fuente: DERIN

Caso 11

En el libro de consulta Tópicos de Cálculo Volumen I, de Máximo Mitacc y Luis Toro en la página 264 se propone: "Determine los intervalos de crecimiento y los valores extremos de la función $f(x) = x^3 - 3x^2 - 9x + 2$" . En la resolución los autores describen la existencia de un máximo local cuando x= -1 y un mínimo cuando x= 3.En la ejecución del software DERIN se obtiene un máximo en (-1;7) , un mínimo en (3;-25) y una inflexión de primera especie en (1;-9) tal como muestra la figura 41

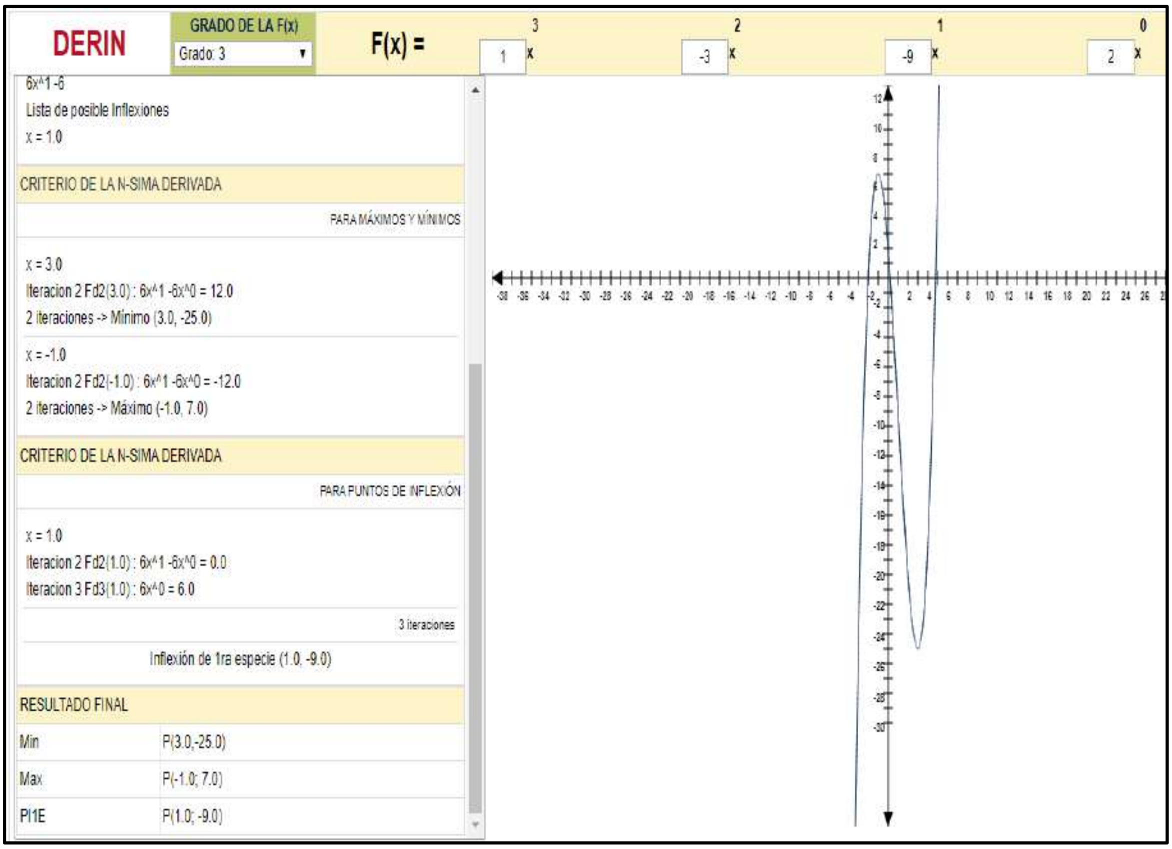


Figura 41: Problema Mitacc
Fuente: DERIN

Caso 12

En el texto Cálculo con Geometría analítica de los autores Ron Larson, Robert Hostetler y Bruce Edwards en la página 180 se propone: “determinar los intervalos abiertos sobre los cuales la función $f(x) = x^3 - \frac{3}{2}x^2$ es creciente y decreciente”

En la resolución del ejercicio se hace referencia a los puntos críticos de la función en x= 0 y x= 1 .Luego en la página 181 reutiliza el ejercicio para definir el máximo de la funcion que es ubicado en (0;0) y el mínimo en (1;-0,5)

Al ejecutar DERIN se obtiene igual resultado ubicando un máximo en (0;0), un mínimo en (1; -0,5) y un punto de inflexión en (0,5; -0,25) tal como registra la figura 42

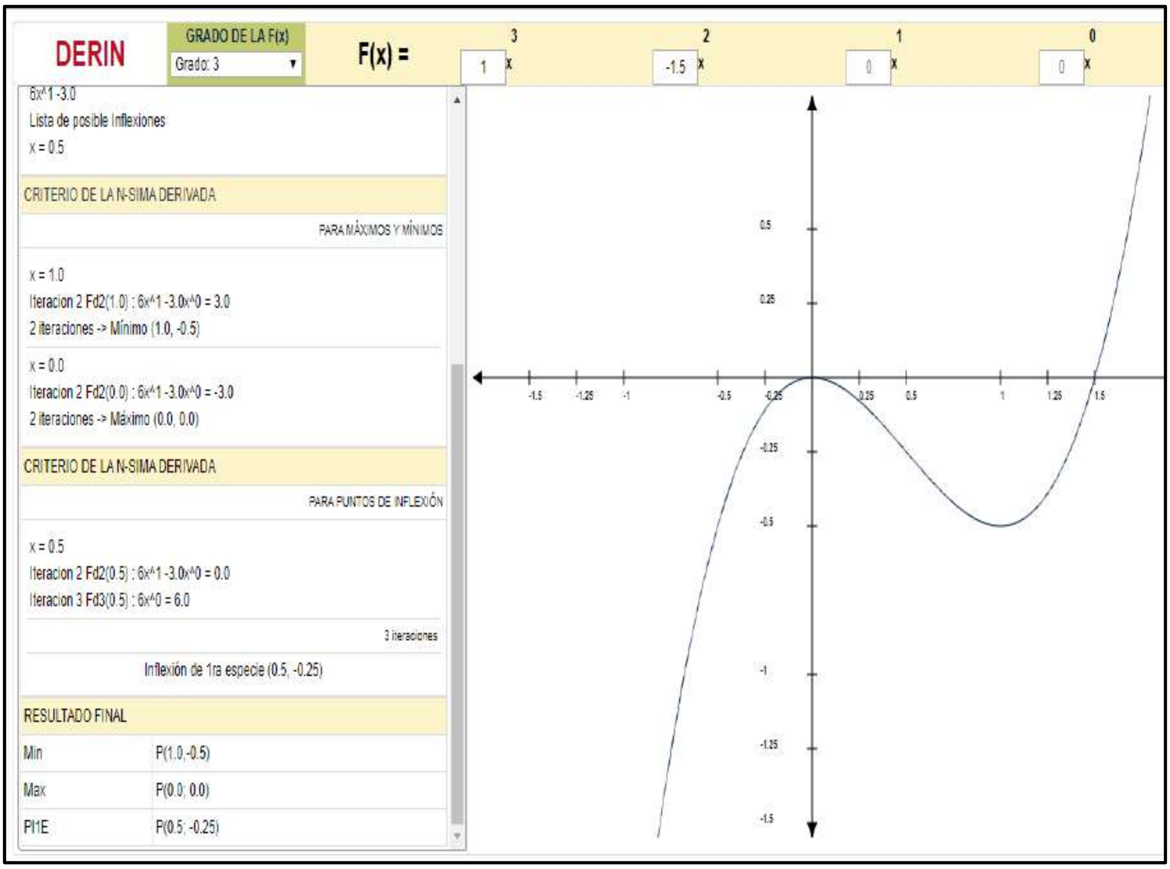


Figura 42: Problema_Larson
Fuente: DERIN

Caso 13

En el libro Cálculo Diferencial de Moisés Lázaro (Lázaro, 2014) en el capítulo 5 sobre aplicaciones de la derivada , página 301 se propone un problema referente a optimización : “Se desea construir una caja sin tapa y de base cuadrada disponiendo de 300 dm^2 de material. Halle las dimensiones para que el volumen sea máximo”

En el marco de resolución del ejercicio el autor formula el modelo matemático que representa el volumen (V) en función de la arista de base (x) según $V(x) = \frac{x}{4}(300 - x^2)$ y luego indica que en x= 10 el volumen es el mayor posible

Al usar DERIN, se ingresa la función $V(x) = \frac{x}{4}(300 - x^2)$ en el formato $V(x) = 75x - 0{,}25x^3$ y se obtiene un máximo en (10;500) lo que indica que cuando x= 10 se alcanza el máximo volumen, cuyo valor lo representa 500 y por lo cual la altura de la caja es de 5 dm. La figura 43 muestra los resultados

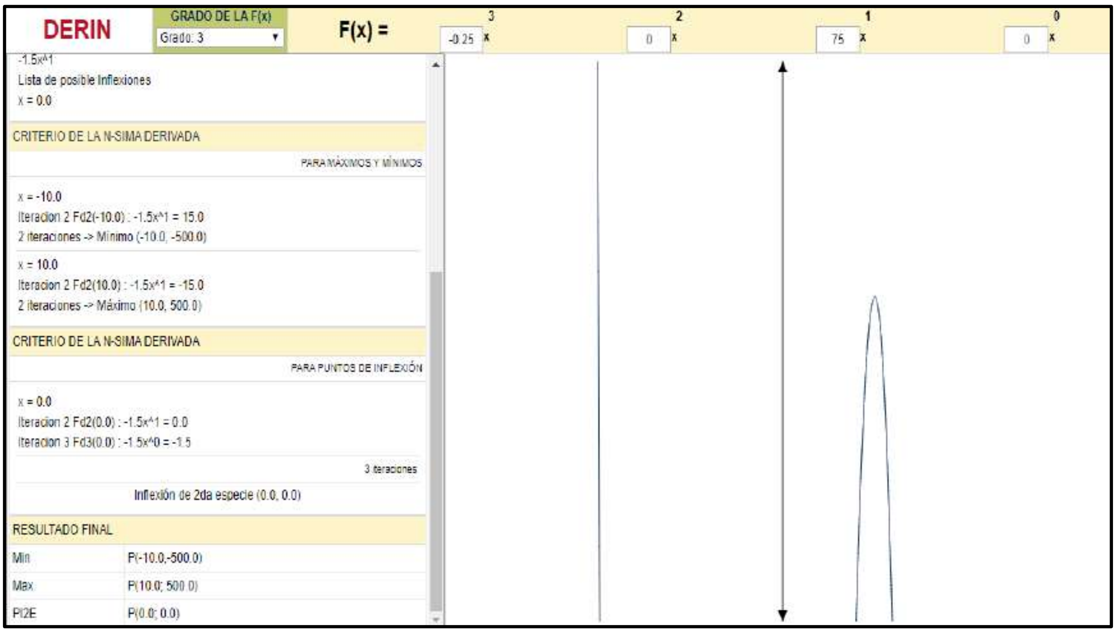


Figura 43: Problema_Lazaro
Fuente: DERIN

Caso 14

José Tola en la página 304 de su libro Análisis. propone el siguiente problema: "¿Cuáles deben ser las dimensiones de los lados de un rectángulo cuyo perímetro es 400 m para que su área sea máxima?

En la resolución del problema el autor formula el modelo matemático del área según $A(x) = 200x - x^2$ donde x representa uno de los lados del rectángulo y en la página 305 indica el valor de x debe ser igual a 100 m.

Al ejecutar DERIN (figura N°44) insertando la función área se determina un máximo en el punto (100;10000)

Figura 44: Problema_Tola
Fuente: DERIN

4.7.2 Prueba para funciones cuadráticas

4.7.2.1 Ejecución con el software DERIN

Función 1: $f_{(x)} = x^2$

Se obtiene un mínimo en el punto (0;0) , tal como indica la figura 45

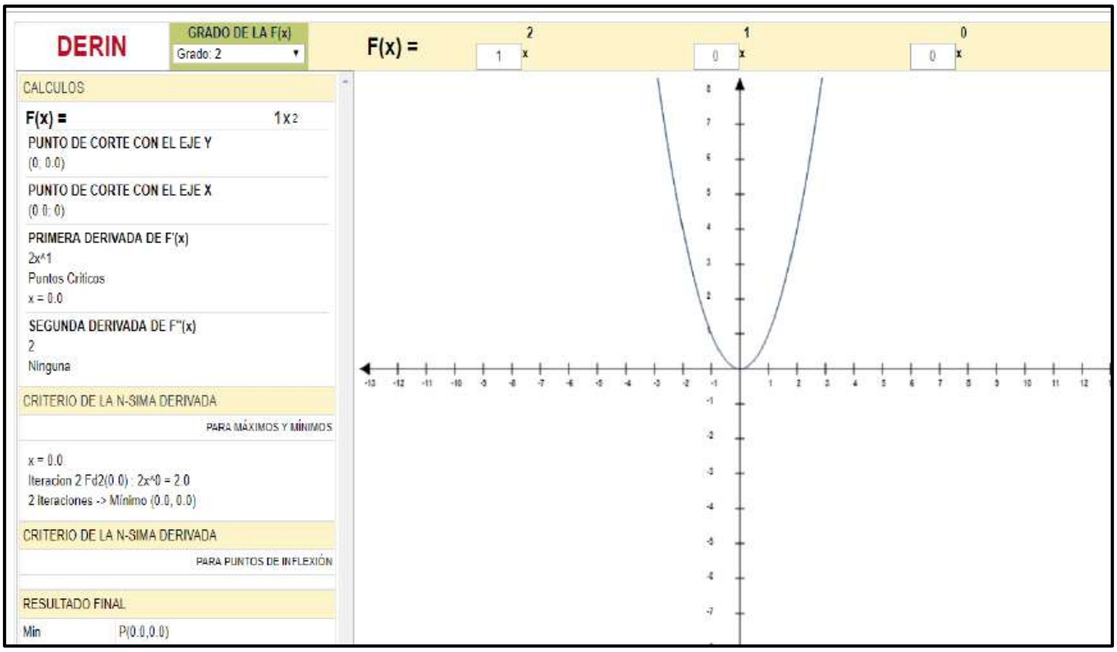


Figura N° 45 Función cuadrática 1
Fuente: software DERIN

Función 2 : $f_{(x)} = x^2 - 2x$

Describe un mínimo en el punto (1; -1) visualizado en la figura 46.

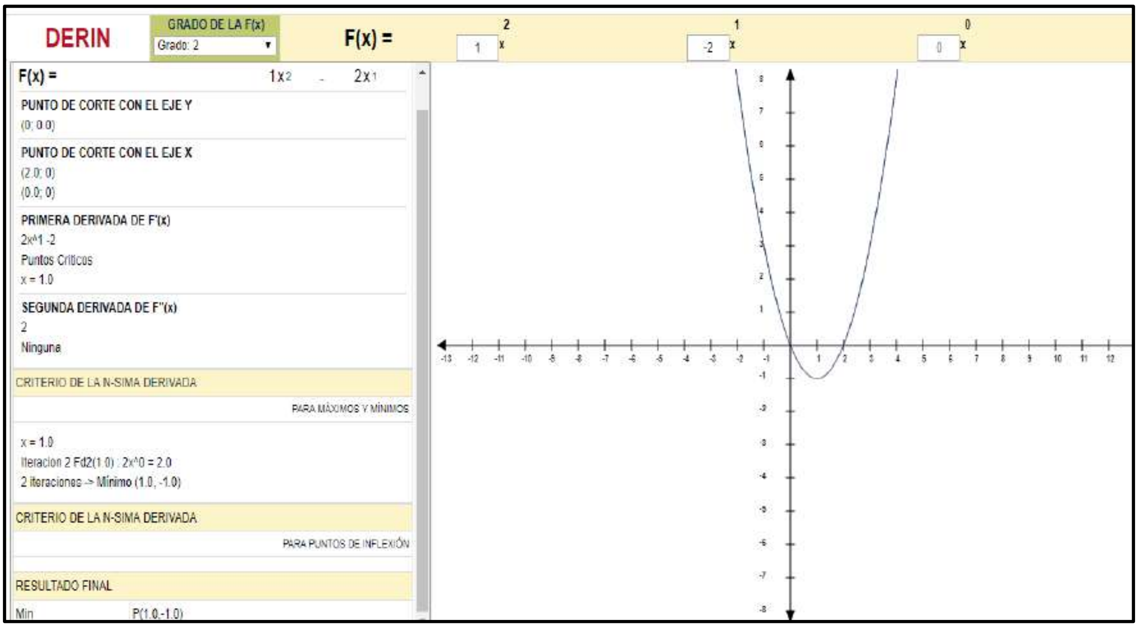


Figura N° 46 Función cuadrática 2
Fuente: software DERIN

Función 3 $f_{(x)} = x^2 - 2x - 3$

Obtiene un valor mínimo en el punto (1;-4) tal como muestra la figura 47

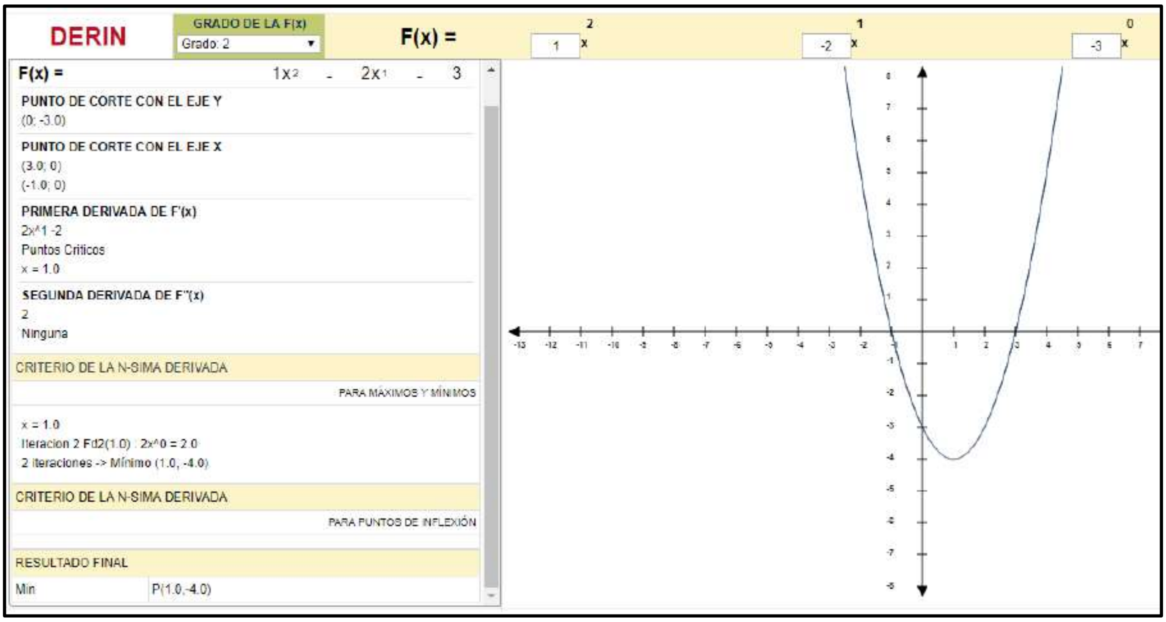


Figura N° 47 Función cuadrática 3
Fuente: Software DERIN

Función 4 $f_{(x)} = -x^2 + 5x + 24$

Muestra un máximo en el punto (2,25; 30,25) como se observa en la figura 48

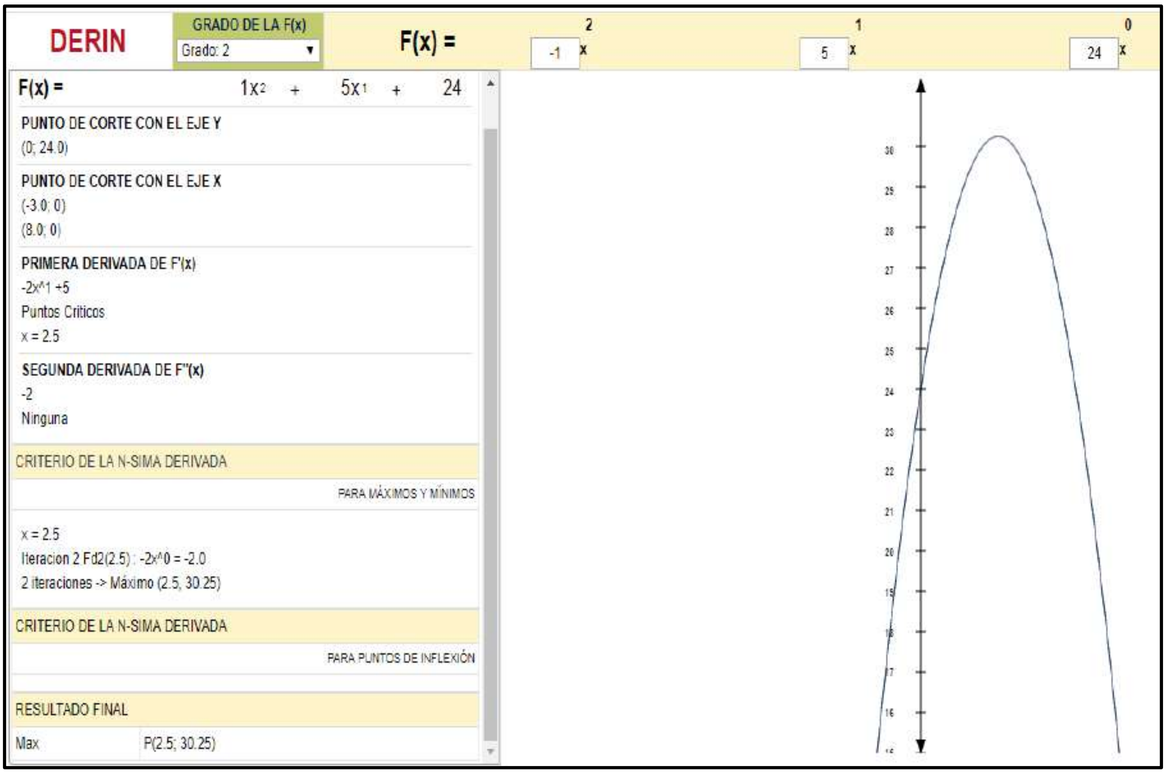


Figura N° 48 Función cuadrática 4
Fuente: Software DERIN

Función 5 $f_{(x)} = 0.5x^2 - 10$

Presenta como un mínimo en el punto (0;-10) como muestra la figura 49

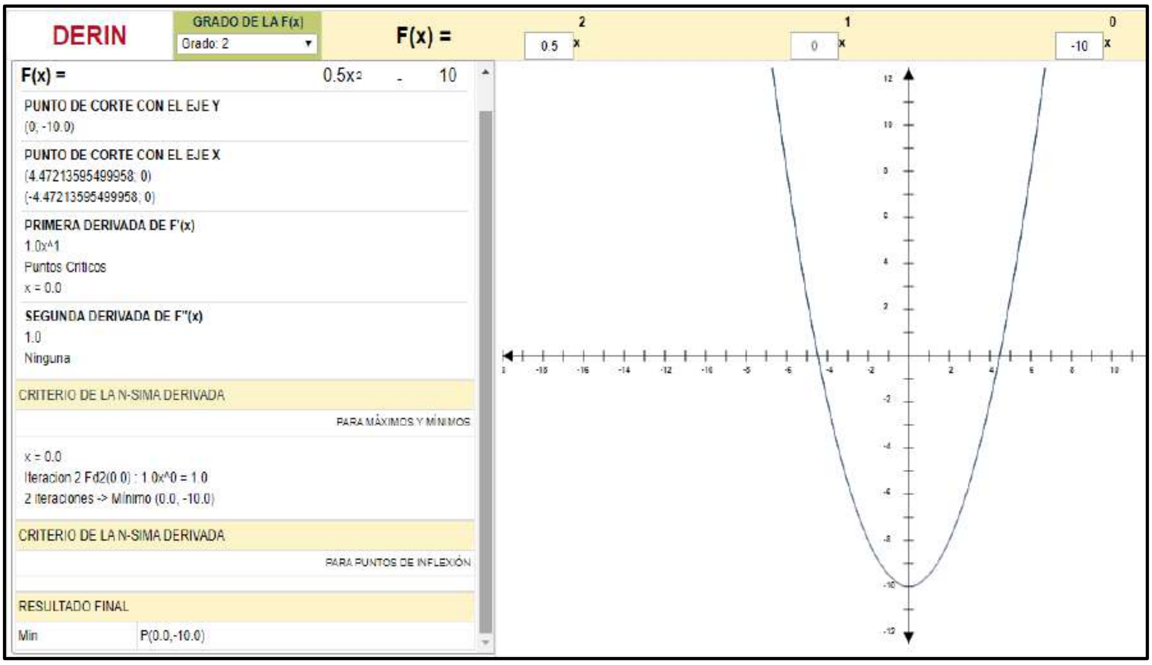


Figura N° 49 Función cuadrática 5
Fuente: Software DERIN

4.7.2.2 Ejecución con el software WolframAlpha

Función 1: $f_{(x)} = x^2$

Se obtiene un mínimo en el origen como describe la figura 50

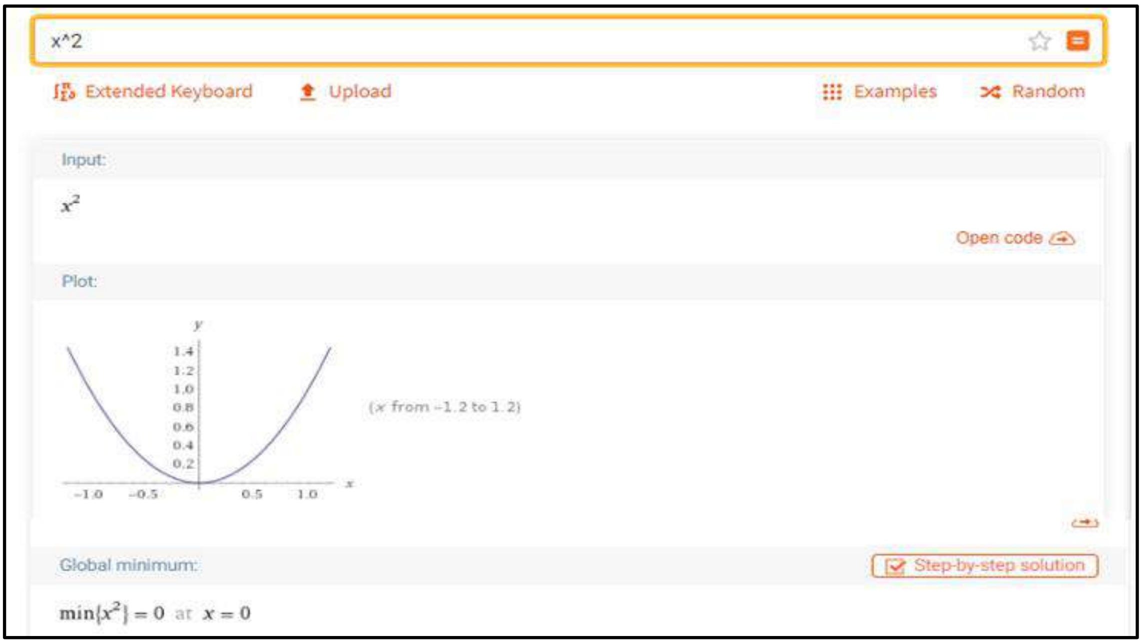


Figura N° 50 Función Cuadrática 1 Wolfram
Fuente: WolframAlpha

Función 2 : $f_{(x)} = x^2 - 2x$

Se obtiene un mínimo en (1;-1) según lo registra la figura 51

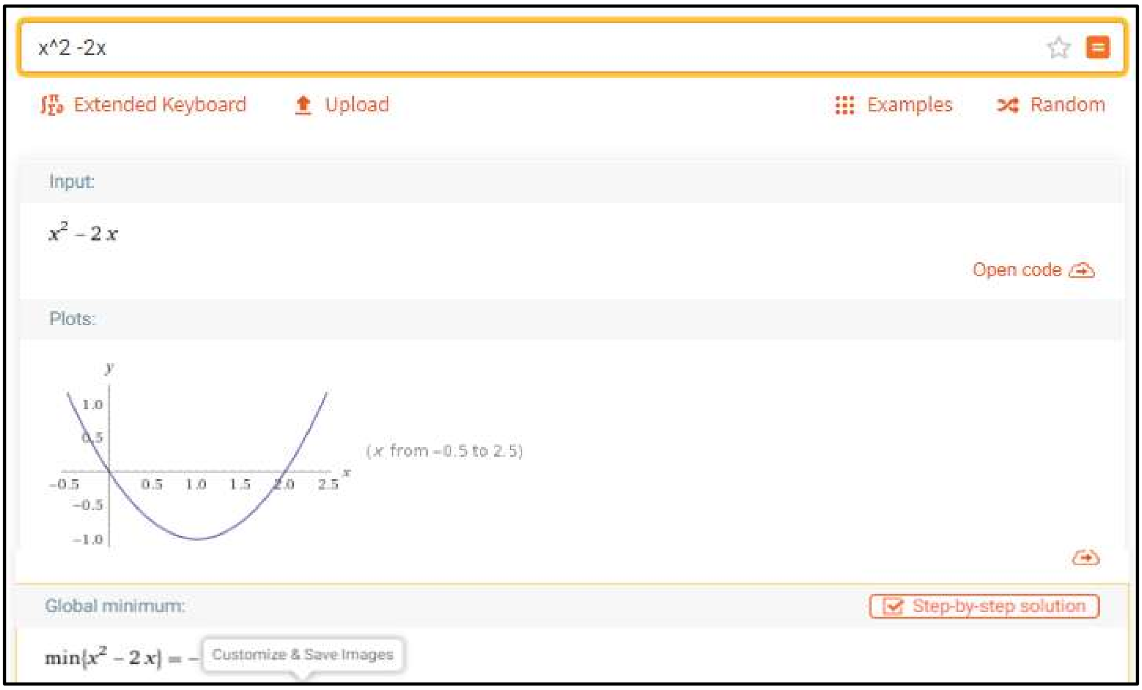


Figura N° 51 Función cuadrática 2 Wolfram
Fuente: WolframAlpha.

Función 3 $f_{(x)} = x^2 - 2x - 3$

Presenta un mínimo en el punto (1;-4) tal como muestra la figura 52

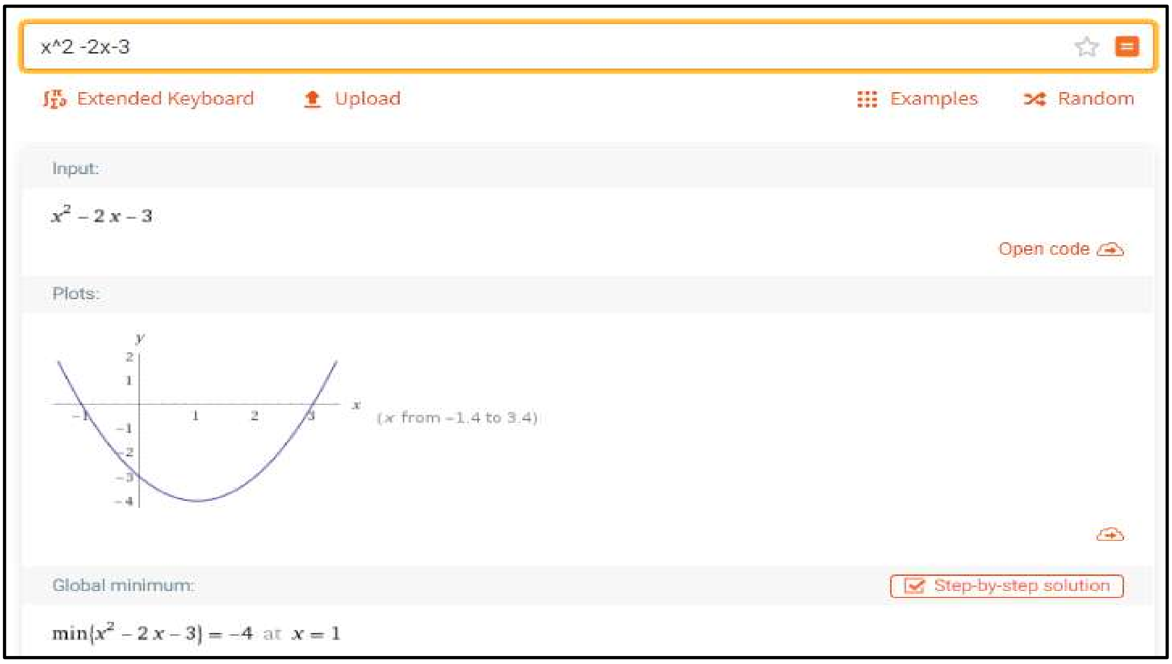


Figura N° 52 Función cuadrática 3 -Wolfram
Fuente: WolframAlpha

Función 4 $f_{(x)} = -x^2 + 5x + 24$

Define un máximo en el punto (2,25; 30,25) según muestra la figura 53

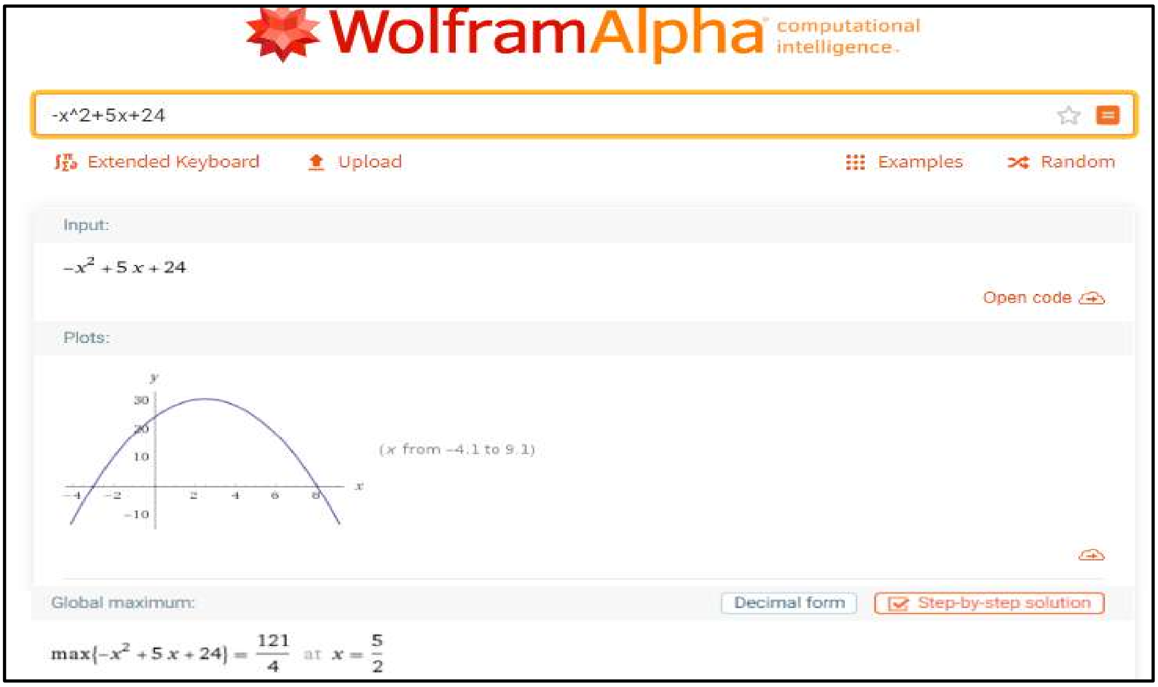


Figura N° 53 Función cuadrática 4 Wolfram
Fuente: WolframAlpha

Función 5 $f_{(x)} = 0.5x^2 - 10$

Define un mínimo en el punto (0; -10) tal como indica la figura 54

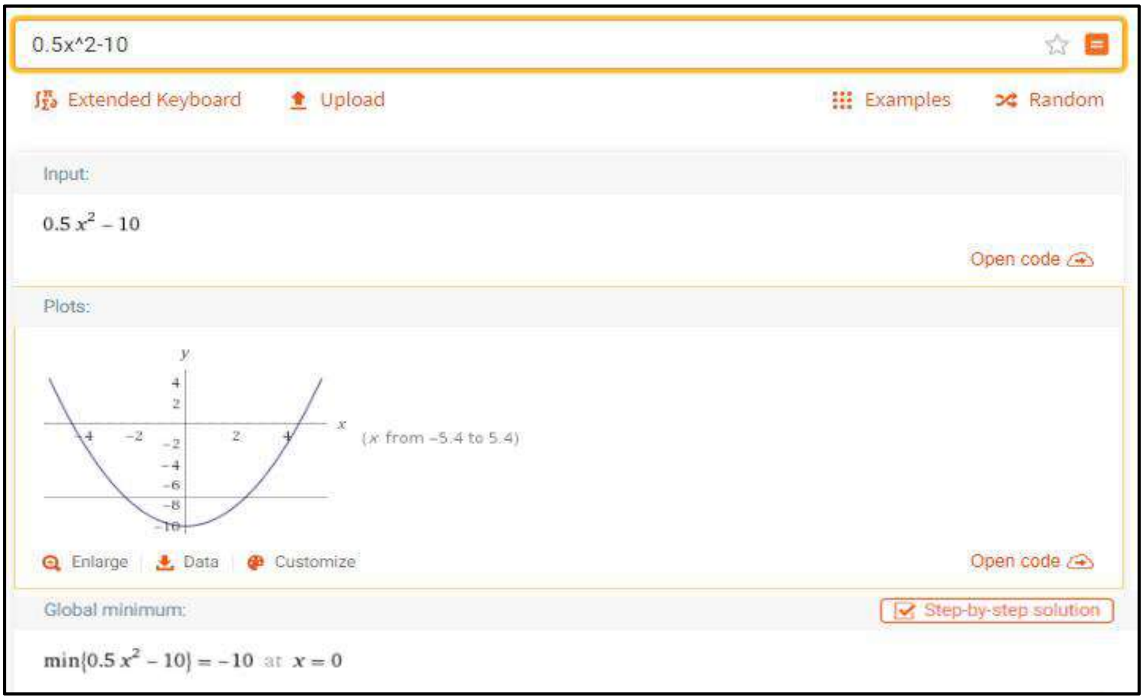


Figura N° 54 Función cuadrática 5 - Wolfram
Fuente: WolframAlpha

4.7.2.3 Ejecución con el software GeoGebra

Se debe considerar que la ejecución en GeoGebra requiere los comandos Extremo(polinomio) y Puntoinflexión(polinomio) para obtener los resultados

Función 1: $f_{(x)} = x^2$

Obtiene un mínimo en (0;0) tal como muestra la figura 55.

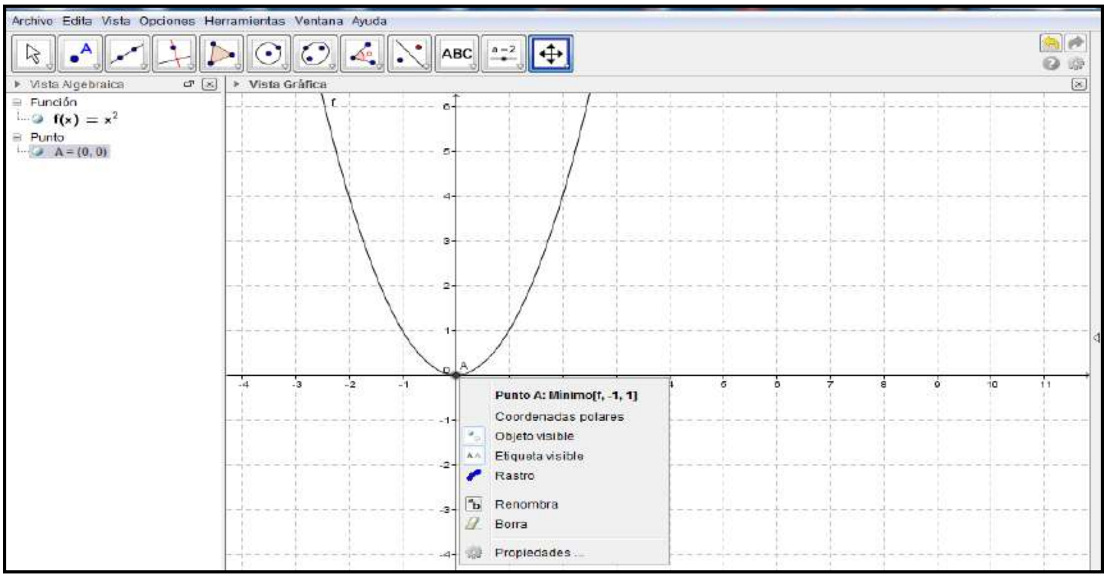


Figura N° 55 Función cuadrática 1 GeoGebra
Fuente: GeoGebra

Función 2 : $f_{(x)} = x^2 - 2x$

Se obtiene un mínimo en (-1,1) tal como muestra la figura 56

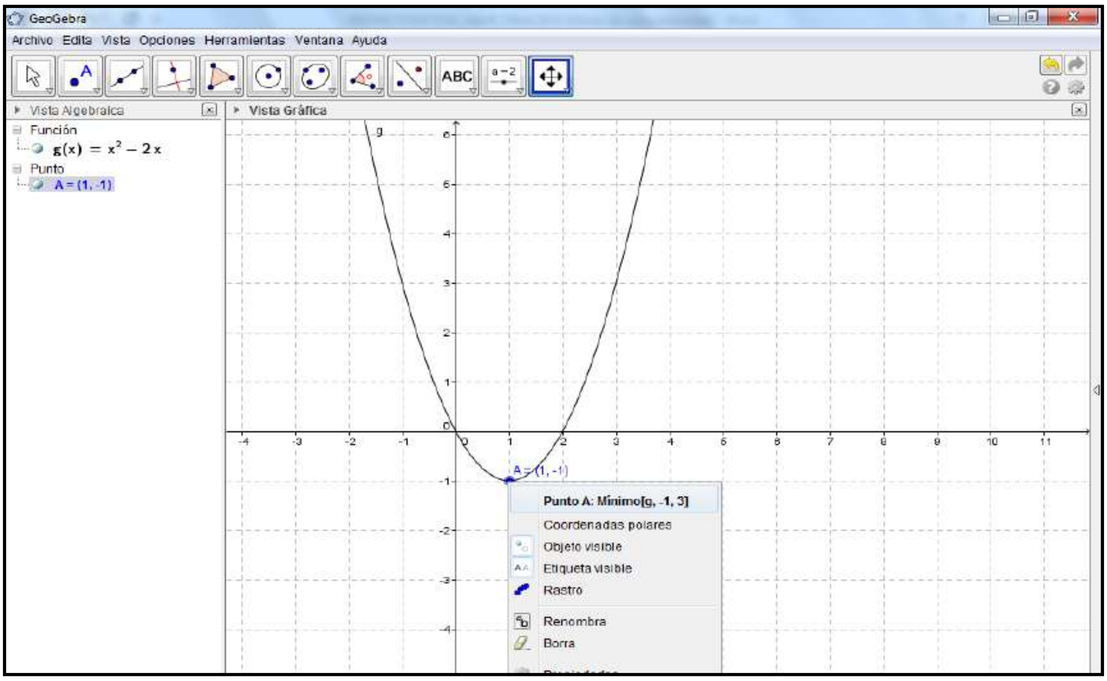


Figura N°56 Función cuadrática 2 GeoGebra
Fuente: GeoGebra

Función 3 $f_{(x)} = x^2 - 2x - 3$

Presenta un mínimo en (1;-4)

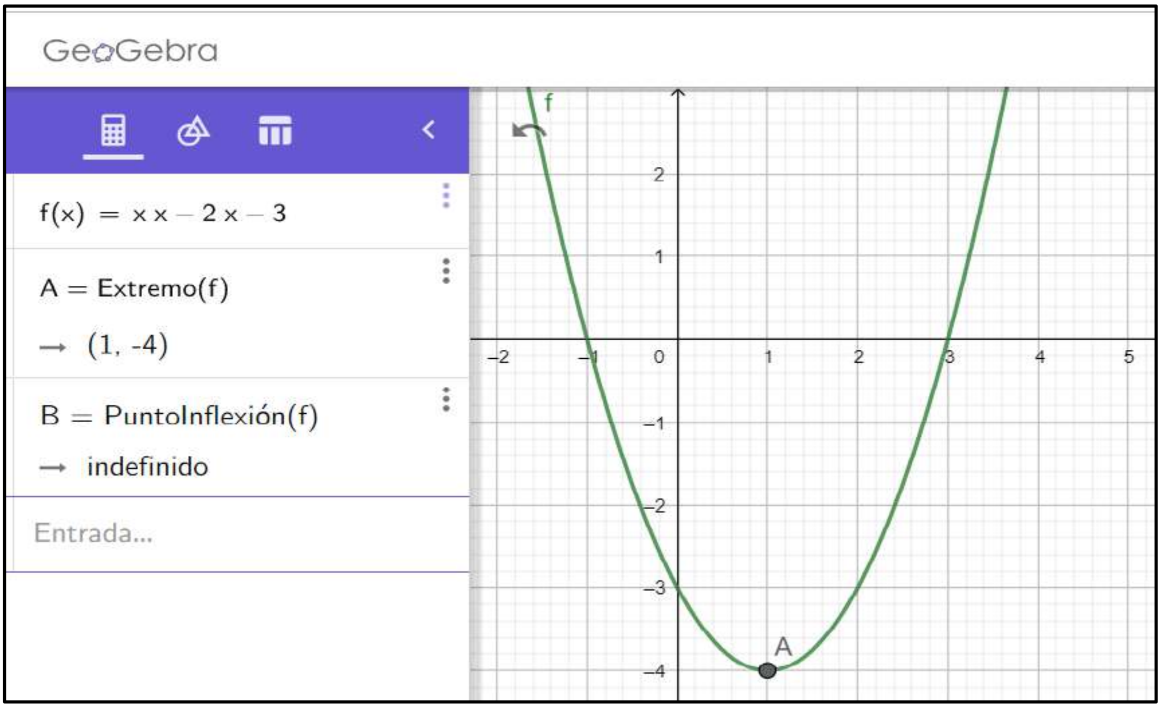


Figura N° 57 Función cuadrática 3 GeoGebra
Fuente: GeoGebra

Función 4 $f_{(x)} = -x^2 + 5x + 24$

Presenta un máximo en (2,5; 30,25) como lo muestra la figura 58

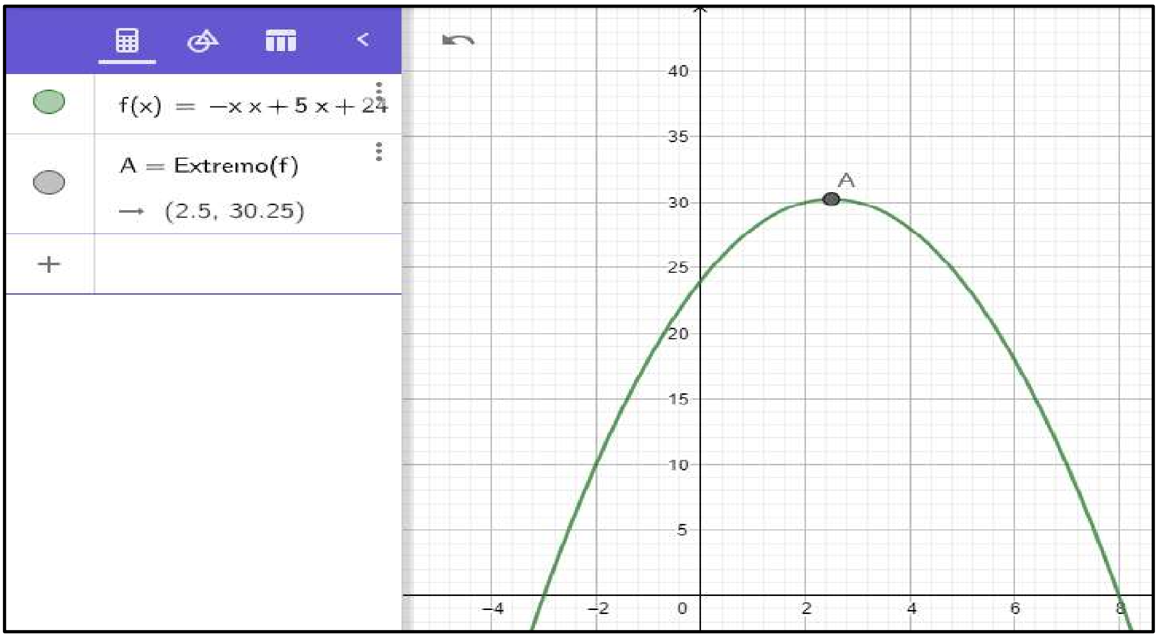


Figura N° 58 Función cuadrática 4 - GeoGebra
Fuente: GeoGebra

Función 5 $f_{(x)} = 0.5x^2 - 10$

Presenta un mínimo en (0;-10) tal como se evidencia en la figura 59

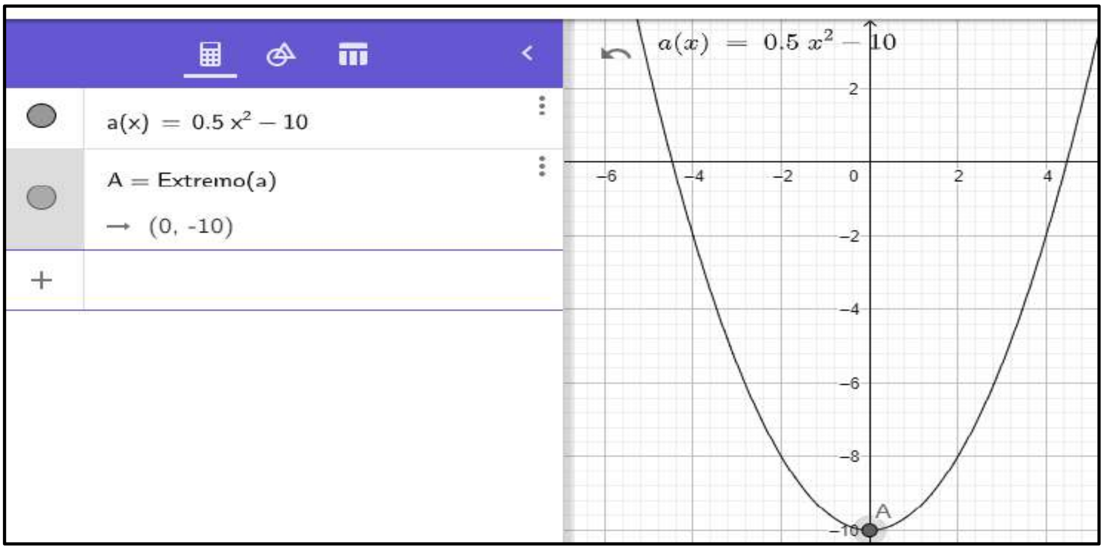


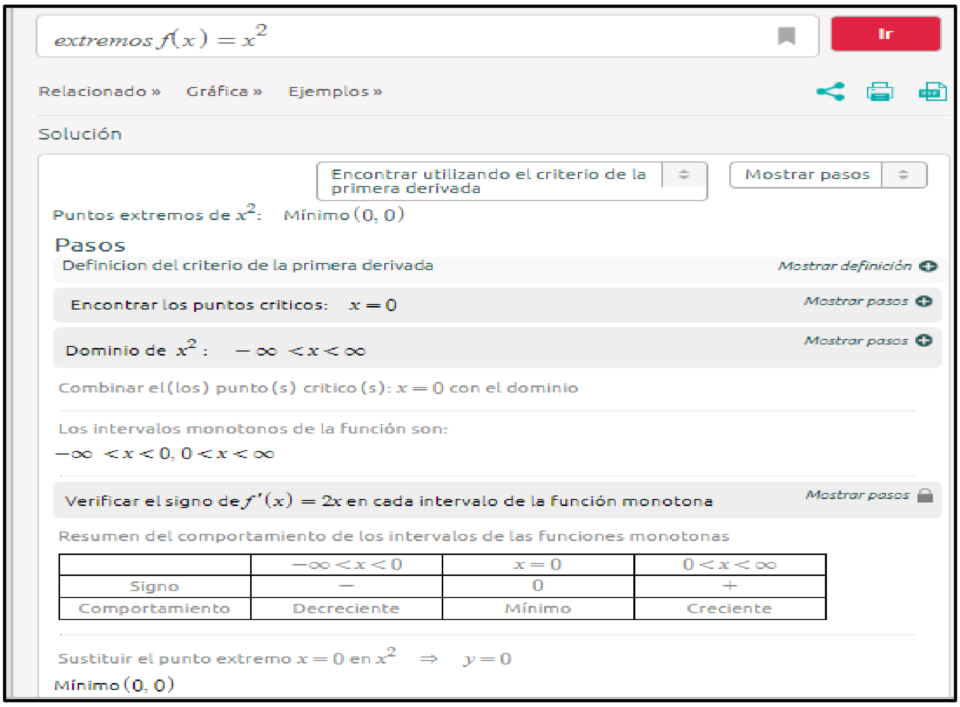
Figura N° 59 Función cuadrática 5 GeoGebra
Fuente: GeoGebra

4.7.2.4 Ejecución con el software Symbolab

Es pertinente señalar que tiene la opción de utilizar el criterio de la primera y de la segunda derivada por separado.

Función 1: $f_{(x)} = x^2$

Obtiene un mínimo en el origen como se contempla en la figura 60



extremos $f(x) = x^2$

Ir

Relacionado » Gráfica » Ejemplos »

Solución

Encontrar utilizando el criterio de la primera derivada

Mostrar pasos

Puntos extremos de x^2: Mínimo $(0, 0)$

Pasos

Definicion del criterio de la primera derivada — Mostrar definición

Encontrar los puntos criticos: $x = 0$ — Mostrar pasos

Dominio de x^2: $-\infty < x < \infty$ — Mostrar pasos

Combinar el(los) punto(s) critico(s): $x = 0$ con el dominio

Los intervalos monotonos de la función son:
$-\infty < x < 0,\ 0 < x < \infty$

Verificar el signo de $f'(x) = 2x$ en cada intervalo de la función monotona — Mostrar pasos

Resumen del comportamiento de los intervalos de las funciones monotonas

	$-\infty < x < 0$	$x = 0$	$0 < x < \infty$
Signo	−	0	+
Comportamiento	Decreciente	Mínimo	Creciente

Sustituir el punto extremo $x = 0$ en $x^2 \Rightarrow y = 0$

Mínimo $(0, 0)$

Figura 60 Función cuadrática 1 Symbolab
Fuente: Symbolab

Función 2 : $f_{(x)} = x^2 - 2x$

Presenta un valor mínimo en el punto (1;-1) como muestra la figura 61

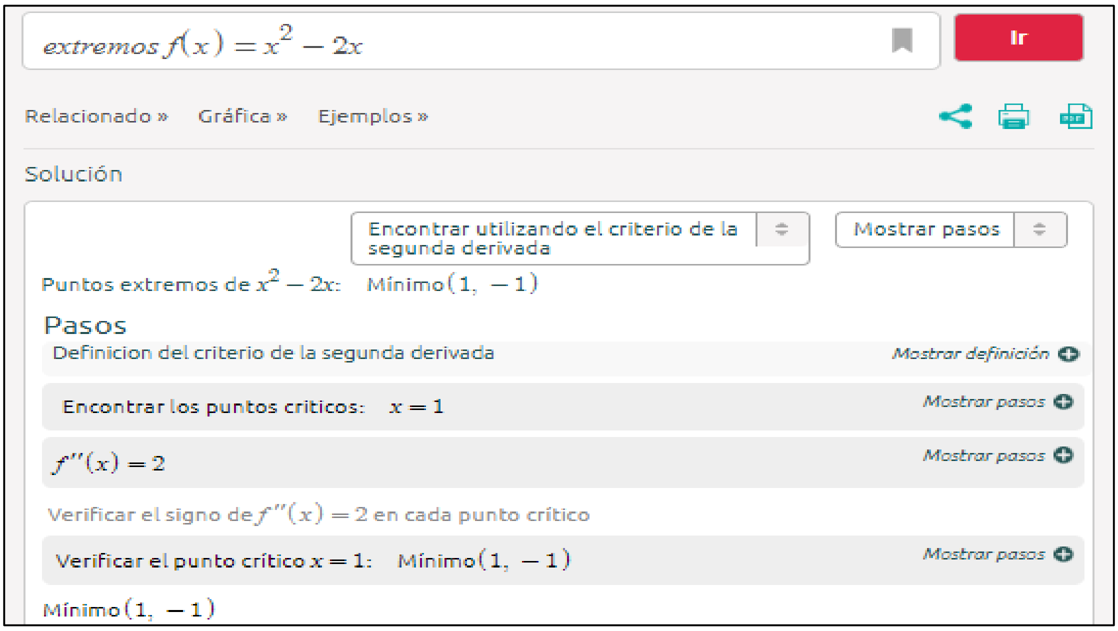


Figura N° 61 Función cuadrática 2 - Symbolab
Fuente: Symbolab

Función 3 $f_{(x)} = x^2 - 2x - 3$

Presenta un mínimo en el punto (1,-4) tal como muestra la figura 62

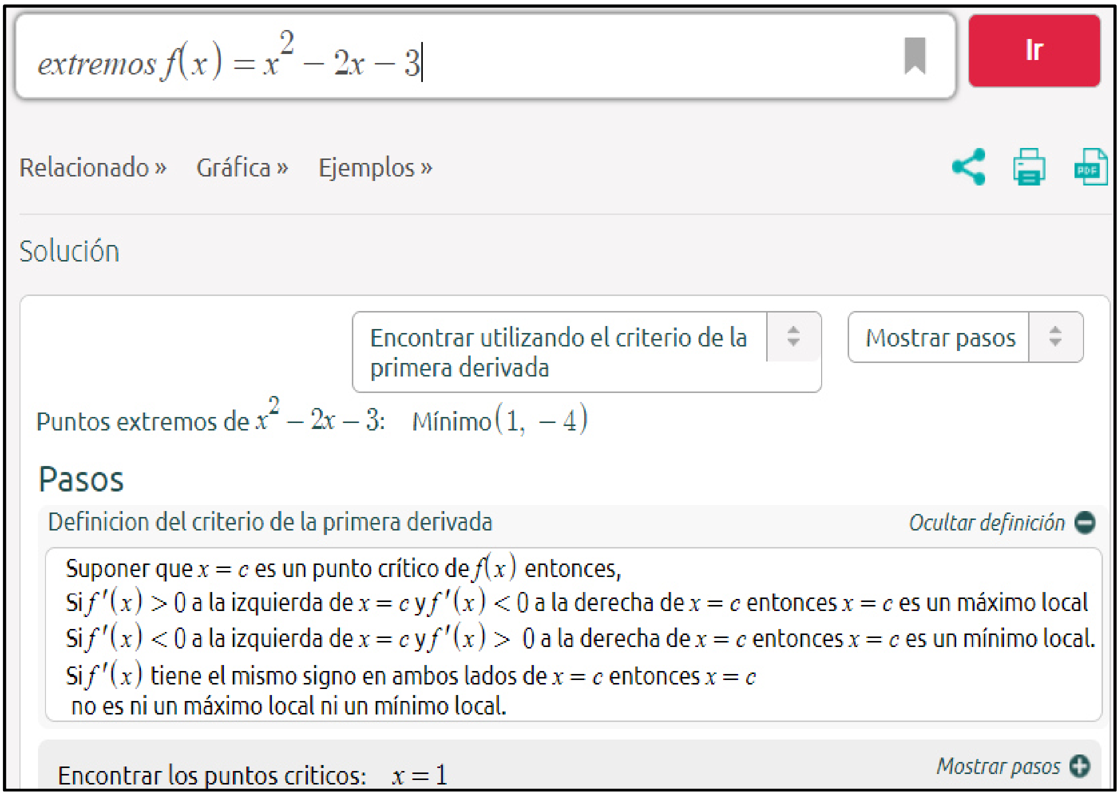


Figura N° 62 Función cuadrática 3 Symbolab
Fuente: Symbolab

Función 4 $f_{(x)} = -x^2 + 5x + 24$

Presenta un máximo en $\left(\frac{5}{2}; \frac{121}{4}\right)$ como muestra la figura 63

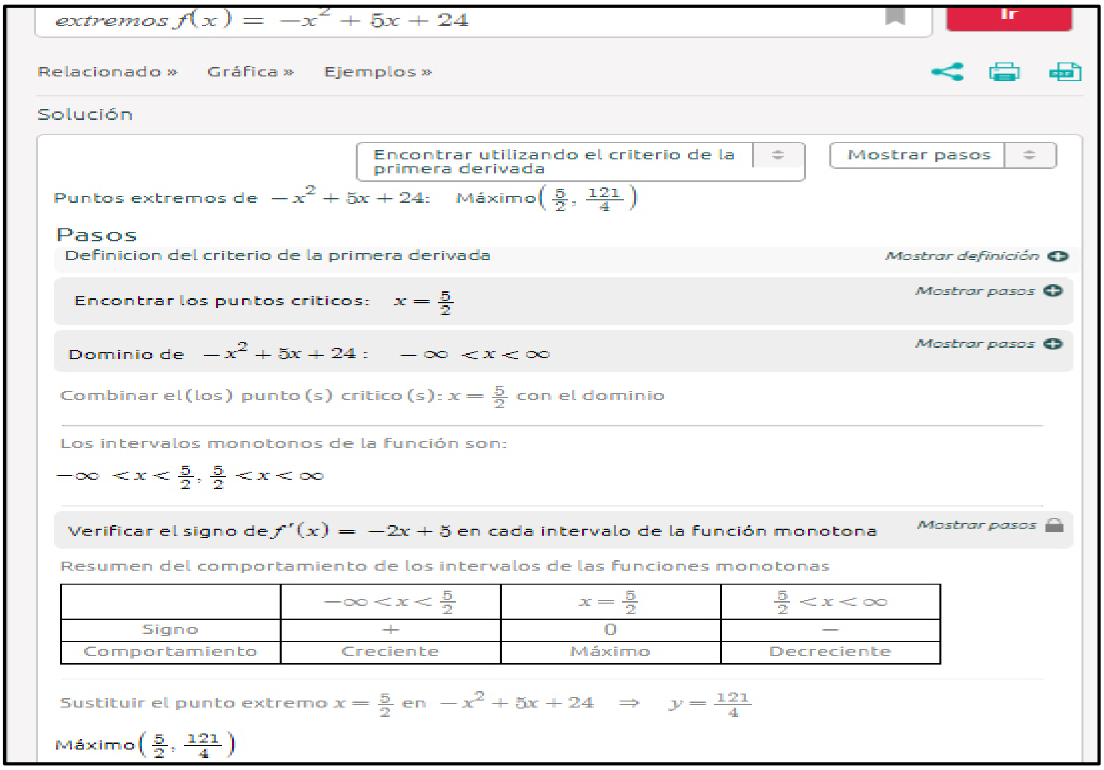


Figura N° 63 Función cuadrática 4 Symbolab
Fuente: Symbolab

Función 5 $f_{(x)} = 0.5x^2 - 10$

Presenta un mínimo en (0; -10) tal como muestra la figura 64

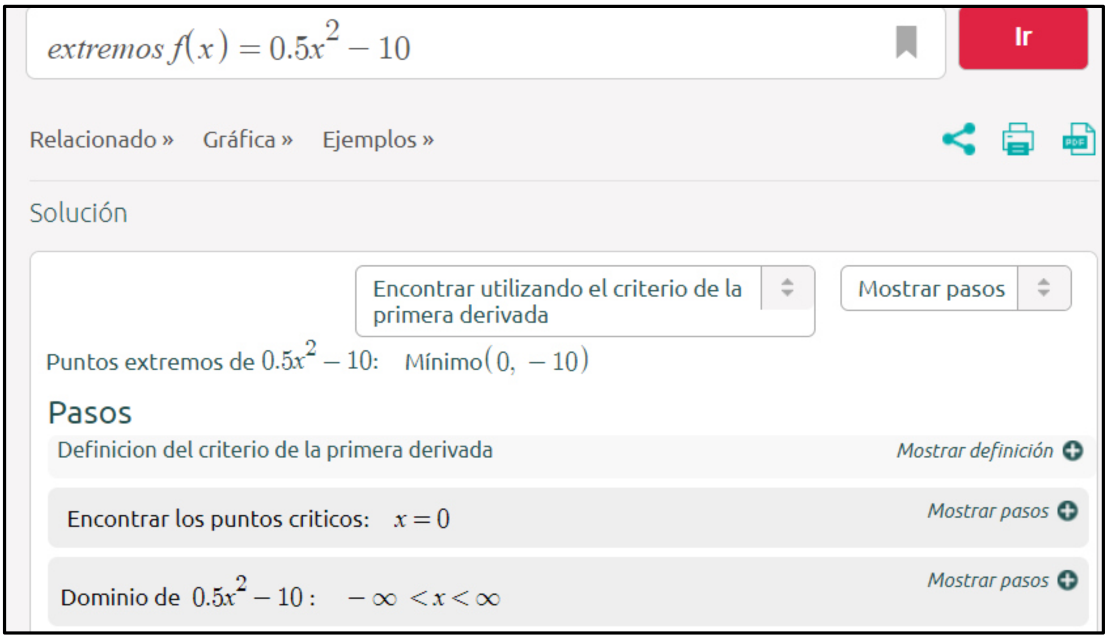


Figura N° 64 Función cuadrática 5 Symbolab
Fuente: Symbolab

4.7.3 Prueba para funciones cúbicas

4.7.3.1 Ejecución con el software DERIN

Función 6 $f_{(x)} = x^3$

Obtiene un punto de inflexión de primera especie (PI1E) en el punto (0;0). No presenta extremos relativos como muestra la figura 65

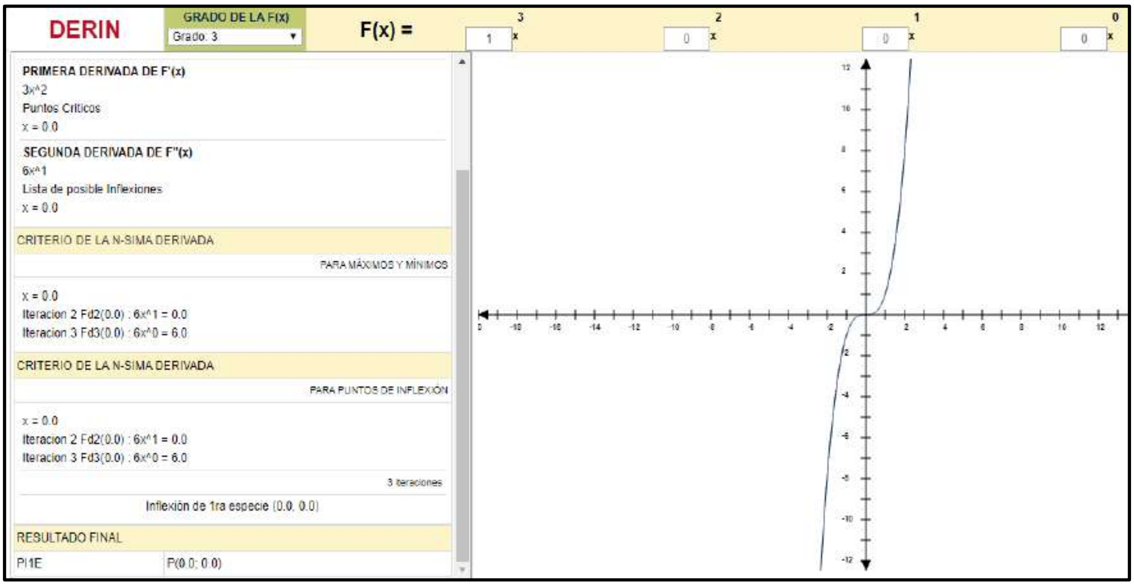


Figura N° 65 Función cúbica 1
Fuente: software DERIN

Función 7 $f_{(x)} = x^3 - 2x^2$

Ubica el mínimo en (1,33;-1,1852), el máximo en (0;0) y una inflexión de primera especie (PI1E) en (0,667;-0,5926) como muestra la figura 66.

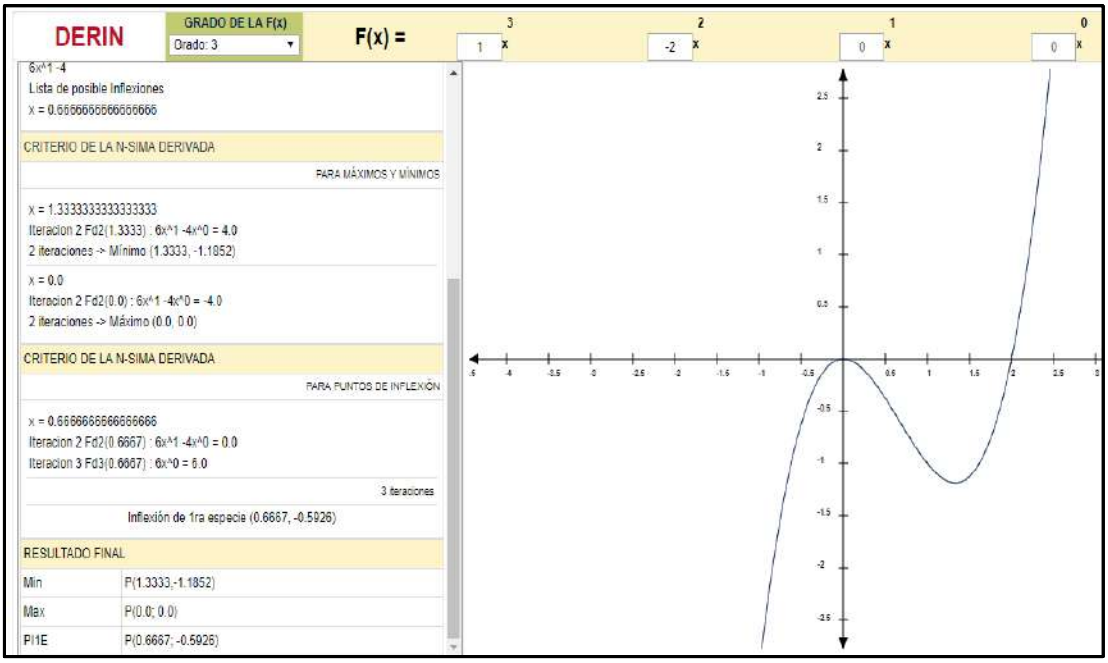


Figura N° 66 Función cúbica 2
Fuente: software DERIN

Función 8 $f_{(x)} = x^3 - 4x^2 + 3$

Ubica un máximo en (0;3), un mínimo en (2,67; -6,48) y un punto de inflexión en (1,33;-1,74) como muestra la figura 67.

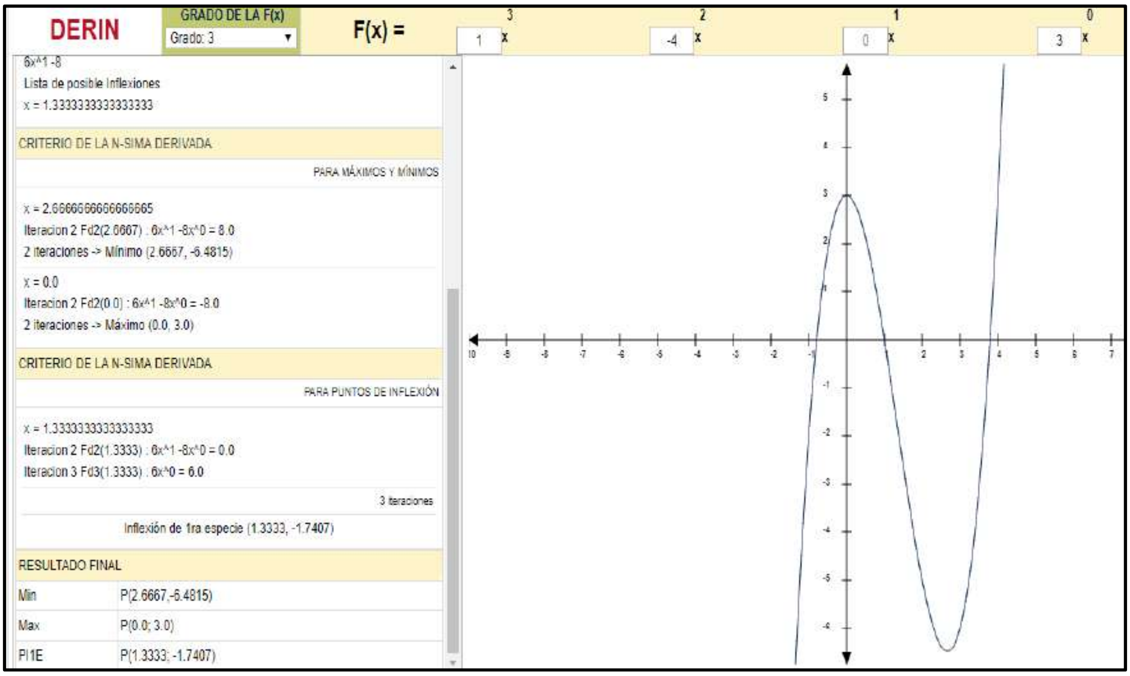


Figura N° 67 Función cúbica 3
Fuente: software DERIN

Función 9 $f_{(x)} = x^3 + 3x$

Ubica un punto de inflexión en (0;0) como muestra la figura 68

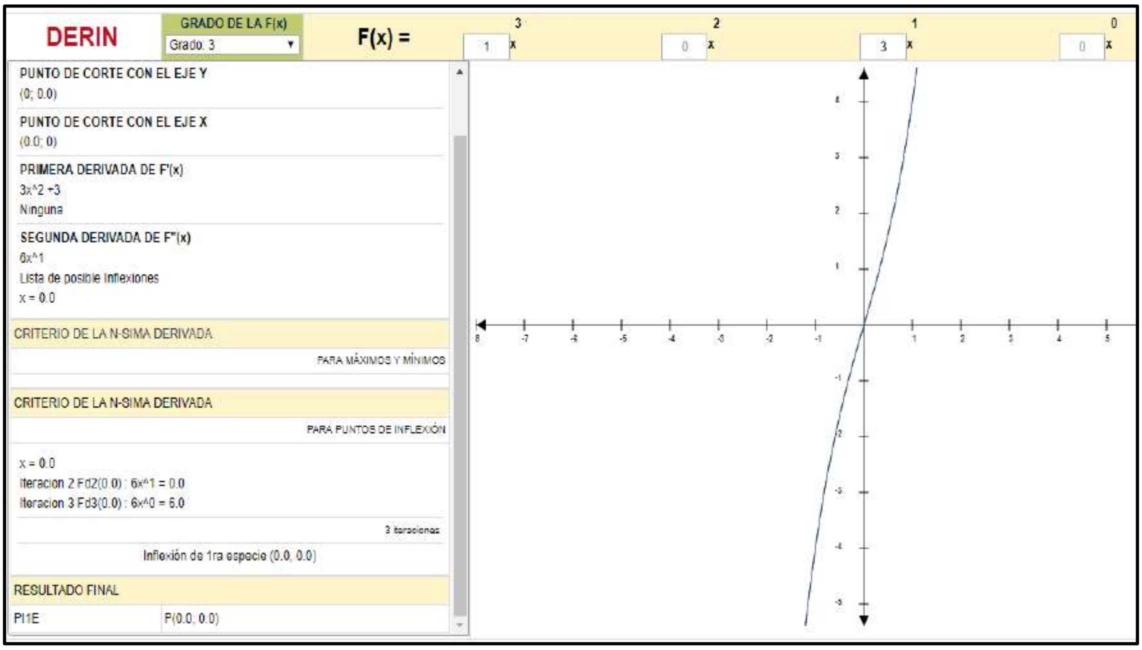


Figura N° 68 Función cúbica 4
Fuente: Software DERIN

Función 10 $f_{(x)} = x^3 - 3x$

Ubica un máximo en (-1;2) , un mínimo en (1;-2) y un punto de inflexión de primera especie en el punto (0,0) como muestra la figura 69

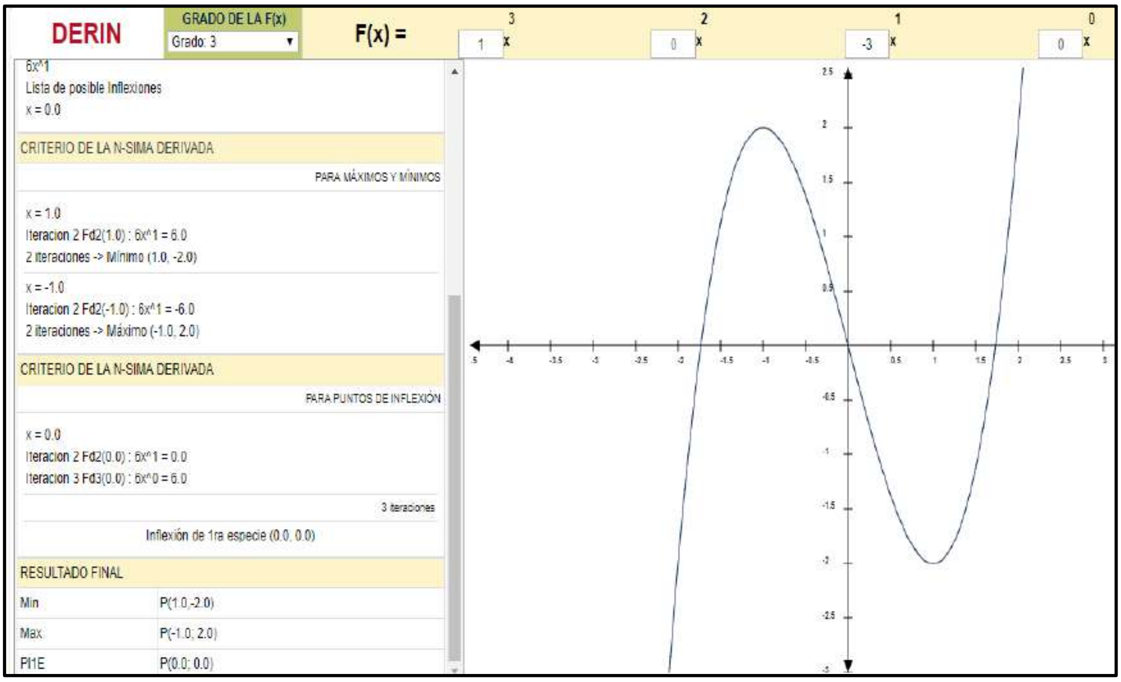


Figura N° 69 Función cúbica 5
Fuente: Software DERIN

Función 11 $f_{(x)} = x^3 - 7x + 6$

Ubica un mínimo local en (1,53;-1,12) , el máximo local en (-1,53;13,13) y una inflexión de primera especie en (0;6) como muestra la figura 70.

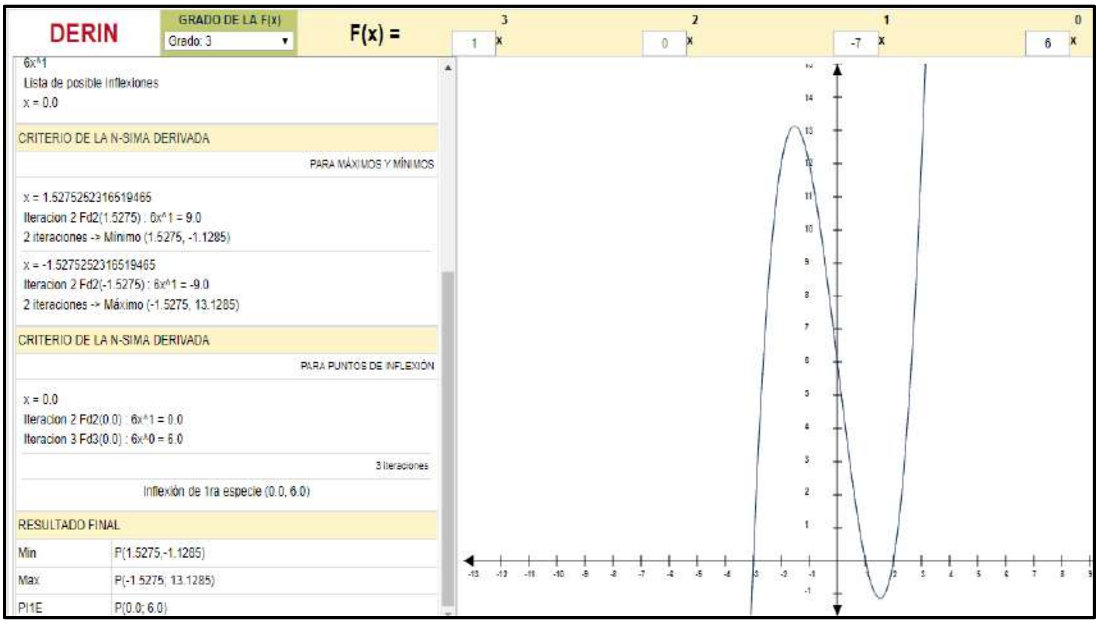


Figura N° 70 Función cúbica 6
Fuente: software DERIN

Función 12 $f_{(x)} = x^3 - 2x^2 + x$

Ubica un mínimo en (1,0), un máximo en (0,33;0,14) y una inflexión en (0,67;0,074) tal como muestra la figura 71

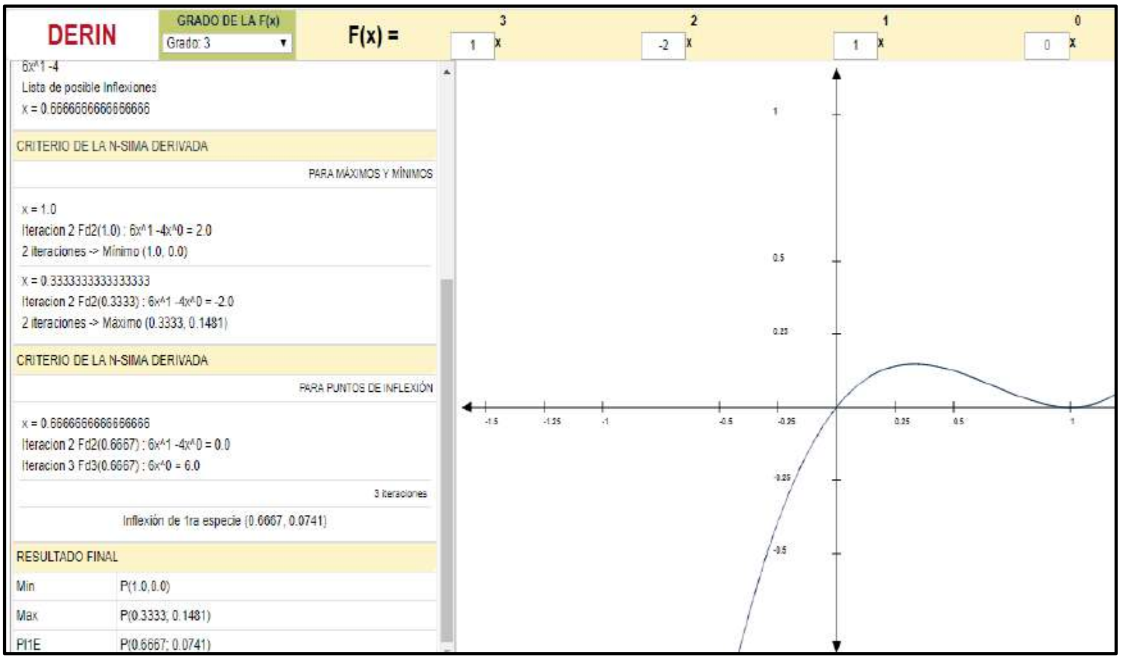


Figura N° 71 Función cúbica 7
Fuente: DERIN

Función 13 $f_{(x)} = x^3 + 2x^2 - 3x - 2$

Ubica un mínimo en (0,54;-2,88) , un máximo en (-1,87;4,06) y un punto de inflexión en (-0,67; 0,59) como muestra la figura 72

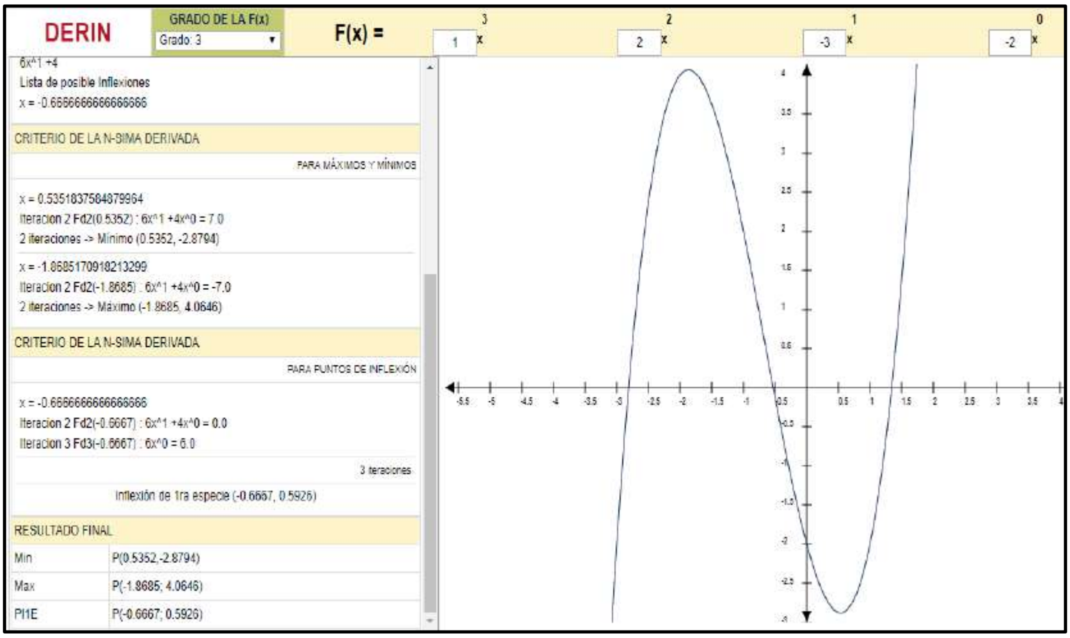


Figura N° 72 Función cúbica 8
Fuente: DERIN

Función 14 $f_{(x)} = -1,2x^3 + 3$

Ubica un punto de inflexión de segunda especie en (0;3) como muestra la figura 73

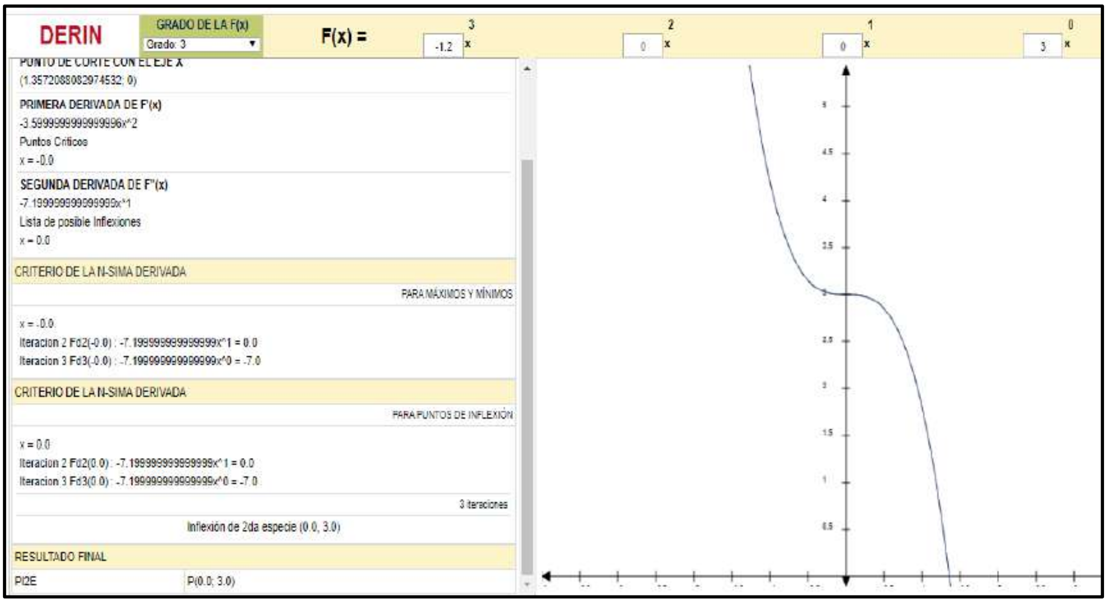


Figura N° 73 Función cúbica 9
Fuente: DERIN

4.7.3.2 Ejecución con el software WolframAlpha

Wolfram no ubica puntos de inflexión, requiere comando adicional

Función 6 $f_{(x)} = x^3$

No indica punto de inflexión como muestra la figura 74

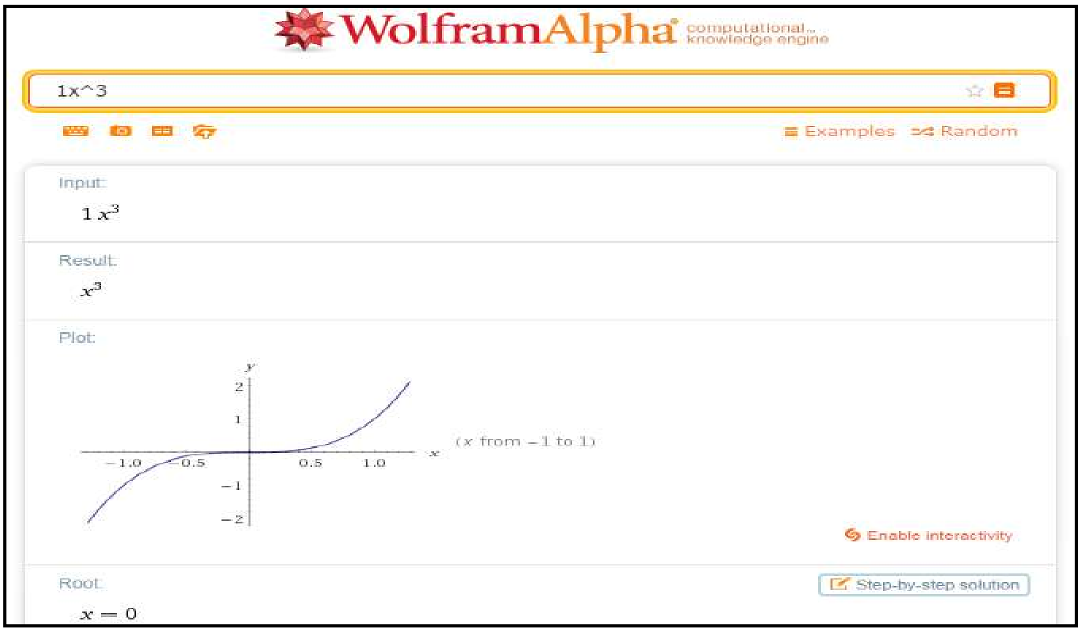


Figura N° 74 Función cúbica 1 Wolfram
Fuente: WolframAlpha

Función 7 $f_{(x)} = x^3 - 2x^2$

Ubica un mínimo en (1,33;-1,1852) , un máximo en (0;0) como muestra la figura 75

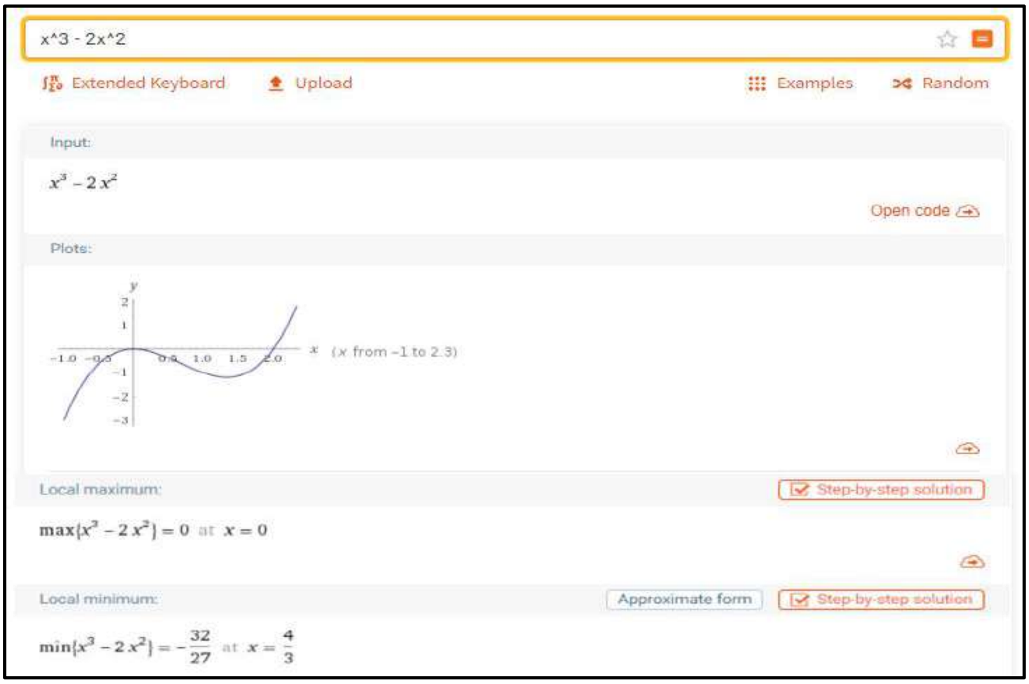


Figura N° 75 Función cúbica 2 Wolfram
Fuente: WolframAlpha

Función 8 $f_{(x)} = x^3 - 4x^2 + 3$

Ubica un mínimo en $\left(\frac{8}{3}; \frac{-175}{27}\right)$ y un máximo en (0;3) como muestra la figura 76

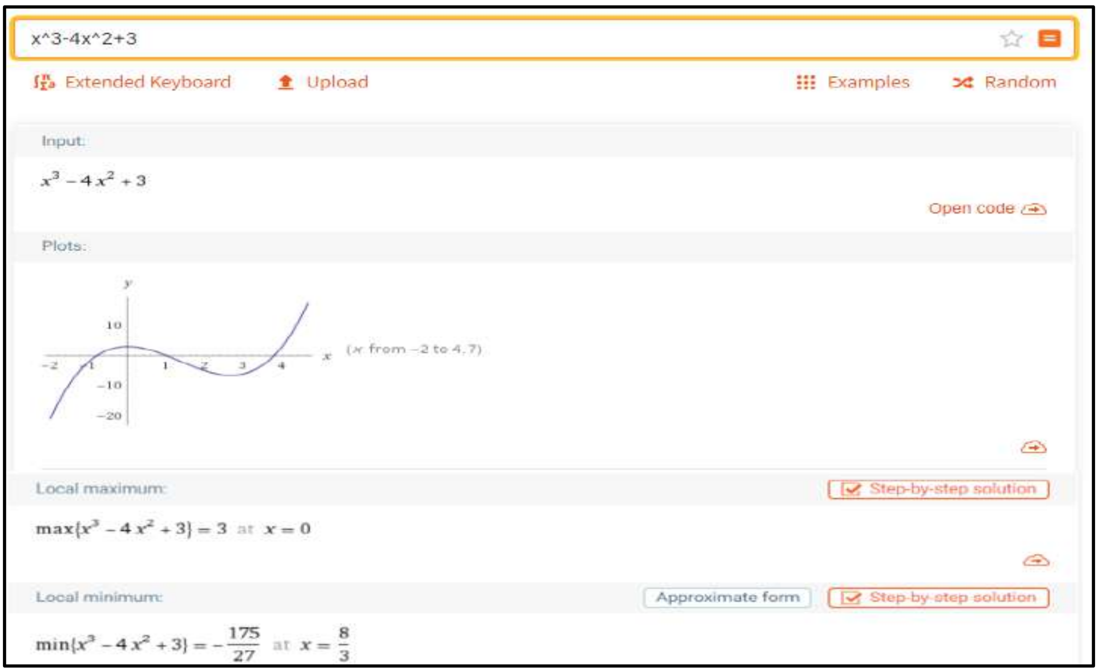


Figura N° 76 Función cúbica 3 Wolfram
Fuente: WolframAlpha

Función 9 $f_{(x)} = x^3 + 3x$

WolframAlpha : Al igual que DERIN no proporciona puntos extremos, pero no determina el punto de inflexión como ilustra la figura 77

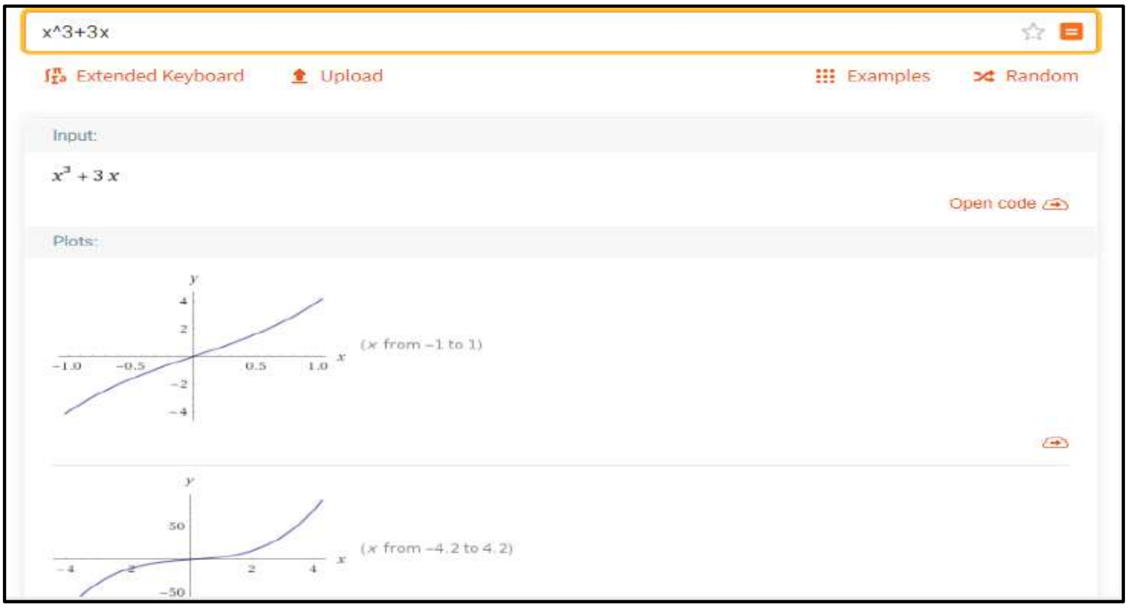


Figura N° 77 Función cúbica 4 Wolfram
Fuente: WolframAlpha

Función 10 $f_{(x)} = x^3 - 3x$

Ubica un mínimo en (1;-2) y un máximo en (-1;2) como muestra la figura 78

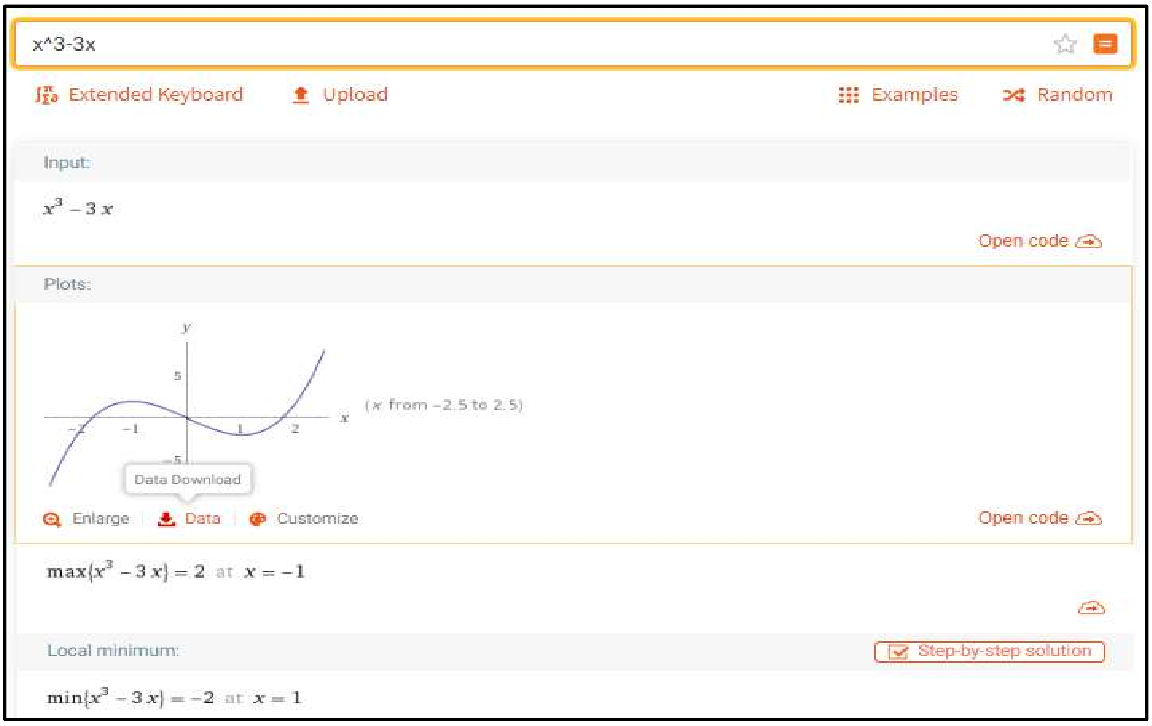


Figura N° 78 Función cúbica 5 - Wolfram
Fuente: WolframAlpha

Función 11 $f_{(x)} = x^3 - 7x + 6$

Ubica un mínimo en $x = \sqrt{\frac{7}{3}}$ y máximo en $x = -\sqrt{\frac{7}{3}}$ cuyos valores son equivalentes a $x = \pm 1{,}53$ como indica la figura 79

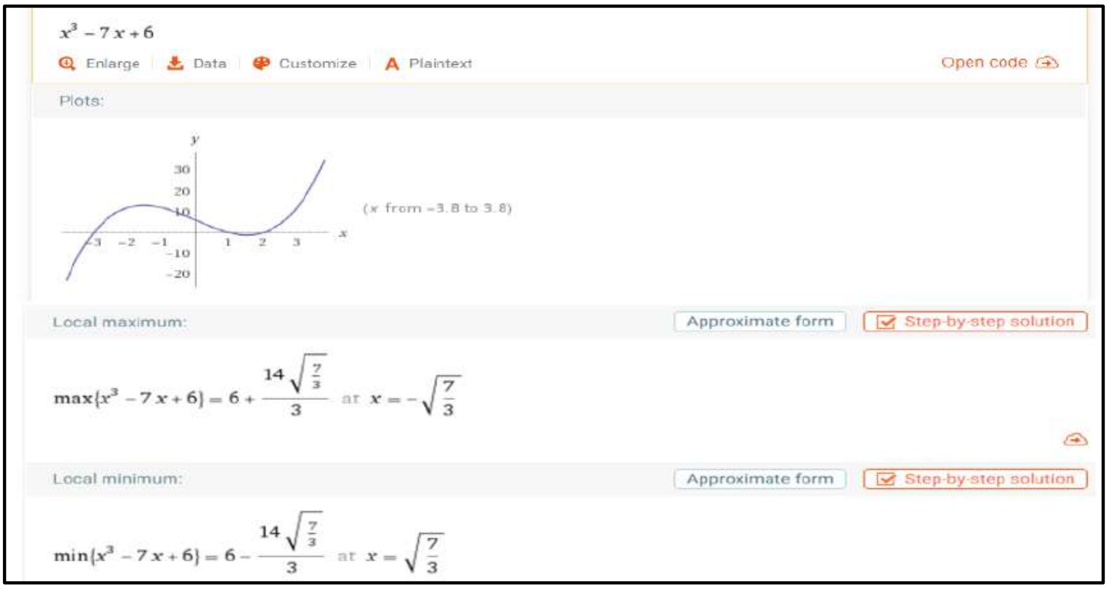


Figura N° 79 Función cúbica 6 Wolfram
Fuente: WolframAlpha

Función 12 $f_{(x)} = x^3 - 2x^2 + x$

Ubica un mínimo en (1,0) , un máximo en (0,33;0,14) como se ilustra en la figura 80

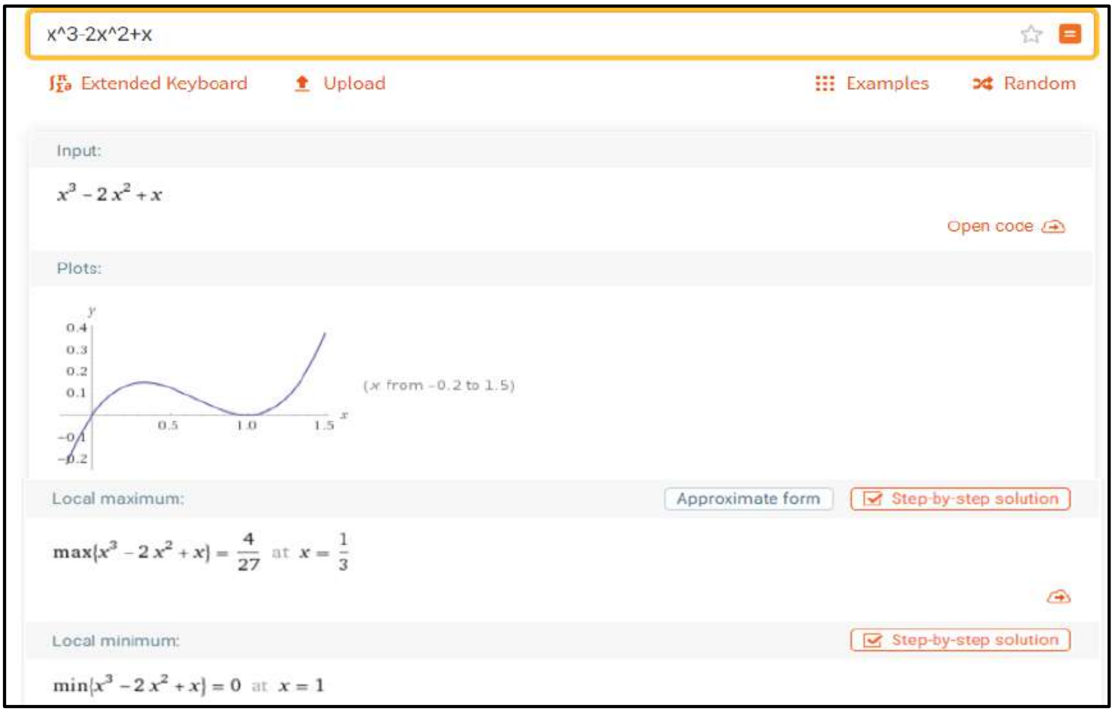


Figura N° 80 Función cúbica 7 Wolfram
Fuente: WolframAlpha

Función 13 $f_{(x)} = x^3 + 2x^2 - 3x - 2$

Ubica un mínimo en (0,54;-2,88) , un máximo en (-1,87;4,06) como muestra la figura 81.

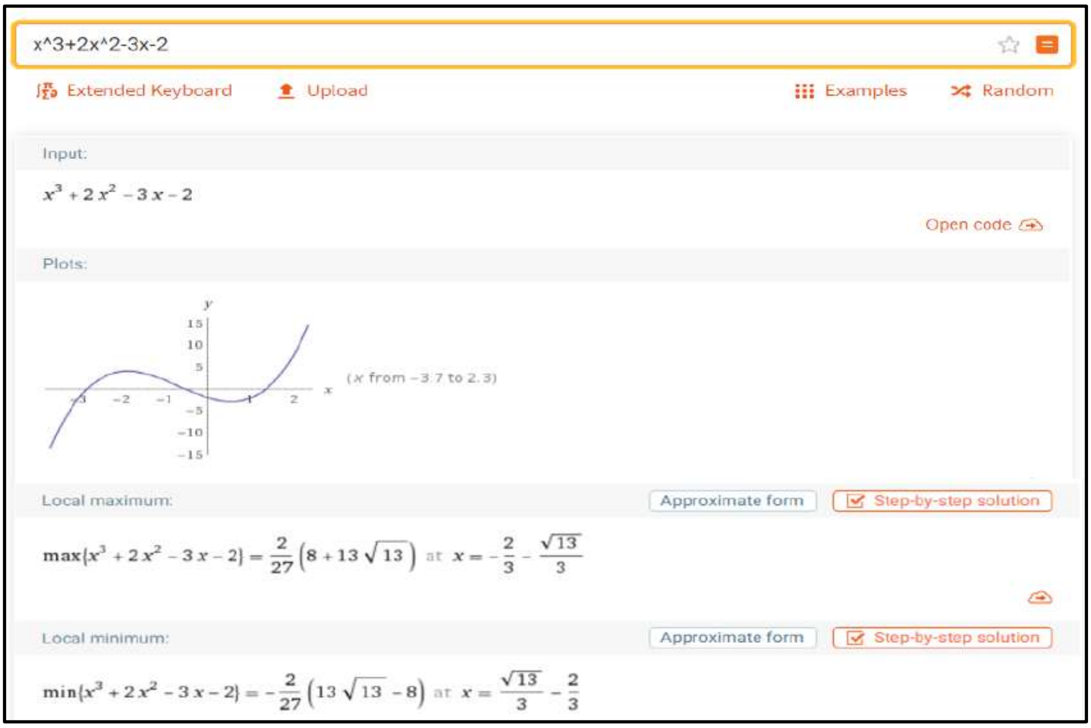


Figura N° 81 Función cúbica 8 Wolfram
Fuente: WolframAlpha

Función 14 $f_{(x)} = -1,2x^3 + 3$

No ubica máximos ni mínimos como indica la figura 82

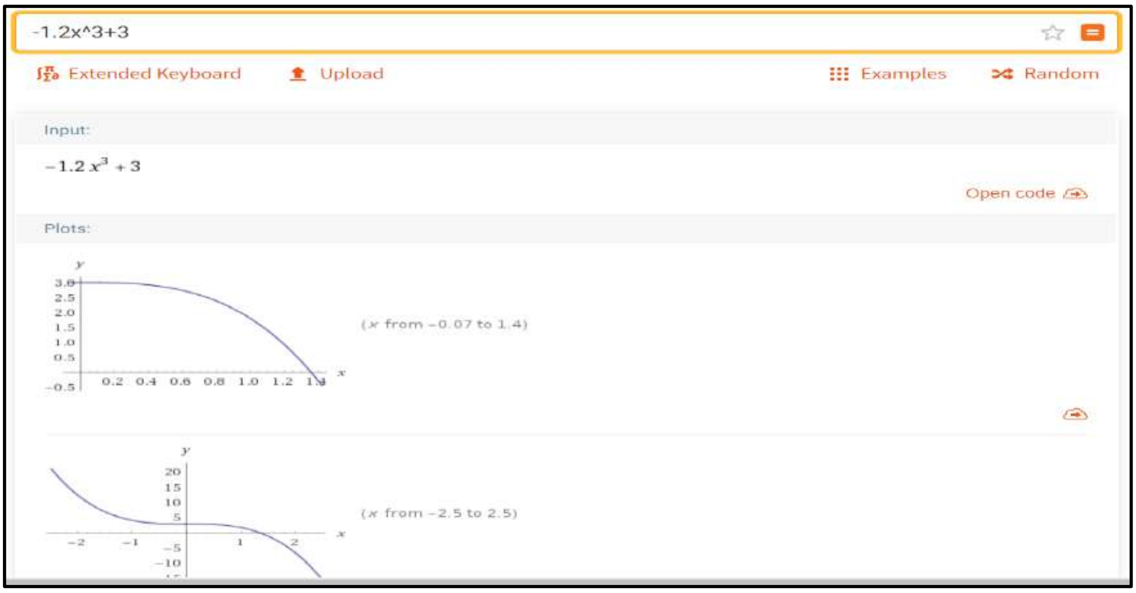


Figura N° 82 Función cúbica 9 Wolfram
Fuente: WolframAlpha

4.7.3.3 Ejecución con el software GeoGebra

Función 6 $f_{(x)} = x^3$

Ubica una inflexión en (0;0) como mmuestra la figura 83

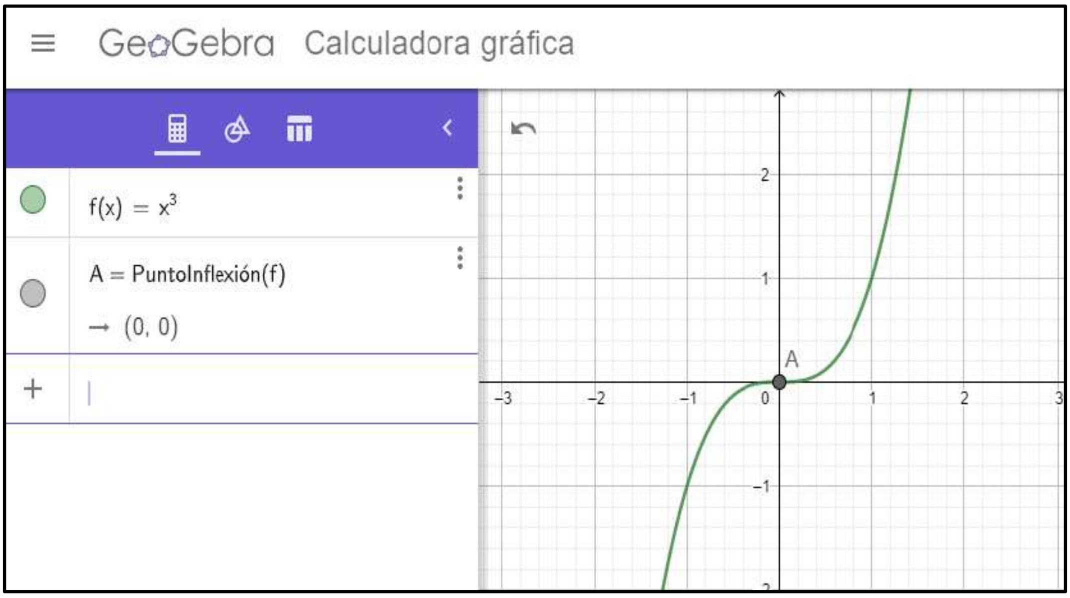


Figura N° 83 Función cúbica 1 GeoGebra
Fuente: GeoGebra

Función 7 $f_{(x)} = x^3 - 2x^2$

Ubica un máximo en (0;0) , un mínimo en (1,33; -1,18). Tal como muestra la figura 84

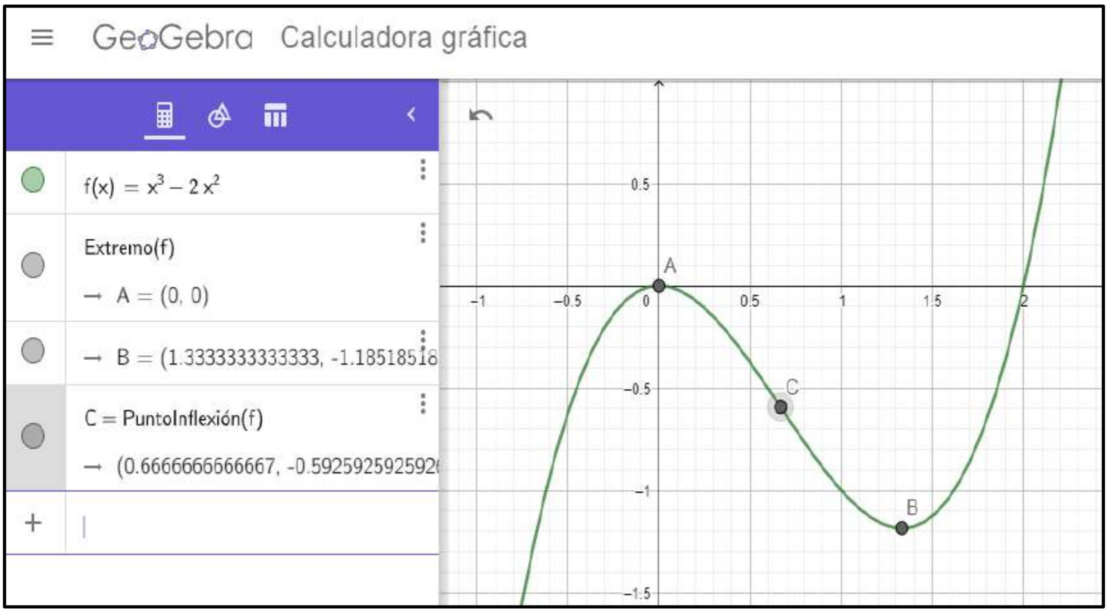


Figura N° 84 Función cubica 2 GeoGebra
Fuente: GeoGebra

Función 8 $f_{(x)} = x^3 - 4x^2 + 3$

Ubica un máximo en (0;3) , un mínimo en (2,667; -6,481) y el punto de inflexión en (1,33; -1,7407) como muestra la figura 85

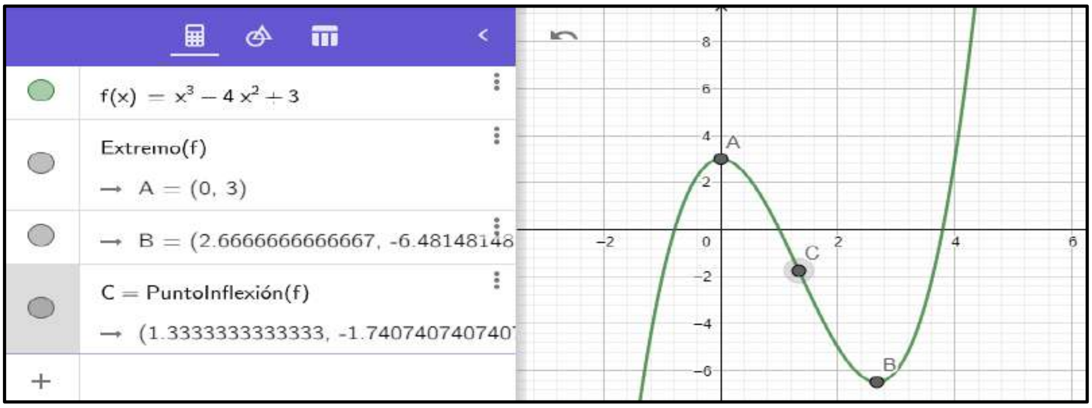


Figura N° 85 Función cúbica 3 GeoGebra
Fuente: GeoGebra

Función 9 $f_{(x)} = x^3 + 3x$

Ubica un punto de inflexión en (0;0) como muestra la figura 86

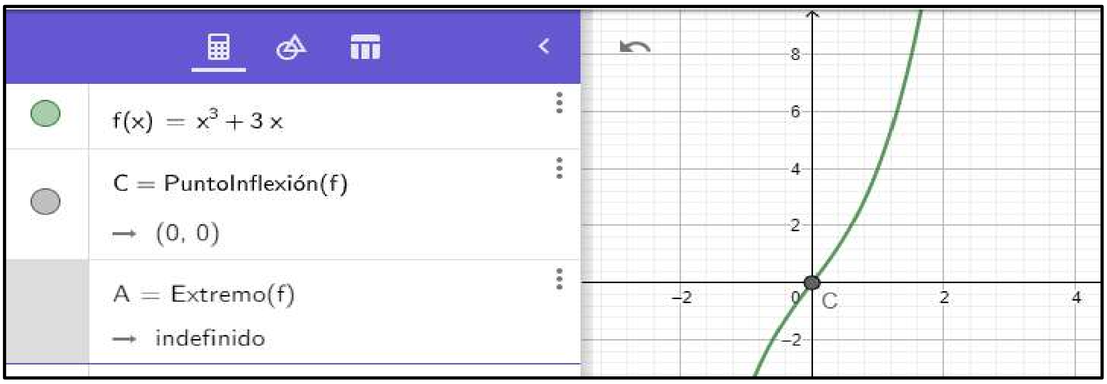


Figura N° 86 Función cúbica 4 GeoGebra
Fuente: GeoGebra

Función 10 $f_{(x)} = x^3 - 3x$

Ubica un máximo en (-1;2) , un mínimo en (1;-2) así como un punto de inflexión en (0;0) como ilustra la figura 87

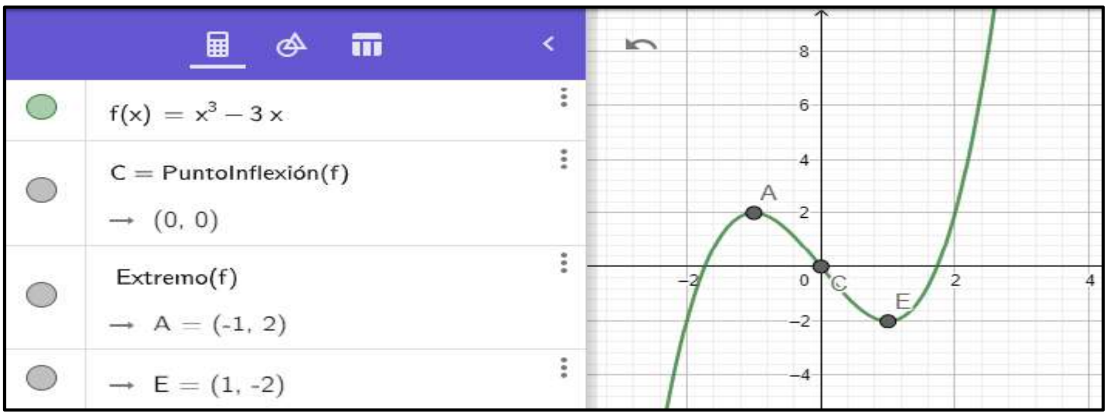


Figura N° 87 Función cúbica 5 - GeoGebra
Fuente: GeoGebra

Función 11 $f_{(x)} = x^3 - 7x + 6$

Ubica un máximo en (-1,527;13,128) , un mínimo en (1,527;-1,128) así como una inflexión en (0;6) como muestra la figura 88

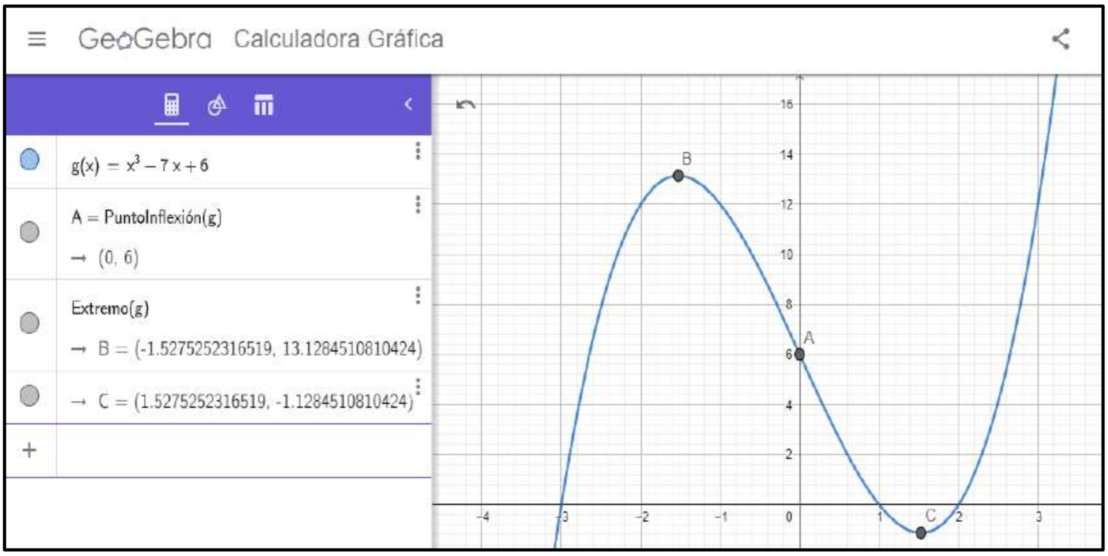


Figura N° 88 Función cúbica 6 GeoGebra
Fuente: GeoGebra

Función 12 $f_{(x)} = x^3 - 2x^2 + x$

Ubica un mínimo en (1,0), un máximo en (0,33;0,14) así como una inflexión en (0,667;0,074) como se ilustra en la figura 89

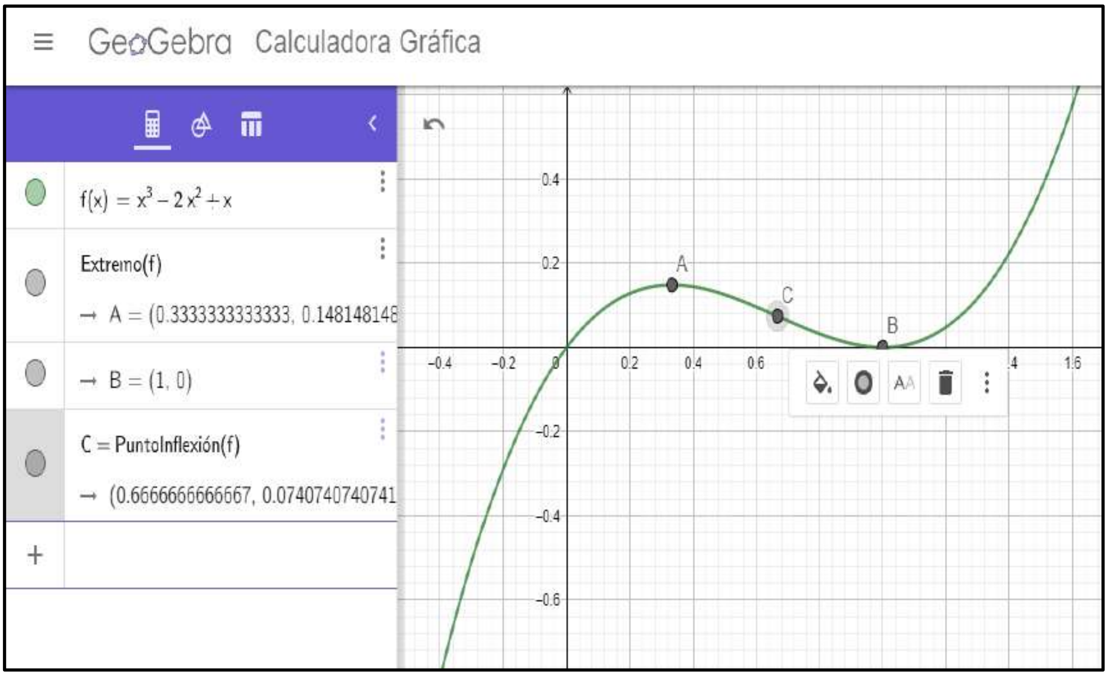


Figura N° 89 Función cúbica 7 GeoGebra
Fuente: GeoGebra

Función 13 $f_{(x)} = x^3 + 2x^2 - 3x - 2$

Ubica un máximo en (-1,868;4,065), un mínimo en (0,535;-2,879) así como una inflexión en (-0,667;0,593) como se ilustra en la figura 90

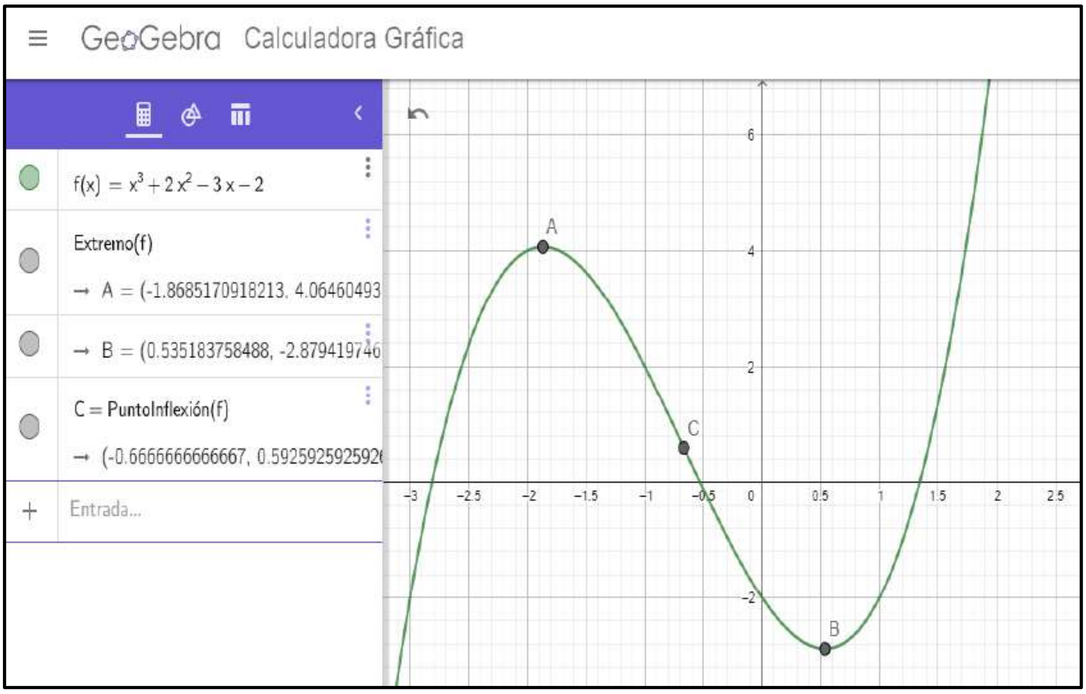


Figura N° 90 Función cúbica 8 - GeoGebra
Fuente: GeoGebra

Función 14 $f_{(x)} = -1,2x^3 + 3$

Ubica una inflexión en (0;3) como muestra la figura 91

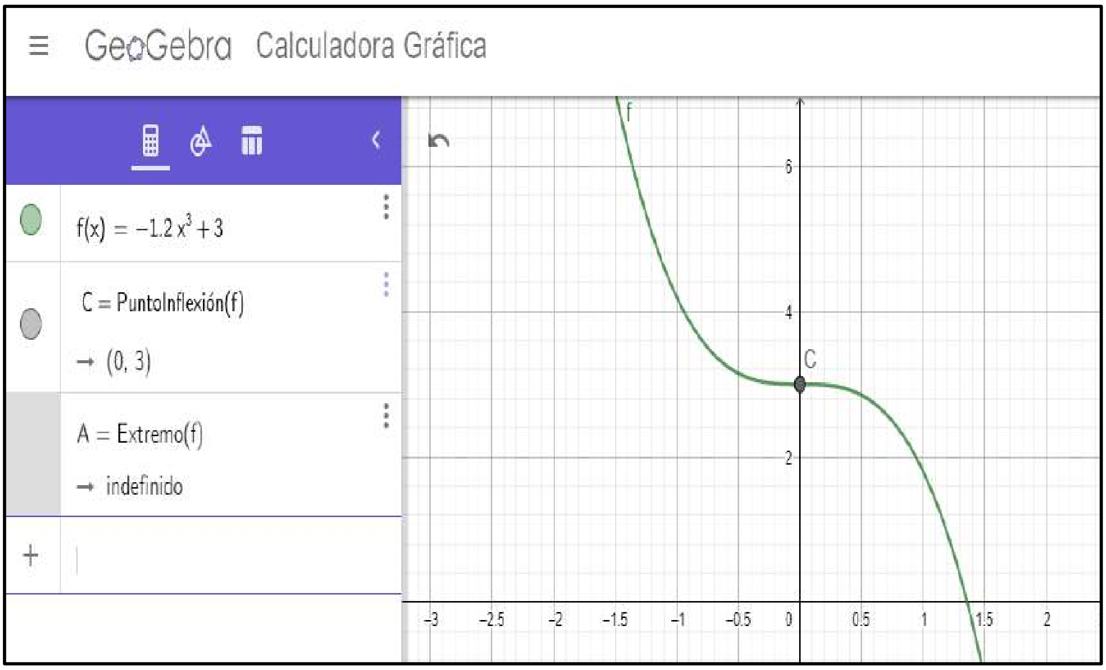


Figura N° 91 Función cúbica 9 GeoGebra
Fuente: GeoGebra

4.7.3.4 Ejecución con el software Symbolab

Función 6 $f_{(x)} = x^3$

Obtiene un punto de silla en el punto (0,0) que no corresponde a un máximo o mínimo luego de aplicar el criterio de la primera derivada como indica la figura 92

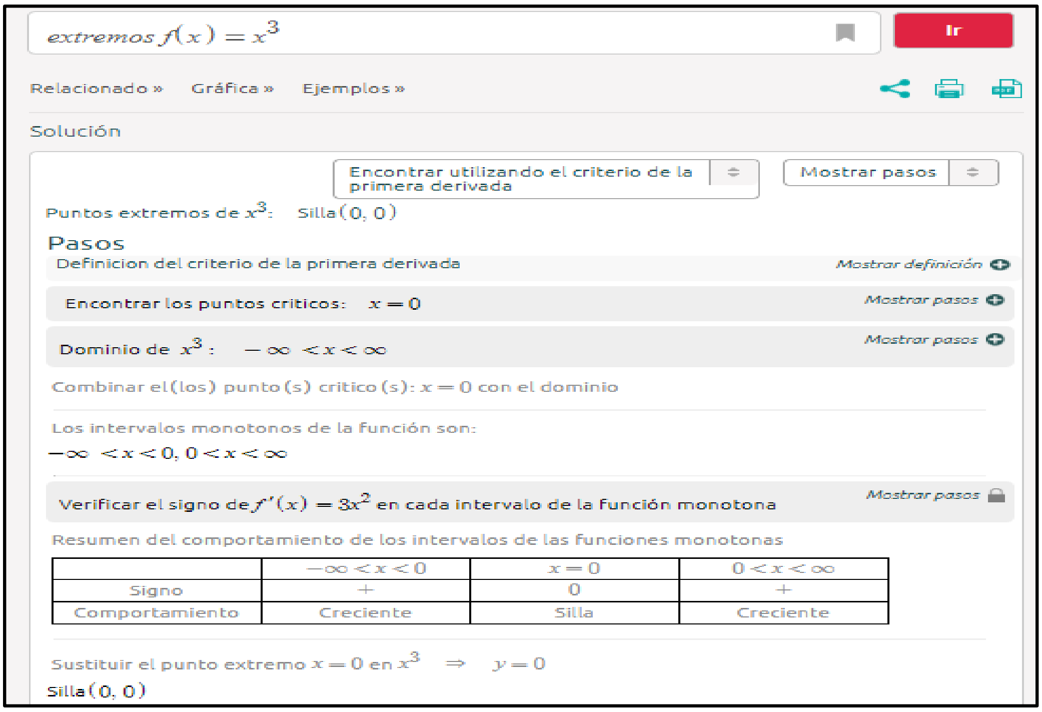


Figura N° 92 Función cúbica 1 Symbolab 1
Fuente: Symbolab

Sin embargo, si utiliza el criterio de la segunda muestra que el criterio falla o es **poco concluyente** como indica la figura 93

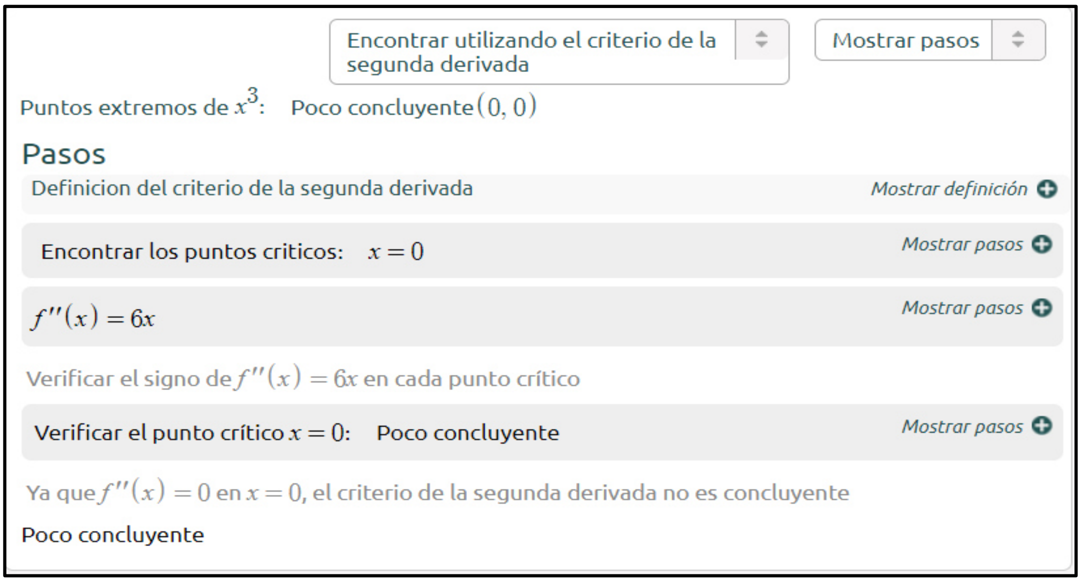


Figura N° 93 Función cúbica 1 Symbolab 2
Fuente: Symbolab

Función 7 $f_{(x)} = x^3 - 2x^2$

Ubica un máximo en (0;0) y un mínimo en $\left(\frac{4}{3}; \frac{-32}{27}\right)$ como muestra la figura 94

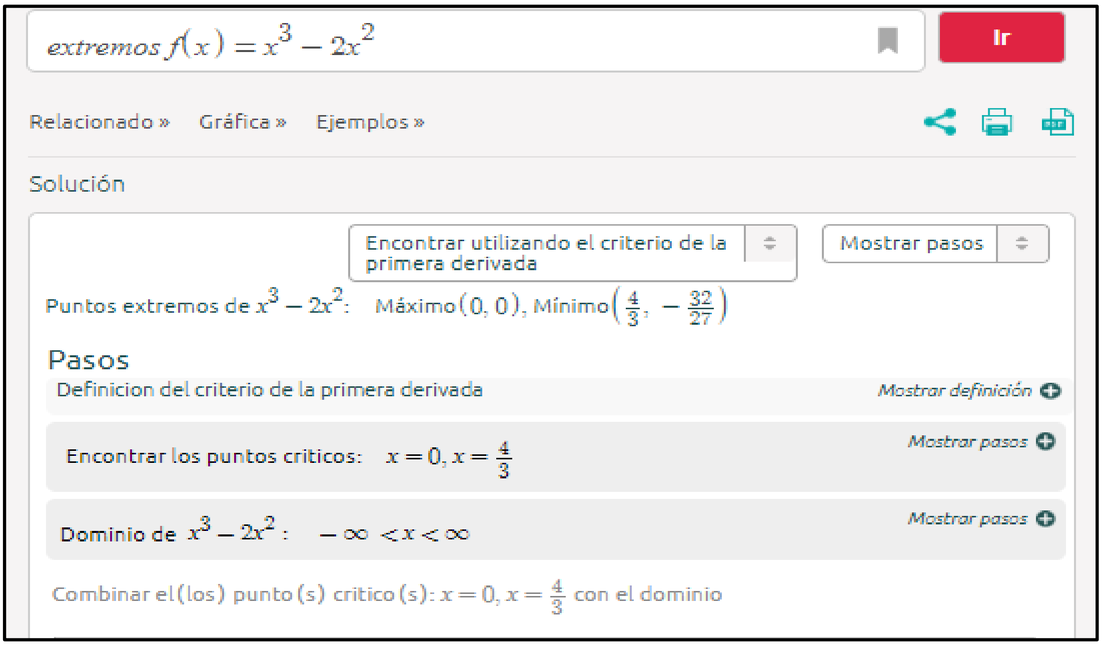


Figura N° 94 Función cúbica 2 Symbolab
Fuente: Symbolab

Función 8 $f_{(x)} = x^3 - 4x^2 + 3$

Ubica un máximo en (0;3), un mínimo en $\left(\frac{8}{3}; \frac{-175}{27}\right)$ como muestra la figura 95

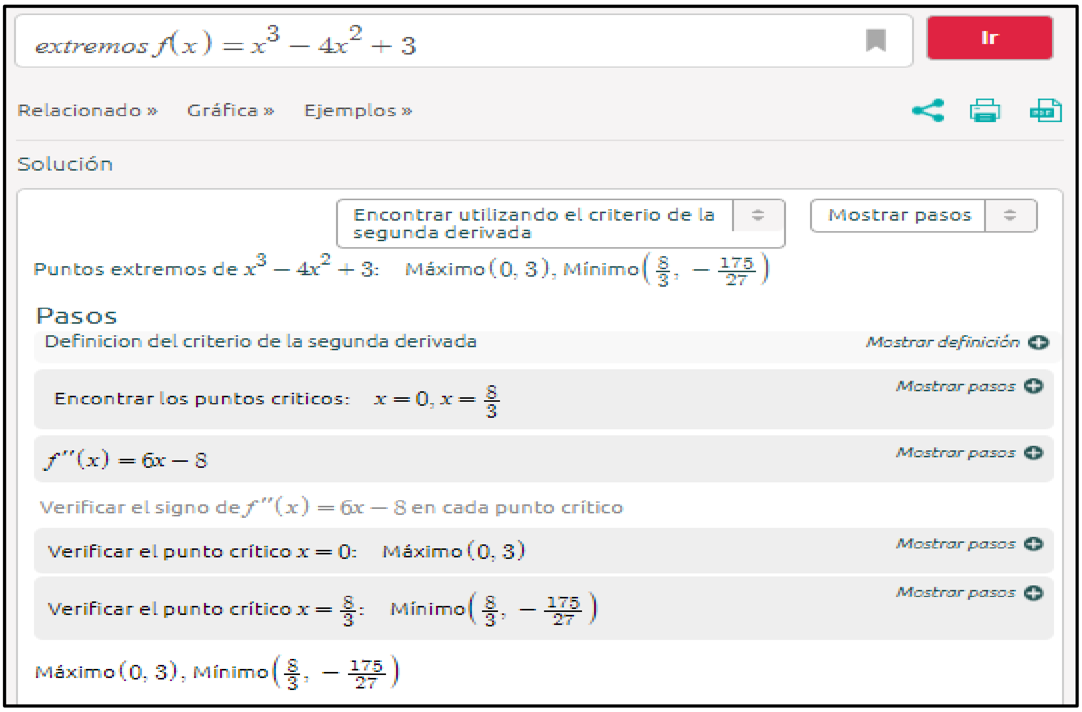


Figura N° 95 Función cúbica 3 Symbolab
Fuente: Symbolab

Función 9 $f_{(x)} = x^3 + 3x$

No ubica máximos ni mínimos como ilustra la figura 96

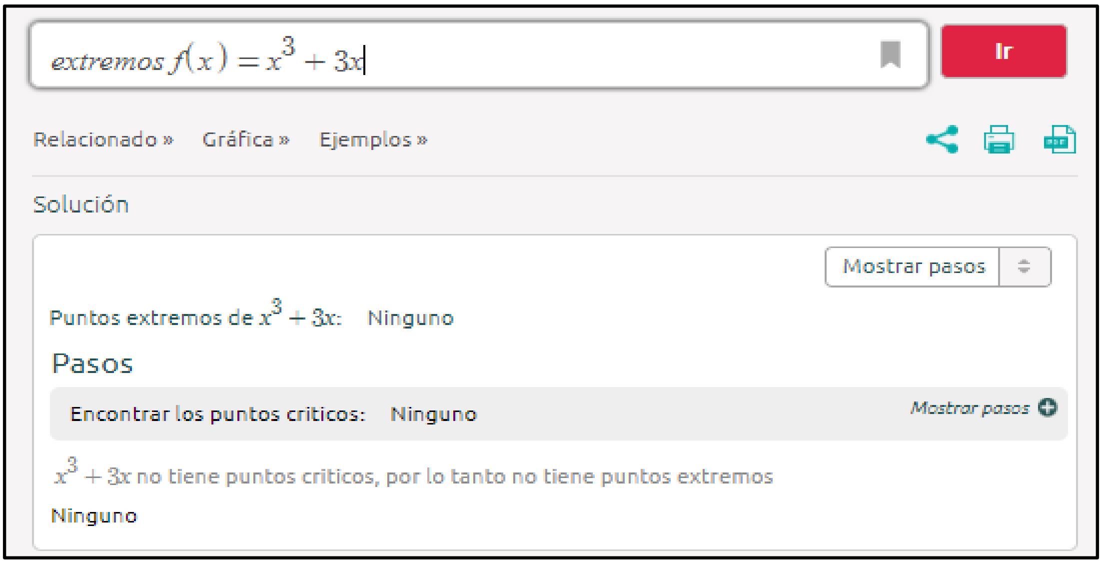


Figura N° 96 Función cúbica 4 Symbolab
Fuente: Symbolab

Función 10 $f_{(x)} = x^3 - 3x$

Ubica un máximo en (-1;2), un mínimo en (1;-2) como muestra la figura 97

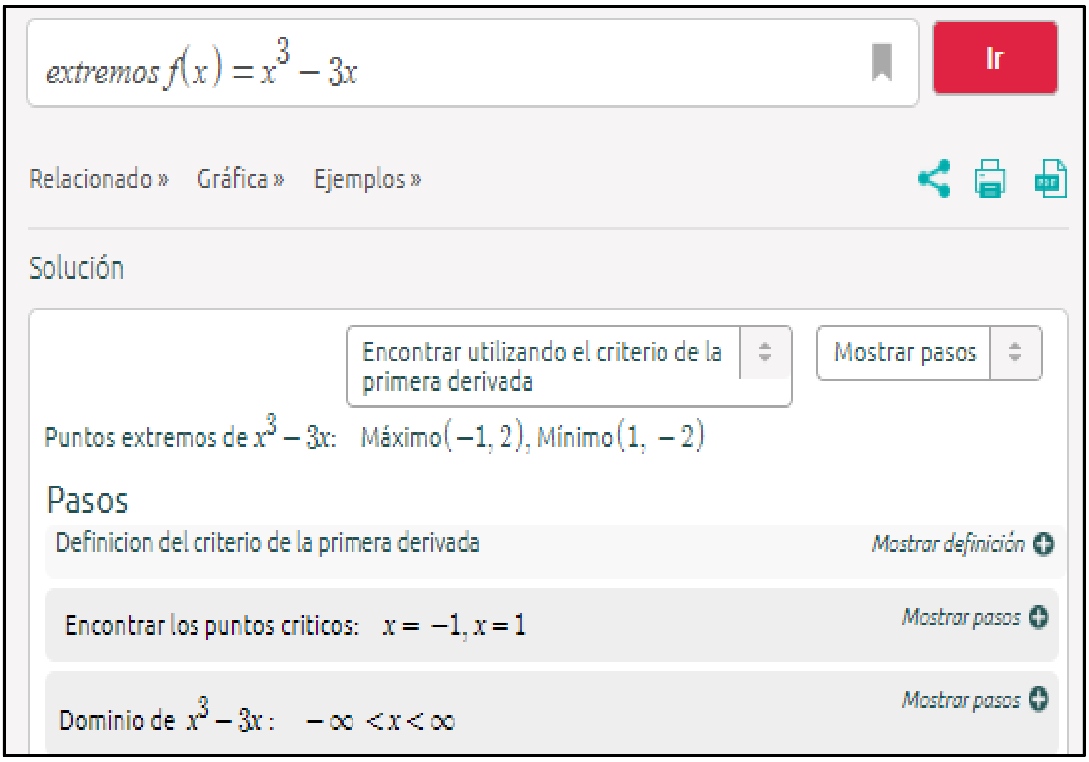


Figura N° 97 Función cúbica 5 Symbolab
Fuente: Symbolab

Función 11 $f_{(x)} = x^3 - 7x + 6$

Ubica un máximo $x = -\sqrt{\frac{7}{3}}$ y un mínimo en $x = \sqrt{\frac{7}{3}}$, como muestra la figura 98. La correspondencia del punto requiere cálculo adicional

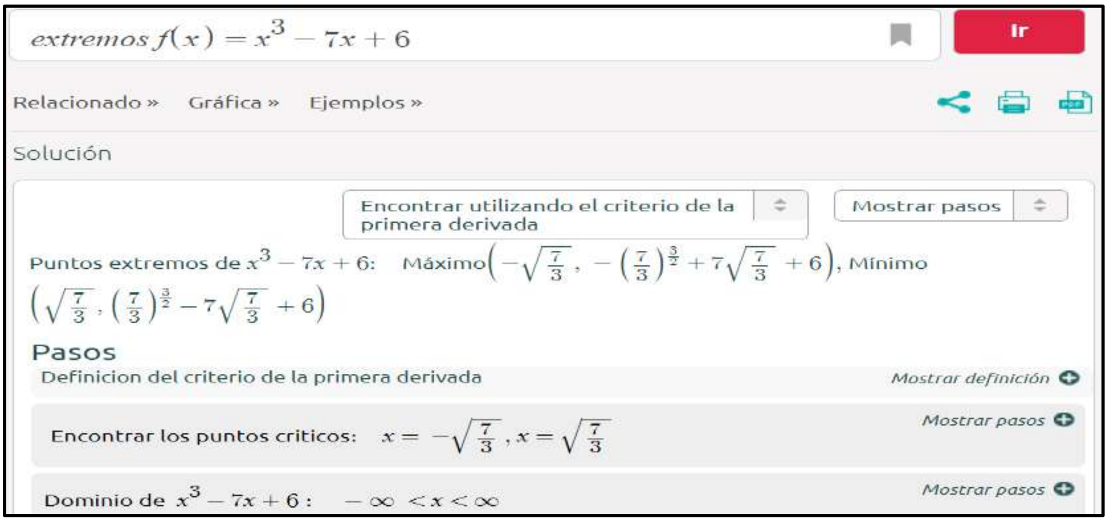


Figura N° 98 Función cúbica 6 - Symbolab
Fuente: Symbolab

Función 12 $f_{(x)} = x^3 - 2x^2 + x$

Ubica un máximo en $\left(\frac{1}{3}; \frac{4}{27}\right)$ y un mínimo en (1;0) como muestra la figura 99

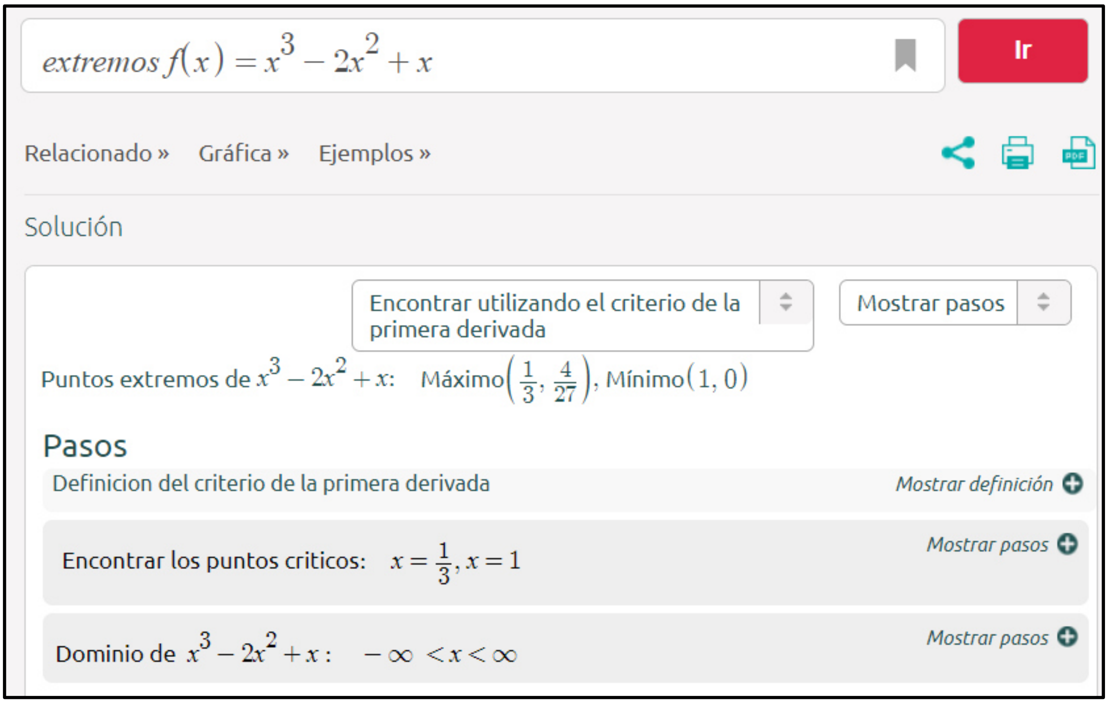


Figura N° 99 Función cúbica 7 Symbolab
Fuente: Symbolab

Función 13 $f_{(x)} = x^3 + 2x^2 - 3x - 2$

Ubica un máximo en $x = -\frac{2+\sqrt{13}}{2}$ y un mínimo en $x = \frac{-2+\sqrt{13}}{2}$ como se muestra en la figura 100

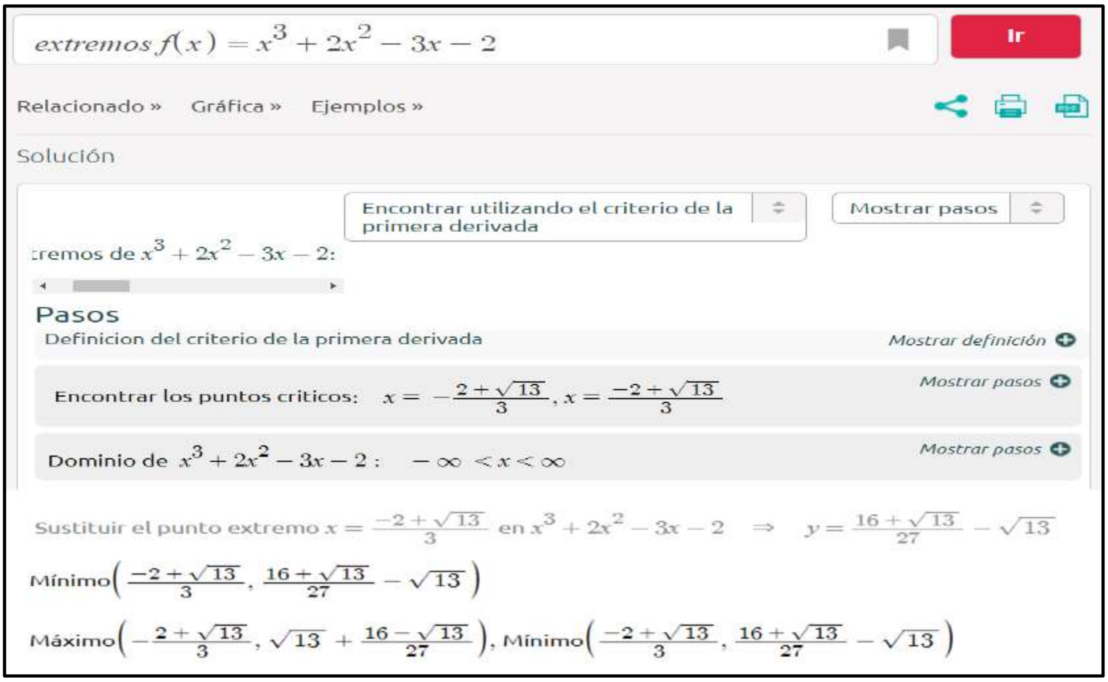


Figura N° 100 Función cúbica 8 - Symbolab
Fuente: Symbolab

Función 14 $f_{(x)} = -1,2x^3 + 3$

Ubica un punto de silla en (0;3) como muestra la figura 101

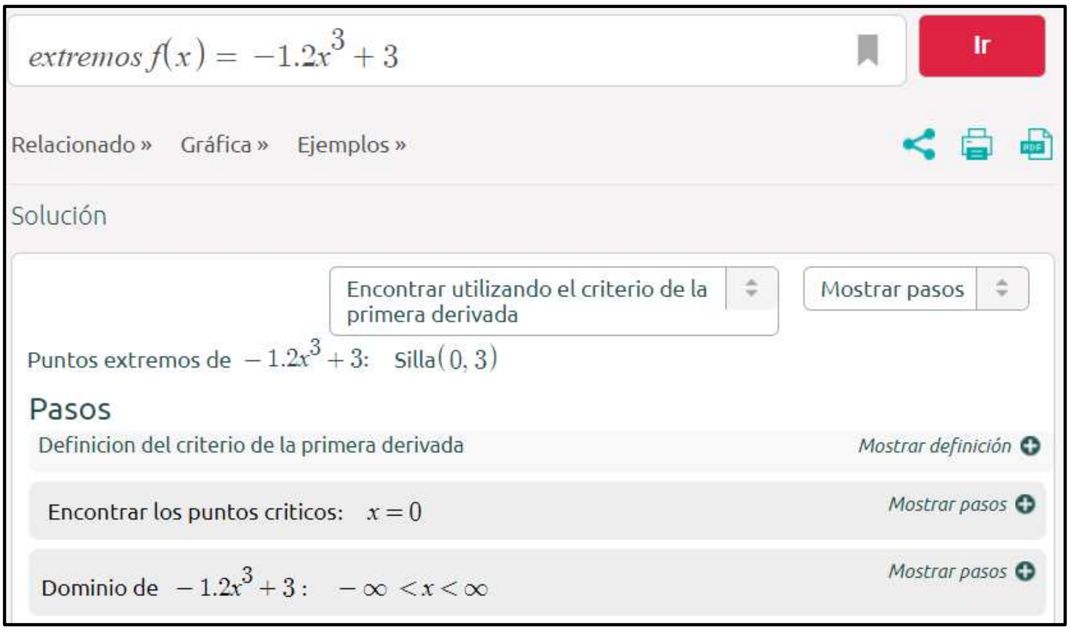


Figura N° 101 Función cúbica 9 Symbolab
Fuente: Symbolab

4.7.4 Prueba para funciones de grado 4

4.7.4.1 Ejecución con el software DERIN

Función 15 $f_{(x)} = x^4 + 1$

Obtiene un valor mínimo en el punto (0,1) como indica la figura 102

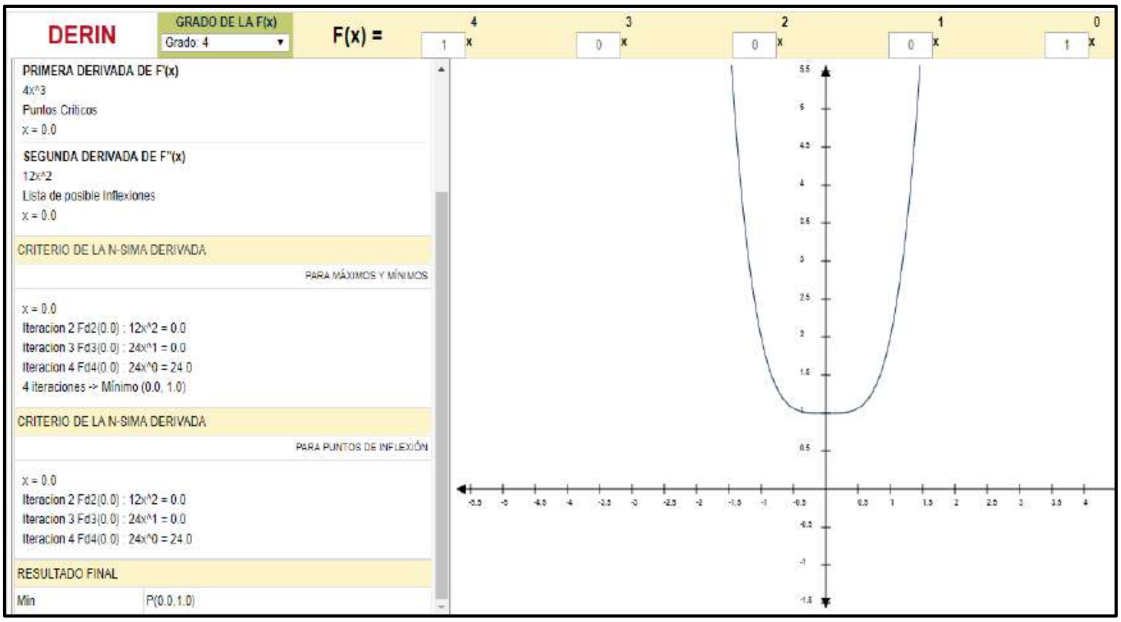


Figura N° 102 Función polinómica 1
Fuente: DERIN

Función 16 $f_{(x)} = x^4 + 2x^3$

Ubica un mínimo en (-1,5; -1,68), una inflexión de primera especie en (0;0) y una inflexión de segunda especie en (-1;-1) como muestra la figura 103

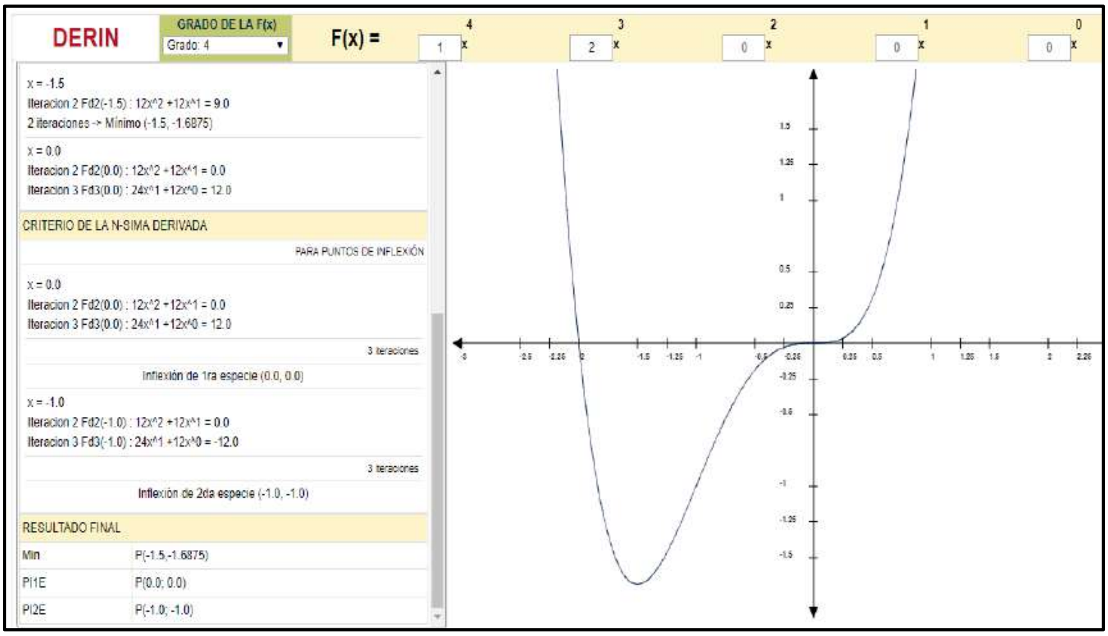


Figura N° 103 Función polinómica 2
Fuente: DERIN

Función 17 $f_{(x)} = x^4 - 2x^3$

Ubica un mínimo en (1,5;-1,81) , una inflexión de primera especie en (1;-1) y una inflexión de segunda especie en (0;0) como muestra la figura 104

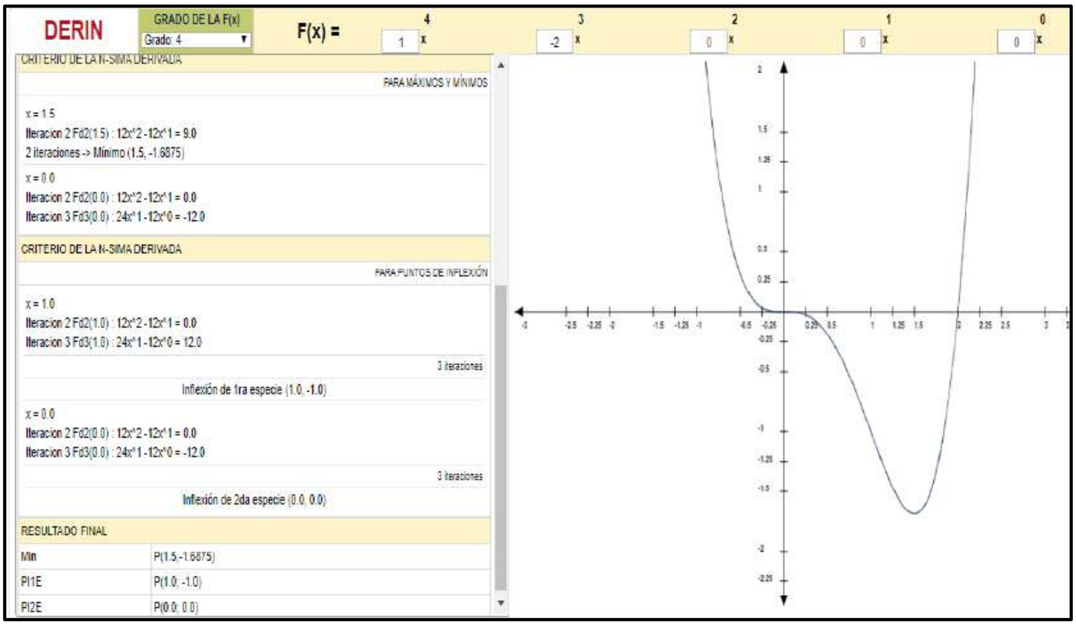


Figura N° 104 Función polinómica 3
Fuente: DERIN

Función 18 $f_{(x)} = x^4 - 2x^2$

Ubica un mínimo en (-1; -1) y (1,-1) , un máximo en (0;0) , una inflexión de primera especie en (0,58;-0,56) y una inflexión de segunda especie en (0,58;-0,56) tal como muestra la figura 105

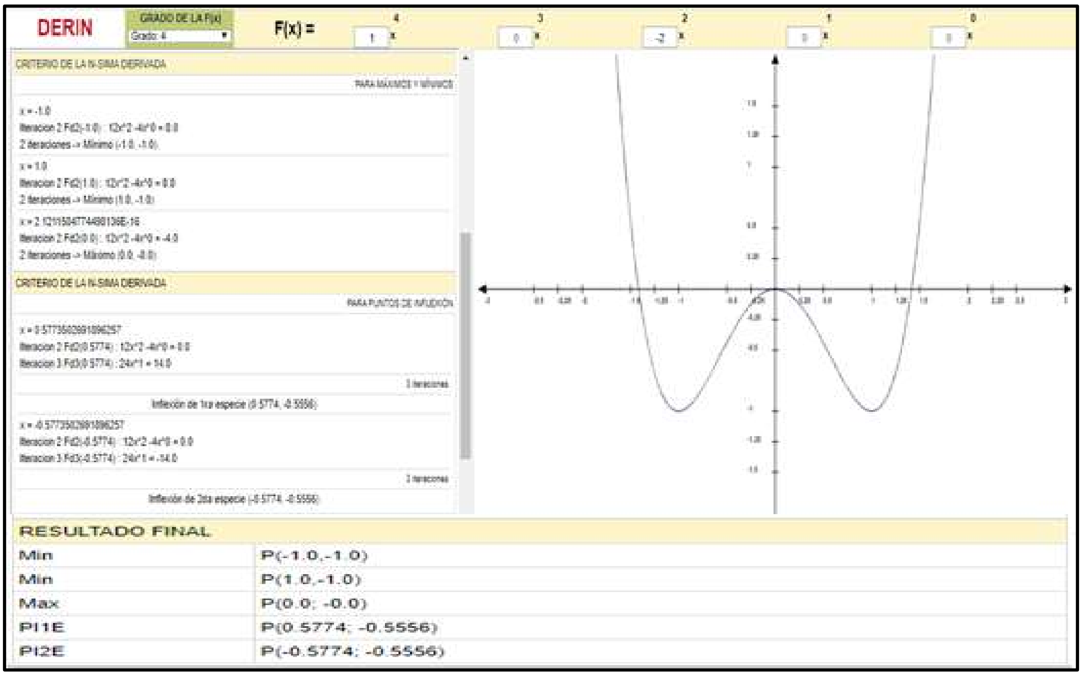


Figura N° 105 Función polinómica 4
Fuente: software DERIN

Funcion 19 $f_{(x)} = x^4 - 2x$

Obtiene un mínimo en (0,79;-1,19) como resume la figura 106

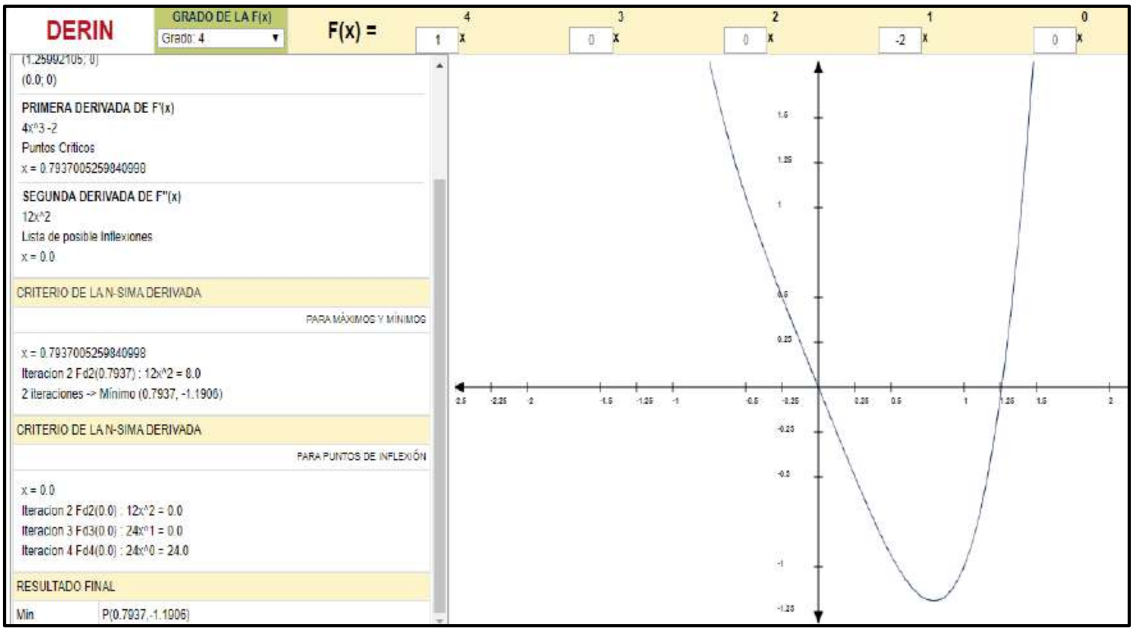


Figura N° 106 Función polinómica 5
Fuente: software DERIN

Funcion 20 $f_{(x)} = x^4 - 2x^3 - 3x^2$

Ubica un máximo en (0;0) y dos mìnimos en (-0,68; -0,54) y (2,18; -12,39) ,y dos inflexiones en (-0,36; -0,28) y (1,36;-7,21) como muestra la figura 107

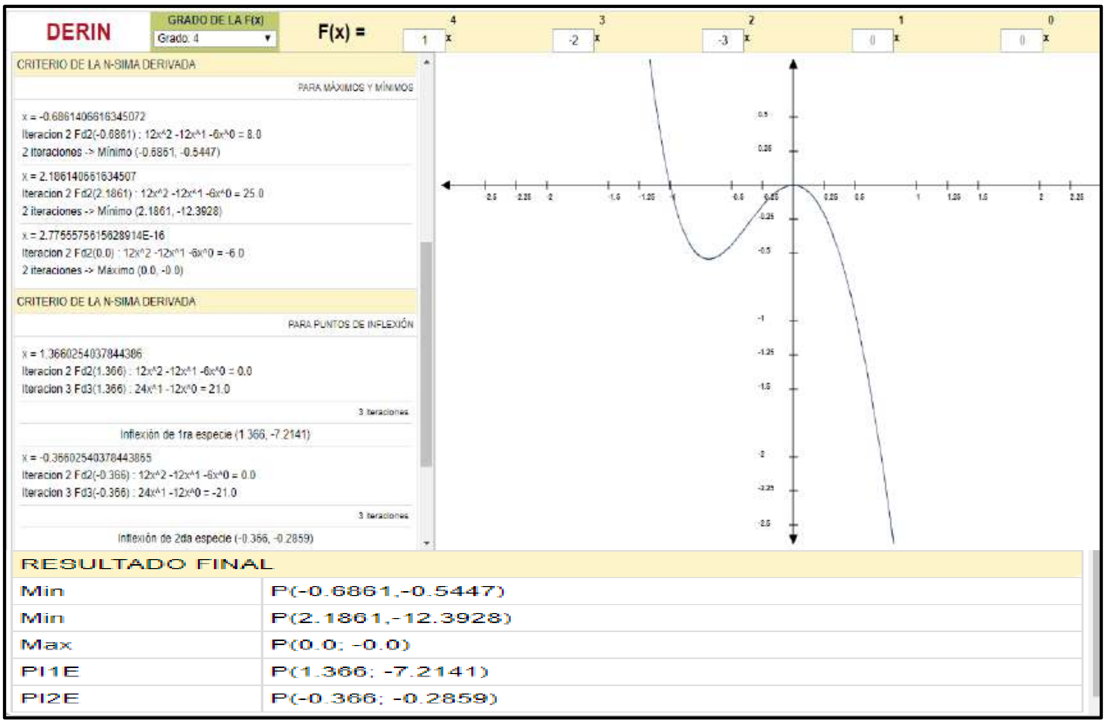


Figura N° 107 Función polinómica 6
Fuente: DERIN

Funcion 21 $f_{(x)} = x^4 - x^3 - 3x$

Ubica un mínimo en (1,24;-3,26) , una inflexión de segunda especie en (0,0) y en (0,5;-1,56) una inflexión de primera especie como ilustra la figura 108

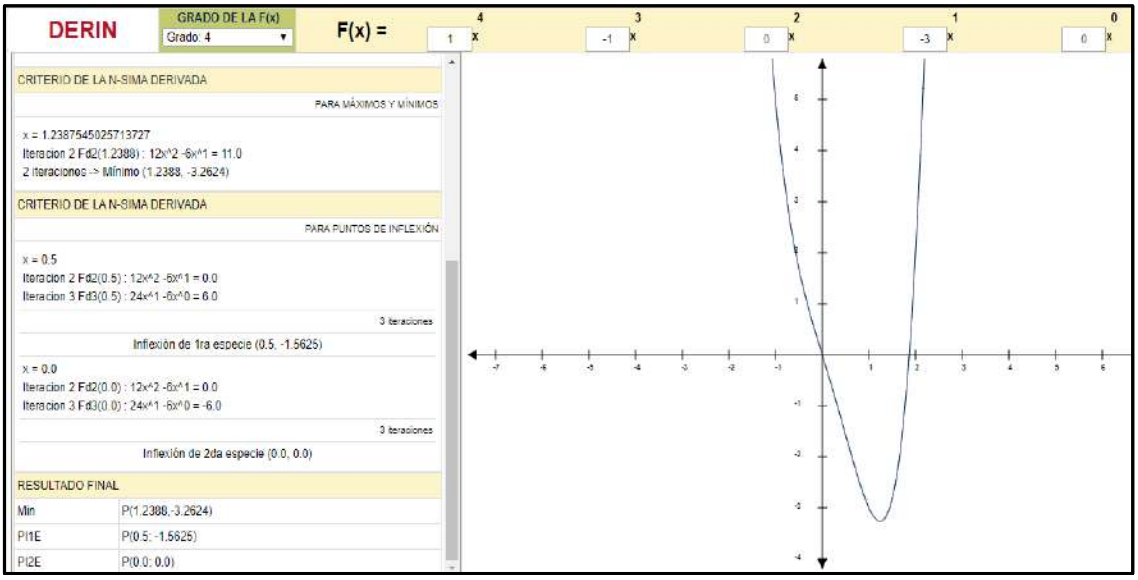


Figura N° 108 Función polinómica 7
Fuente: DERIN

Funcion 22 $f_{(x)} = x^4 - x^3 - 2$

Ubica un mínimo en (0,75;-2,11) , una inflexión de segunda especie en (0;-2) y en (0,5;-2,0625) una inflexión de segunda especie como muestra la figura 109

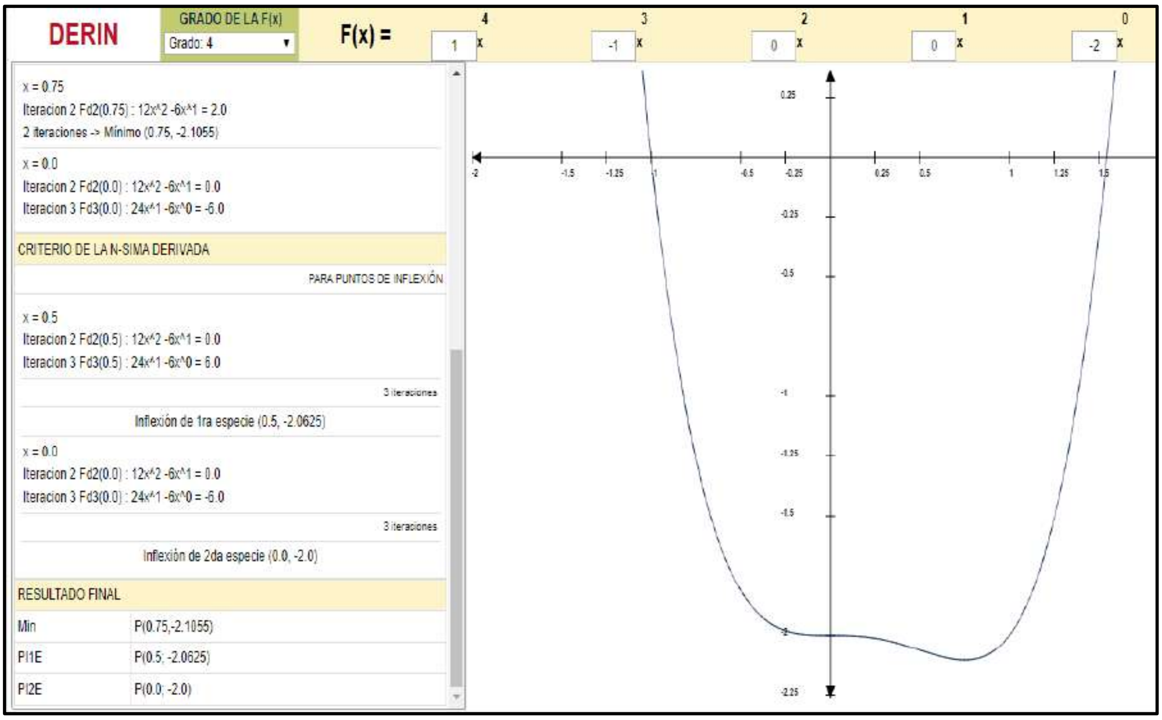


Figura N° 109 Función polinómica 8
Fuente: DERIN

Funcion 23 $f_{(x)} = x^4 - 2x^2 - 3x$

Ubica un mínimo en (1,26;-4,43) , una inflexión de primera especie en (0,57,-2,28) y de segunda especie en (-0,57;1,17) como se muestra en la figura 110

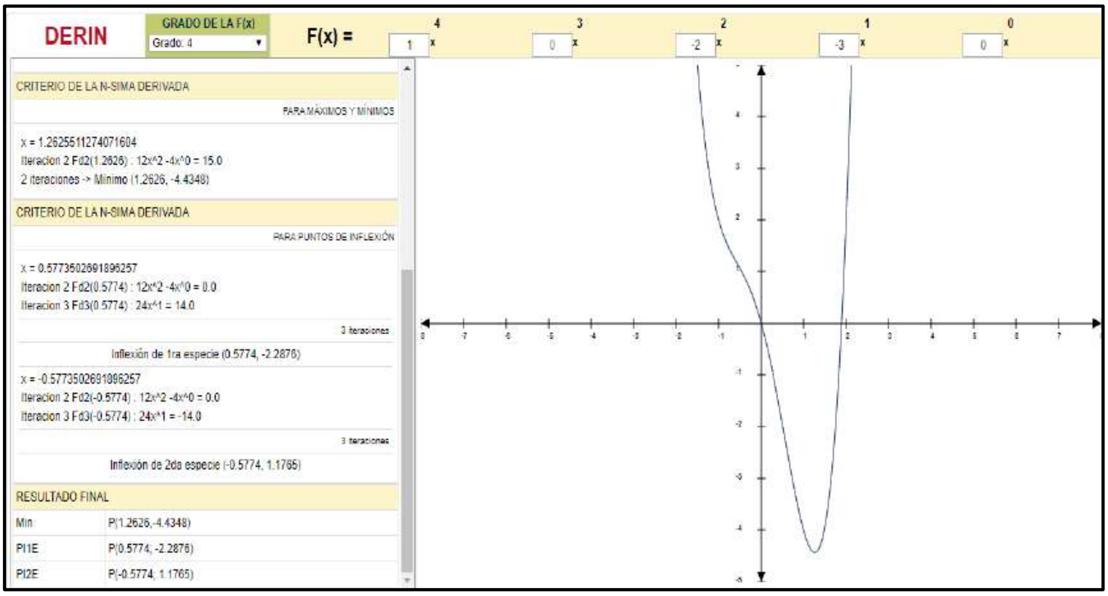


Figura N° 110 Función polinómica 9
Fuente: DERIN

Funcion 24 $f_{(x)} = x^4 - 3x^2 - 2$

Ubica mínimos en (-1,22;-4,25) y en (1,22;-4,25), un máximo en (0;-2) y dos inflexiones en (0,71;-3,25) y en (-0,71;-3,25) tal como indica la figura 111

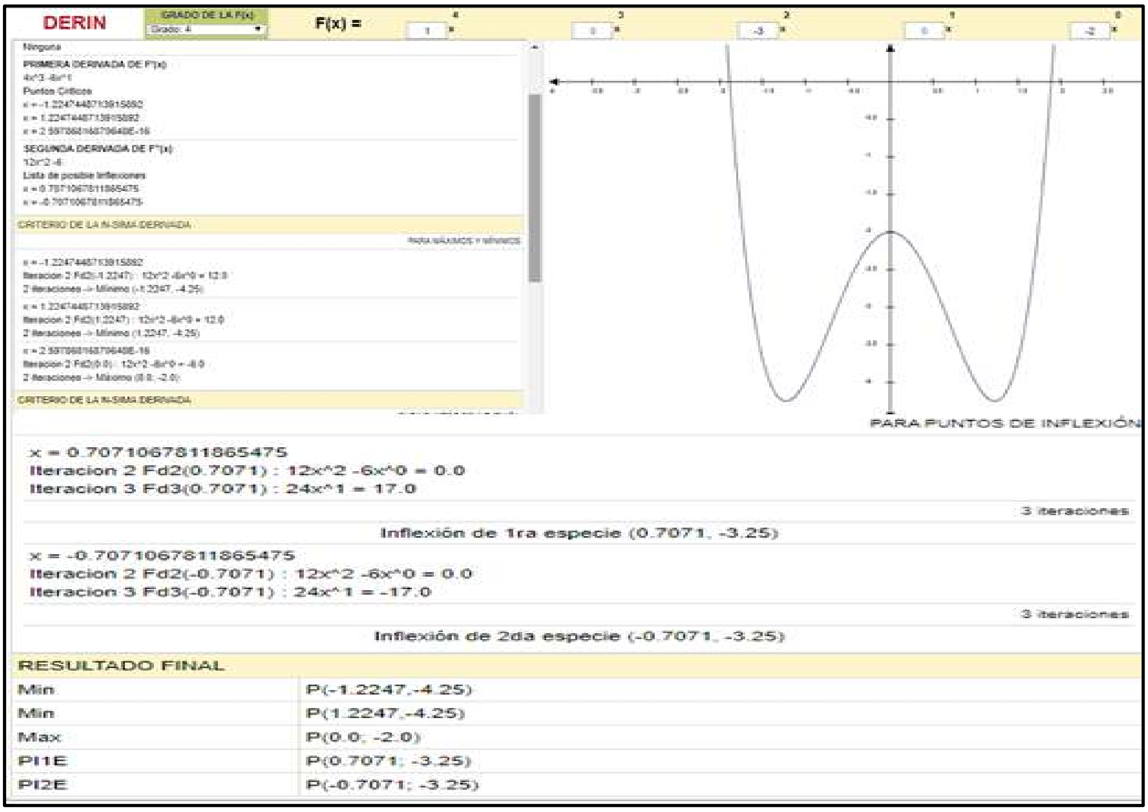


Figura N° 111 Función polinómica 10
Fuente: DERIN

Funcion 25 $f_{(x)} = x^4 - 4x + 3$

Ubica un mínimo en (1,0) como indica la figura 112

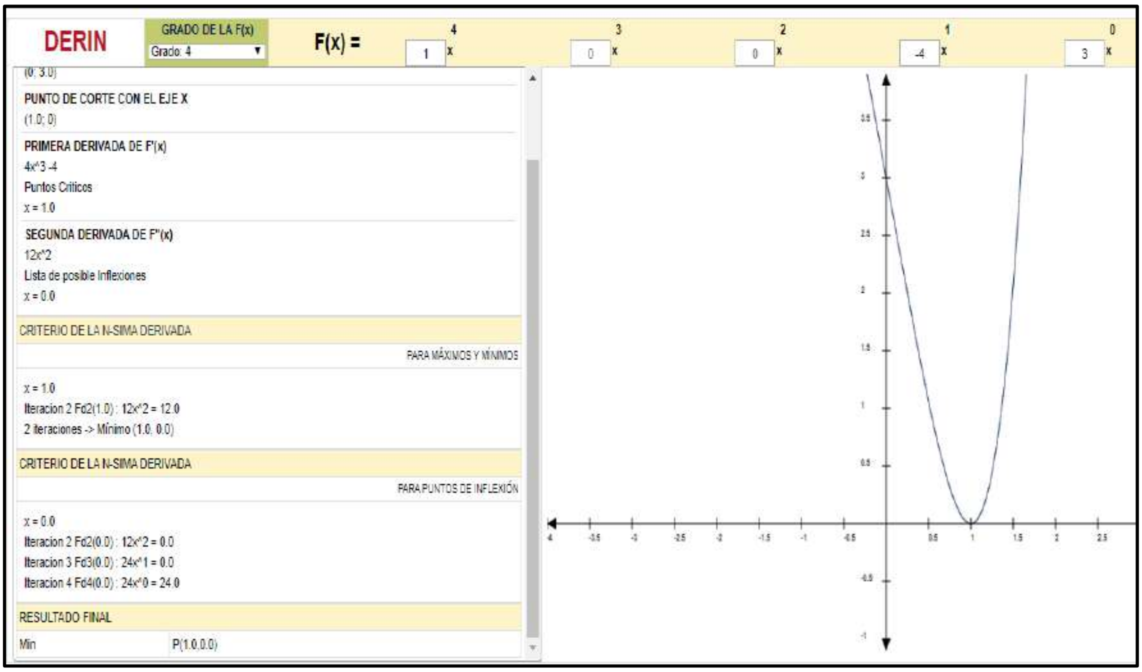


Figura N° 112 Función polinómica 11
Fuente: DERIN

Funcion 26 $f_{(x)} = x^4 - x^3 - 2x^2 - 3x$

Ubica un mínimo en (1,64;-7,47) y dos inflexiones en (0,88;-4,26) y (-0,38;0,925) como muestra la figura 113

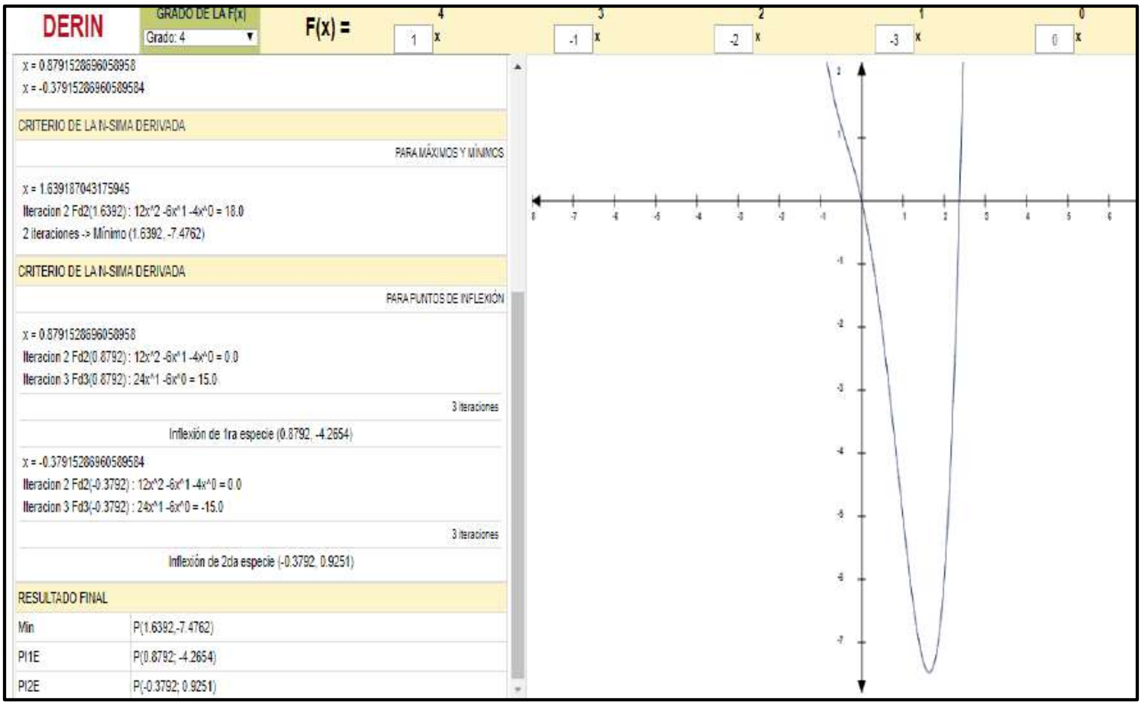


Figura N° 113 Función polinómica 12
Fuente: DERIN

Funcion 27 $f_{(x)} = x^4 - 3x^3 - 2x - 2$

Ubica un mínimo en (2,34; -15,14) y dos inflexiones en (1,5;-10,06) y (0;-2) de primera y segunda especie respectivamente como muestra la figura 114

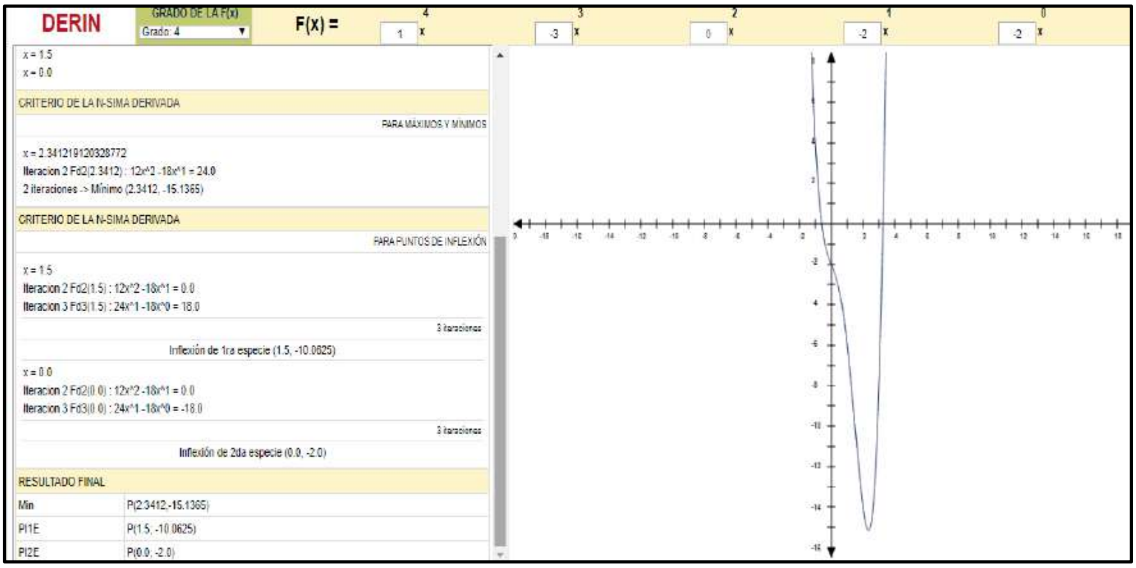


Figura N° 114 Función polinómica 13
Fuente: DERIN

Funcion 28 $f_{(x)} = x^4 - 3x^2 - x - 2$

Ubica mínimos en (-1,13;-3,07) y en (1,3; -5;51) , un máximo en (-0,17;-1,92) y inflexiones en (0,71;-3,95) y (-0,71;-2,54) como muestra la figura 115

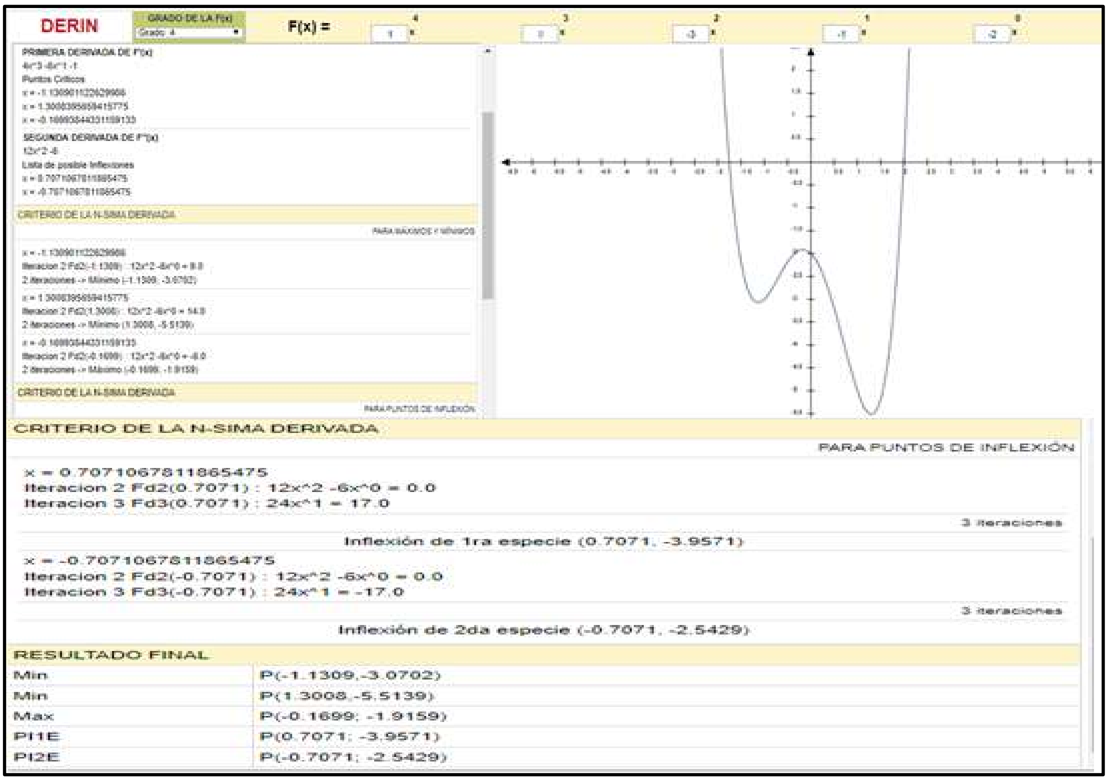


Figura N° 115 Función polinómica 14
Fuente: DERIN

Funcion 29 $f_{(x)} = x^4 - x^3 - 2x^2 - x - 2$

Ubica un mínimo en (1,52;-6,31) y dos inflexiones en (0,88;-4,51) y (-0,38;-1,83) como muestra la figura 116

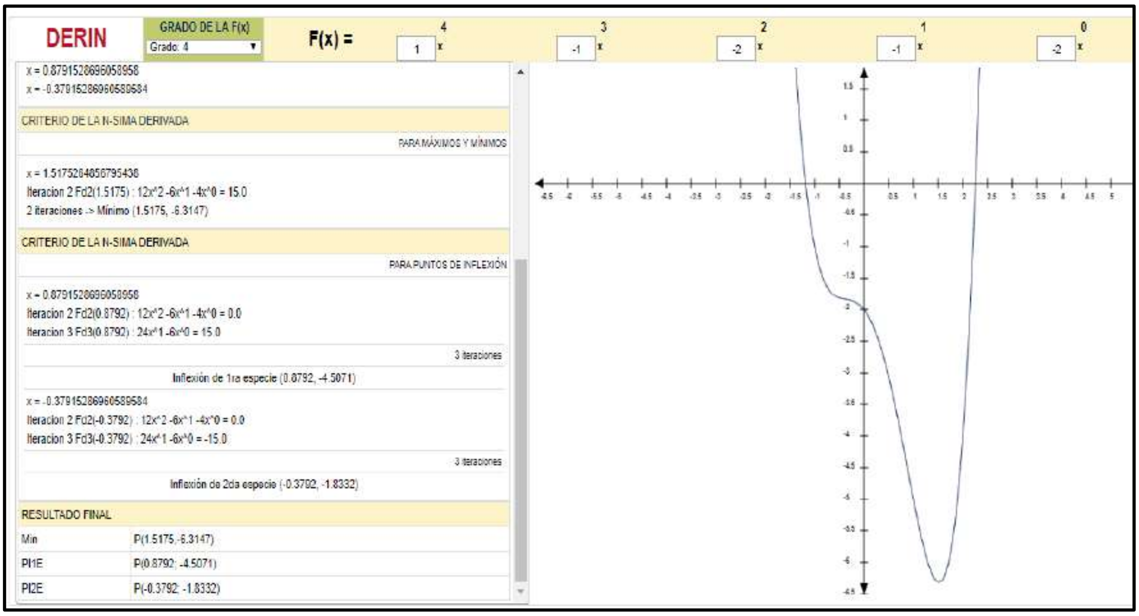


Figura N° 116 Función polinómica 15
Fuente: DERIN

Funcion 30 $f_{(x)} = -x^4 + 1,8x^3 + 1,5x^2 - x + 2$

Ubica dos máximos en (-0,6;2,62) y (1,7;5,12), un mínimo en (0,25;1,87) y dos inflexiones en (-0,22;2,27) y (1,12;3,73) como indica la figura 117

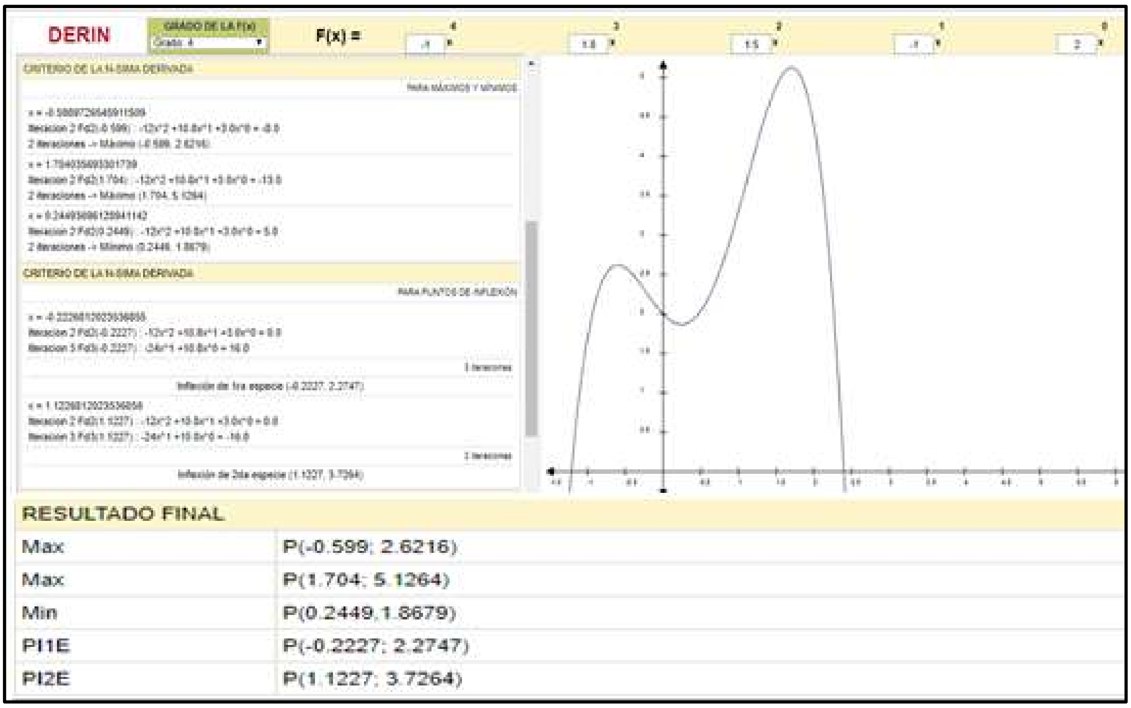


Figura N° 117 Función polinómica 16
Fuente: DERIN

Funcion 31 $f_{(x)} = -0,2x^4 + x^2 - x + 2$

Ubica dos máximos en (-1,79;4,94) y (1,21;1,82); un mínimo en (0,58;1,73) y dos inflexiones en (-0,91;3,6) y (0,91;1,78) como ilustra la figura 118

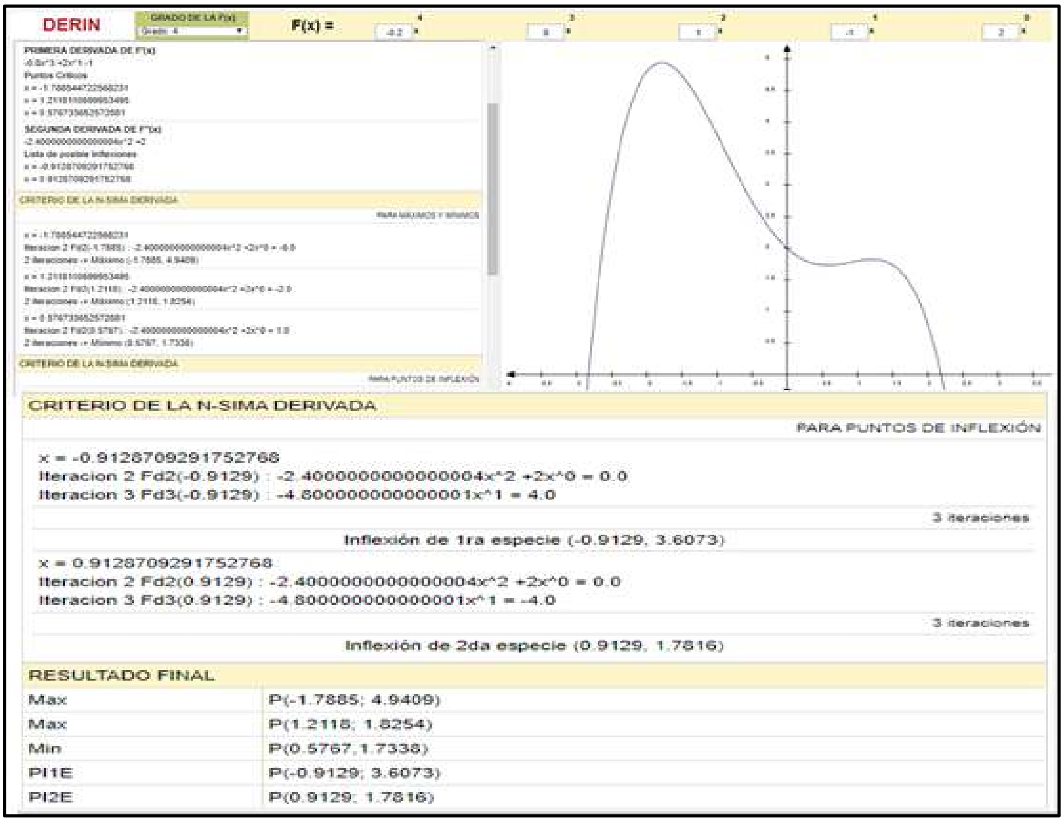


Figura N° 118 Función polinómica 17
Fuente: DERIN

4.7.4.2 Ejecución con el software WolframAlpha

Función 15 $f_{(x)} = x^4 + 1$

Ubica un mínimo en (0,1) como muestra la figura 119

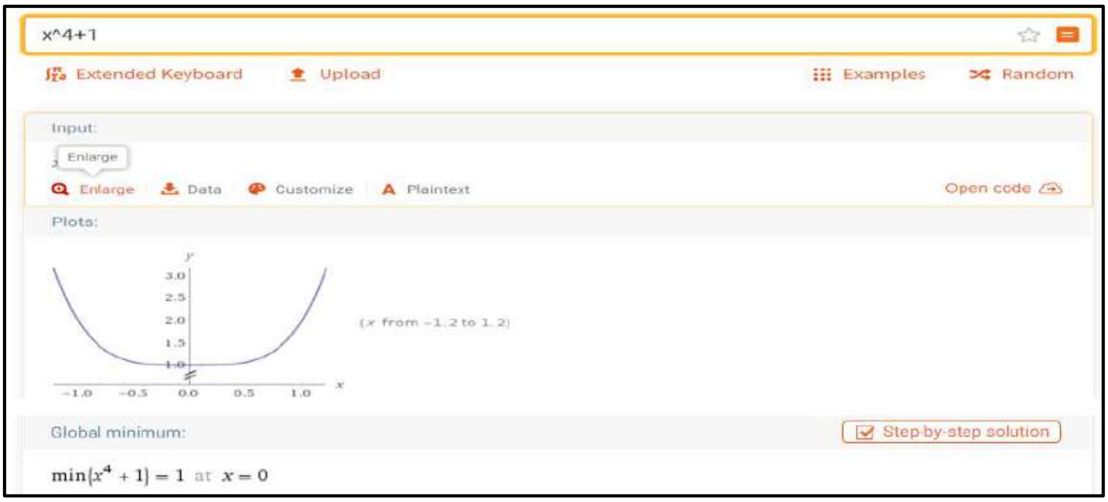


Figura N° 119 Función polinómica 1 - Wolfram
Fuente: WolframAlpha

Función 16 $f_{(x)} = x^4 + 2x^3$

Ubica un mínimo en $\left(-\frac{3}{2}; -\frac{27}{16}\right)$, tal como muestra la figura 120

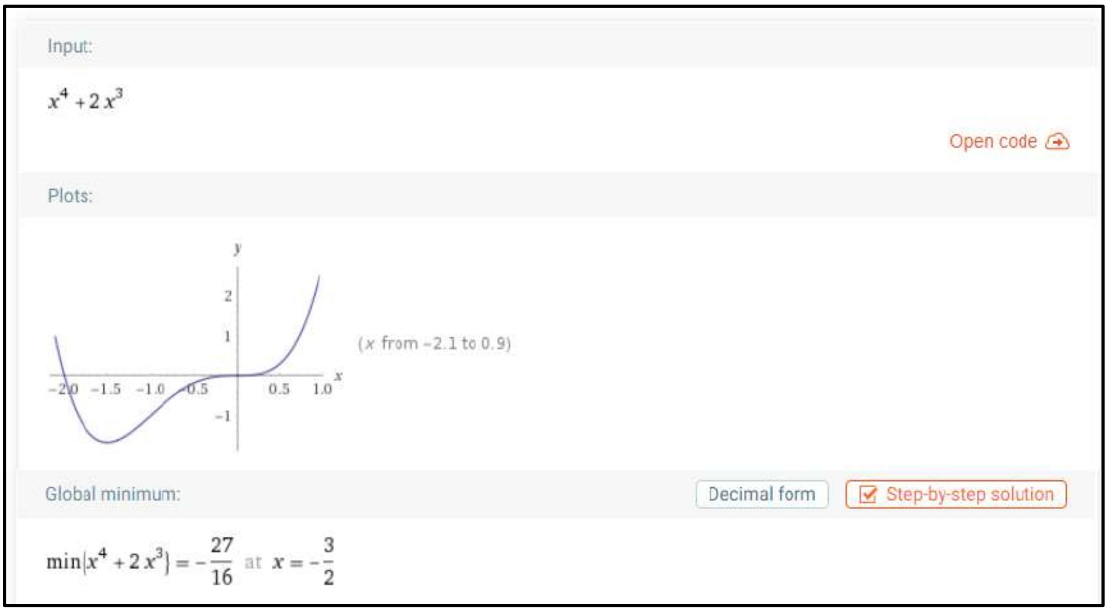


Figura N° 120 Función polinómica 2 Wolfram
Fuente: WolframAlpha

Función 17 $f_{(x)} = x^4 - 2x^3$

Ubica un mínimo en $\left(\frac{3}{2}; -\frac{27}{16}\right)$, como muestra la figura 121

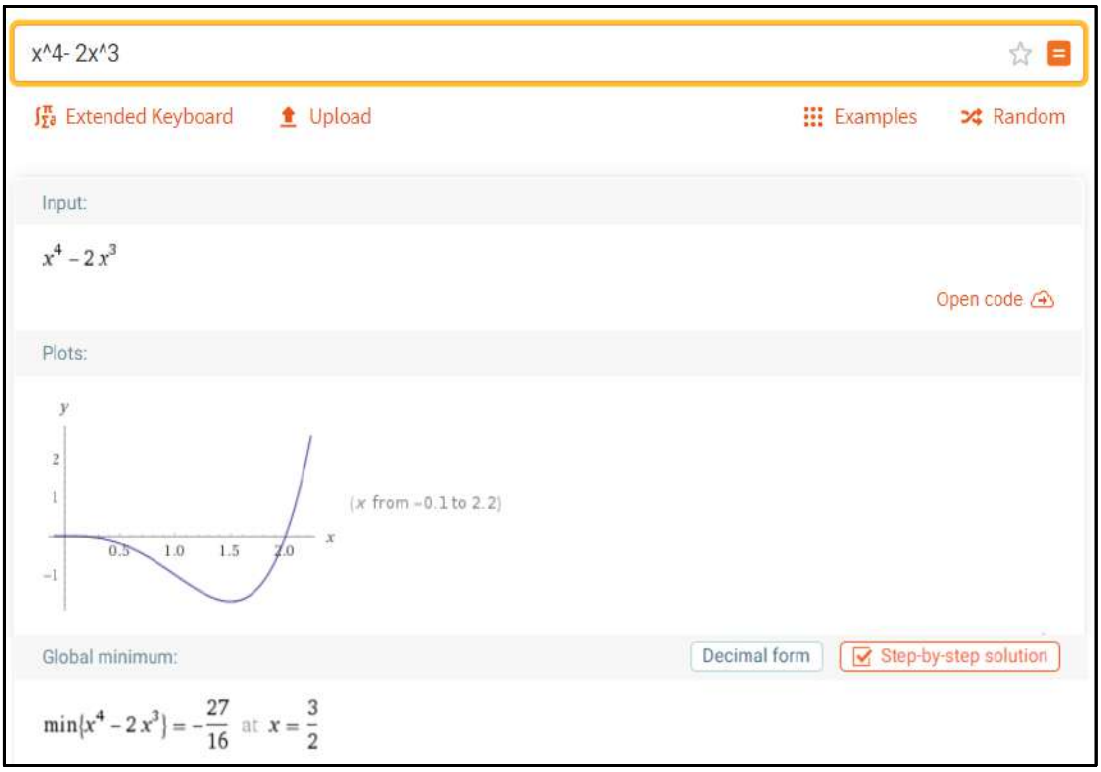


Figura N° 121 Función polinómica 3 Wolfram
Fuente: WolframAlpha

Función 18 $f_{(x)} = x^4 - 2x^2$

Ubica un mínimo en (-1;-1) y (1,-1) , un máximo en (0;0) como muestra la figura 122

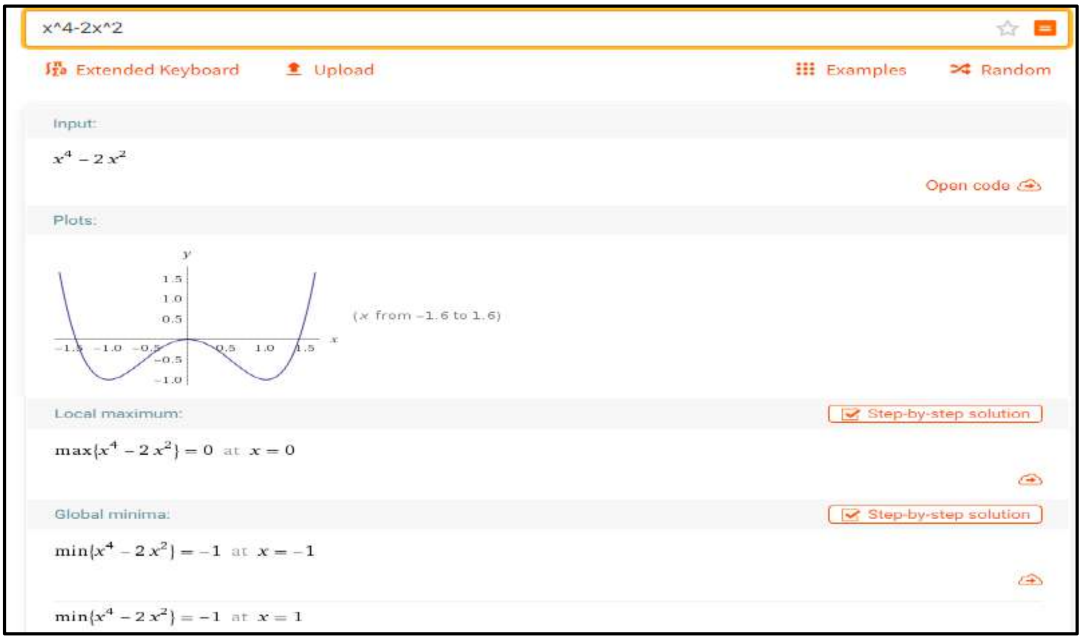


Figura N° 122 Función polinómica 4 Wolfram
Fuente: WolframAlpha

Funcion 19 $f_{(x)} = x^4 - 2x$

No alcanza el valor mínimo como muestra la figura 123

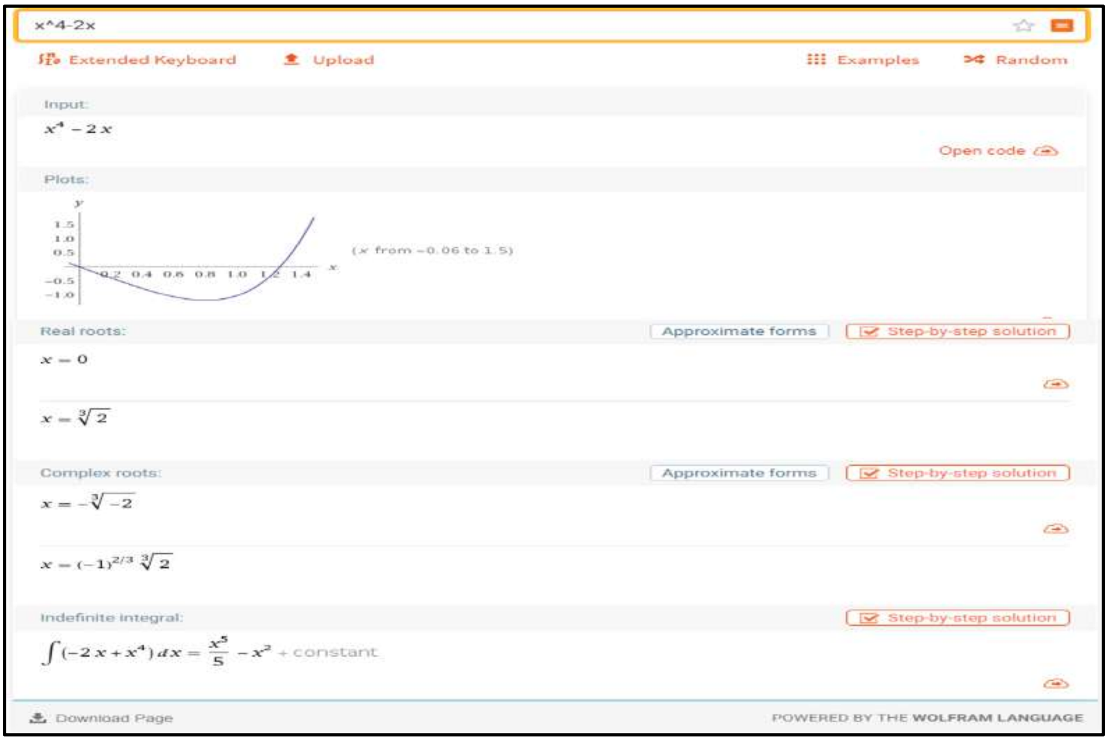


Figura N° 123 Función polinómica 5 Wolfram
Fuente: Wolfram

Funcion 20 $f_{(x)} = x^4 - 2x^3 - 3x^2$

Ubica un mínimo en (2,18; -12,39) y un máximo en (0;0). **No alcanza uno de los mínimos de la función** como muestra la figura 124

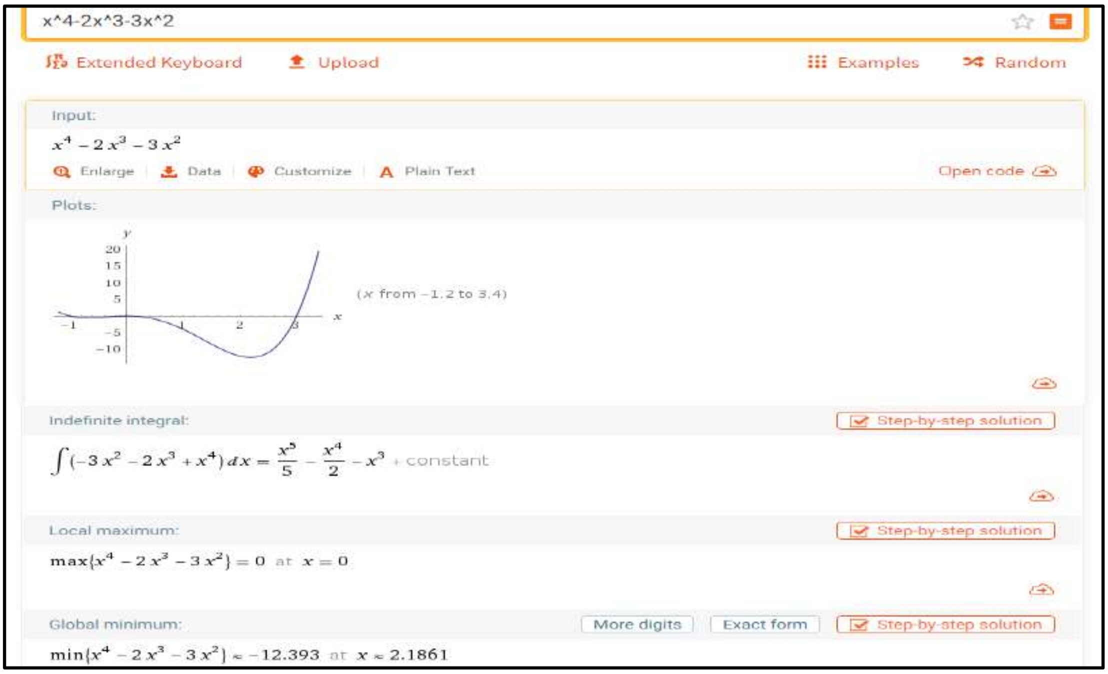


Figura N° 124 Función polinómica 6 - Wolfram
Fuente: WolframAlpha

Funcion 21 $f_{(x)} = x^4 - x^3 - 3x$

Ubica un mínimo en (1,24; -3,26) como muestra la figura 125

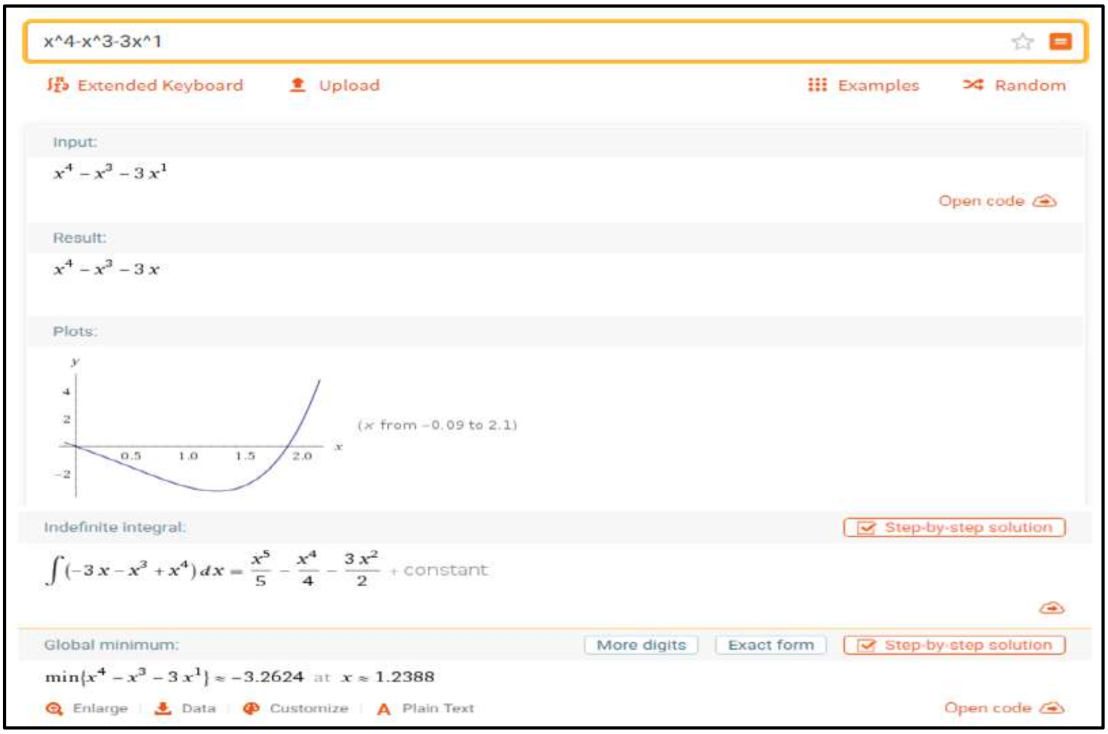


Figura N° 125 Función polinómica 7 Wolfram
Fuente: WolframAlpha

Funcion 22 $f_{(x)} = x^4 - x^3 - 2$

Ubica un mínimo en $\left(\frac{3}{4}; -\frac{539}{256}\right)$. tal como ilustra la figura 126

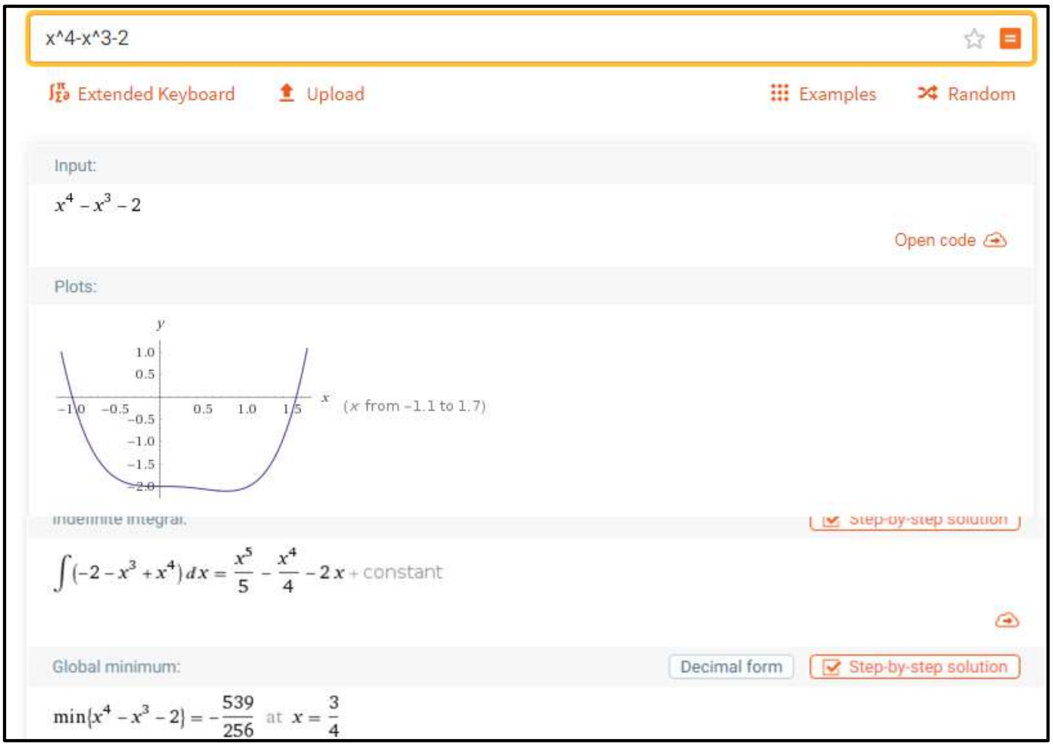


Figura N° 126 Función polinómica 8 Wolfram
Fuente: WolframAlpha

Funcion 23 $f_{(x)} = x^4 - 2x^2 - 3x$

Ubica un mínimo en (1,26;-4,43) tal como se ilustra en la figura 127

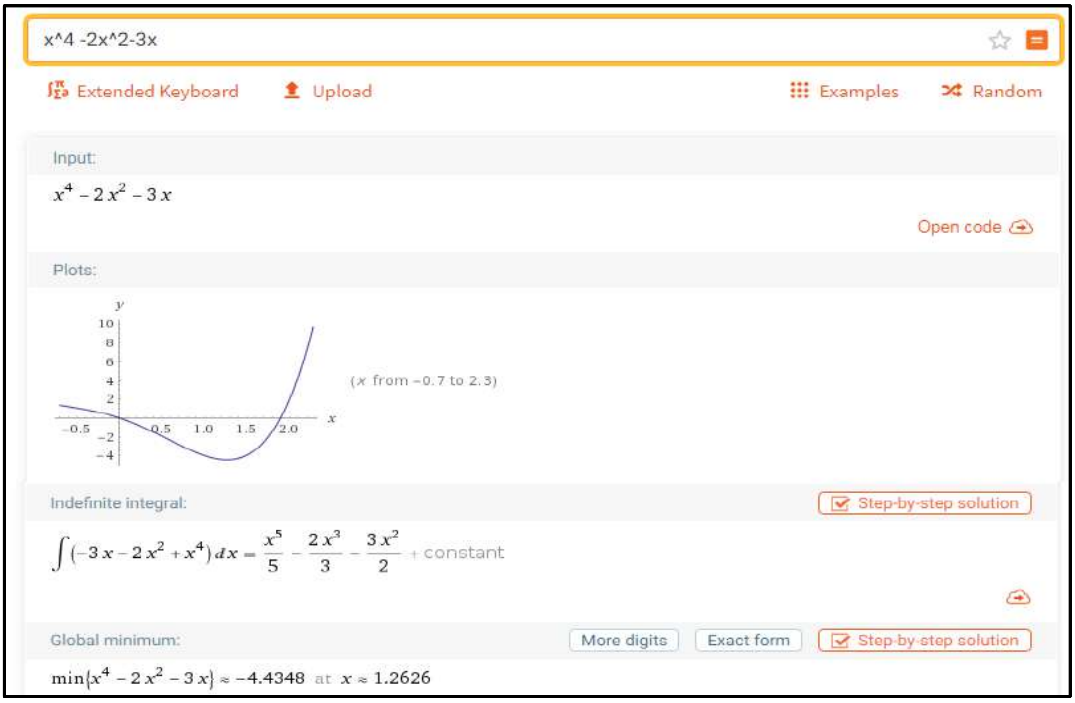


Figura N° 127 Función polinómica 9 - Wolfram
Fuente: WolframAlpha

Funcion 24 $f_{(x)} = x^4 - 3x^2 - 2$

Ubica dos mínimos en $x = \pm\sqrt{\frac{3}{2}} = \pm 1{,}22$ y un máximo en (0;0) como se ilustra en la figura 128

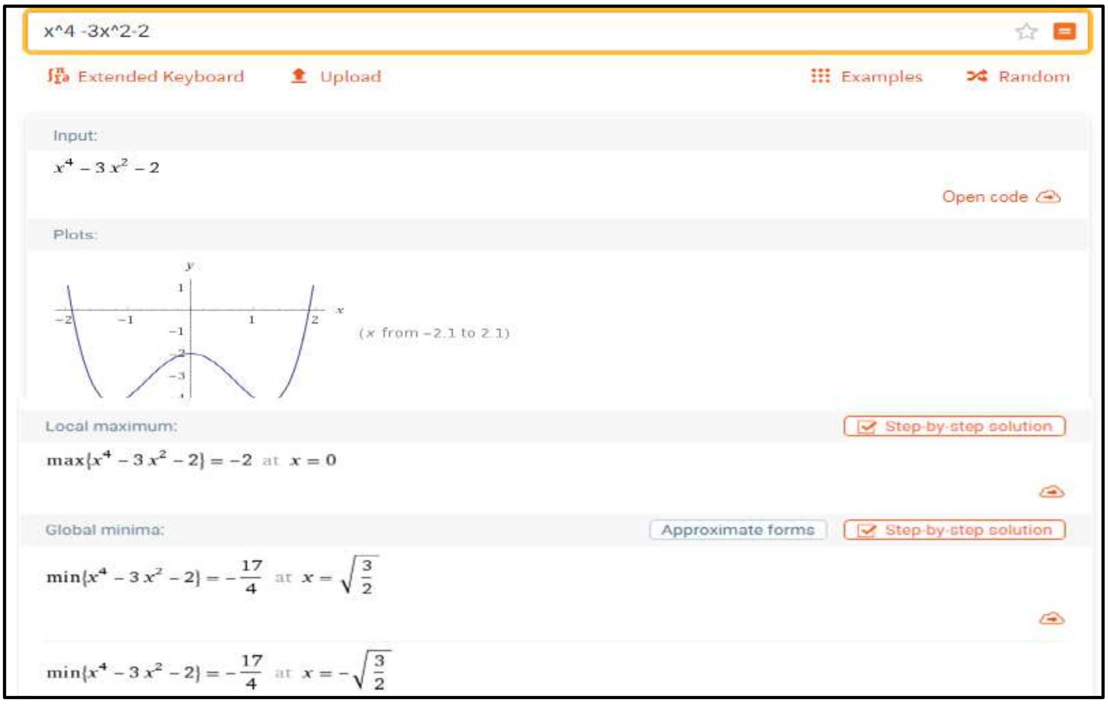


Figura N° 128 Función polinómica 10 Wolfram
Fuente: WolframAlpha

Funcion 25 $f_{(x)} = x^4 - 4x + 3$

Ubica un mínimo en (1;0) como muestra la figura 129

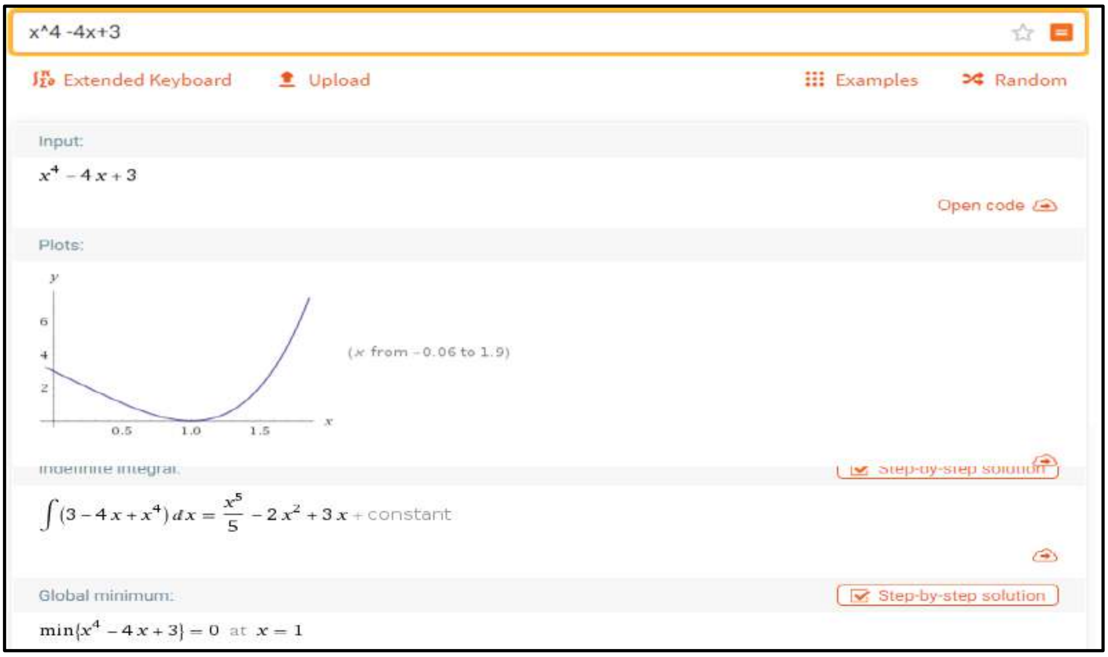


Figura N° 129 Función polinómica 11 Wolfram
Fuente: WolframAlpha

Funcion 26 $f_{(x)} = x^4 - x^3 - 2x^2 - 3x$

Ubica un mínimo en (1,64;-7,47) como muestra la figura 130

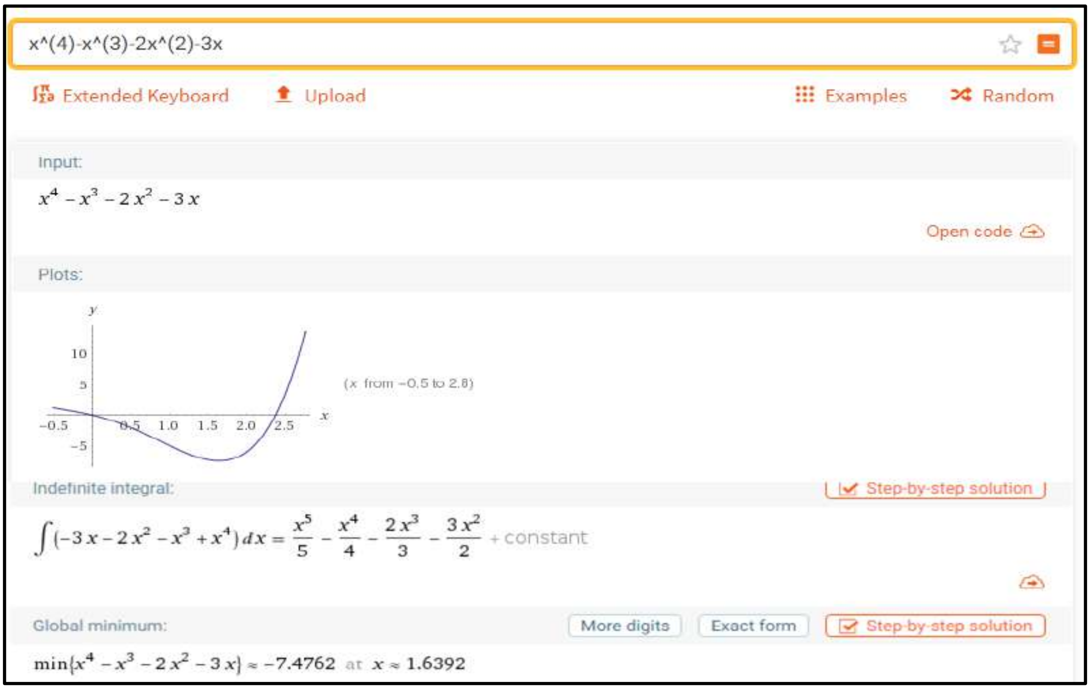


Figura N° 130 Función polinómica 12 - Wolfram
Fuente: WolframAlpha

Funcion 27 $f_{(x)} = x^4 - 3x^3 - 2x - 2$

Ubica un mínimo en el punto (2,34;-15,137) como muestra la figura 131

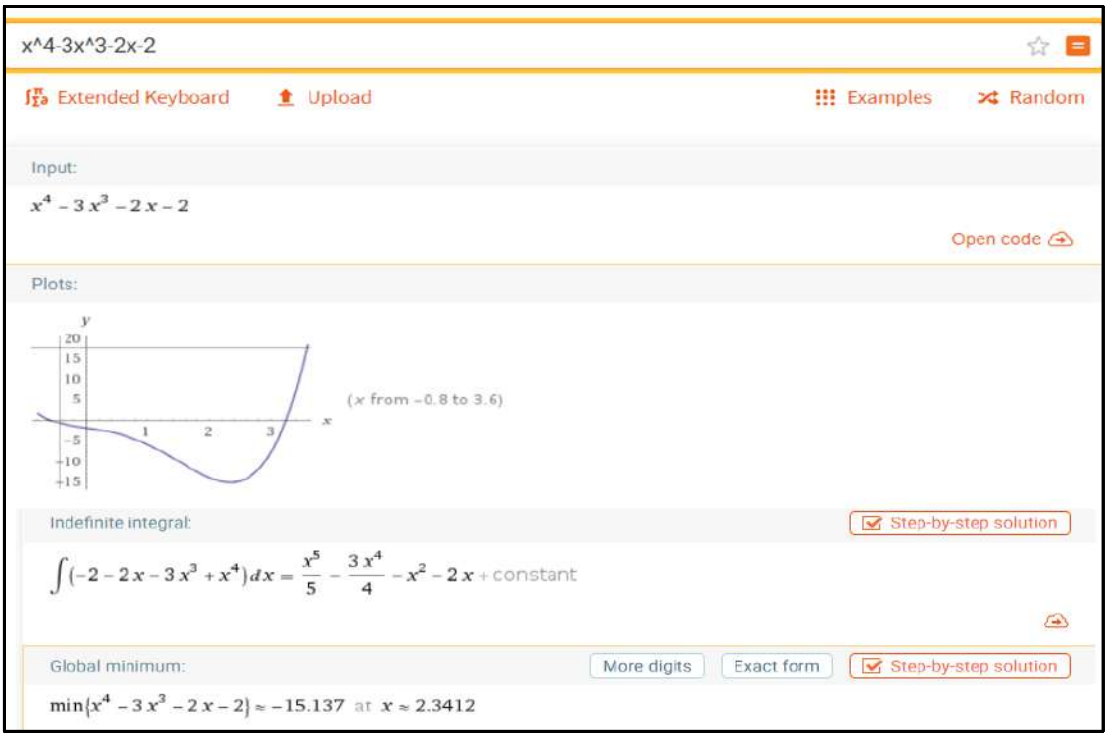


Figura N° 131 Función polinómica 13 Wolfram
Fuente: WolframAlpha

Funcion 28 $f_{(x)} = x^4 - 3x^2 - x - 2$

Ubica un mínimo en (1,3;-5,51) y un máximo en (-0,17;-1.91) como muestra la figura 132

Figura N° 132 Función polinómica 14 Wolfram
Fuente: WolframAlpha

Funcion 29 $f_{(x)} = x^4 - x^3 - 2x^2 - x - 2$

Ubica un mínimo en (1,52,-6,31) como muestra la figura 133

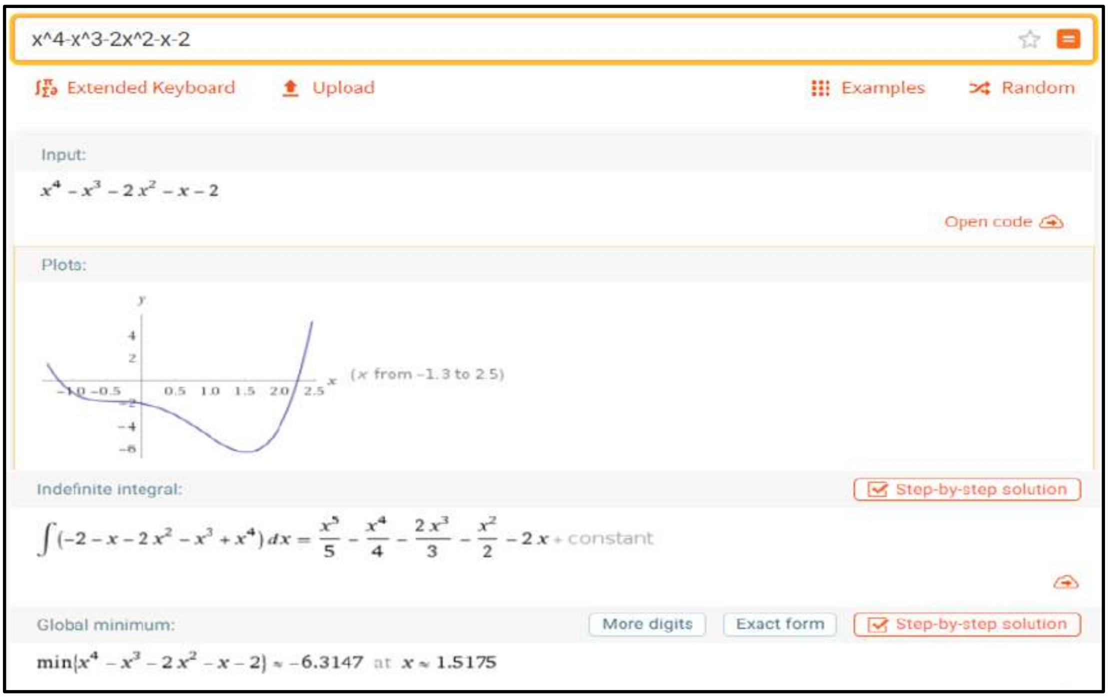


Figura N° 133 Función polinómica 15 Wolfram
Fuente: WolframAlpha

Funcion 30 $f_{(x)} = -x^4 + 1,8x^3 + 1,5x^2 - x + 2$

Ubica un máximo en (1,7;5,12) y un mínimo en (0,24;1,87) pero **no presenta** el otro máximo tal como muestra la figura 134

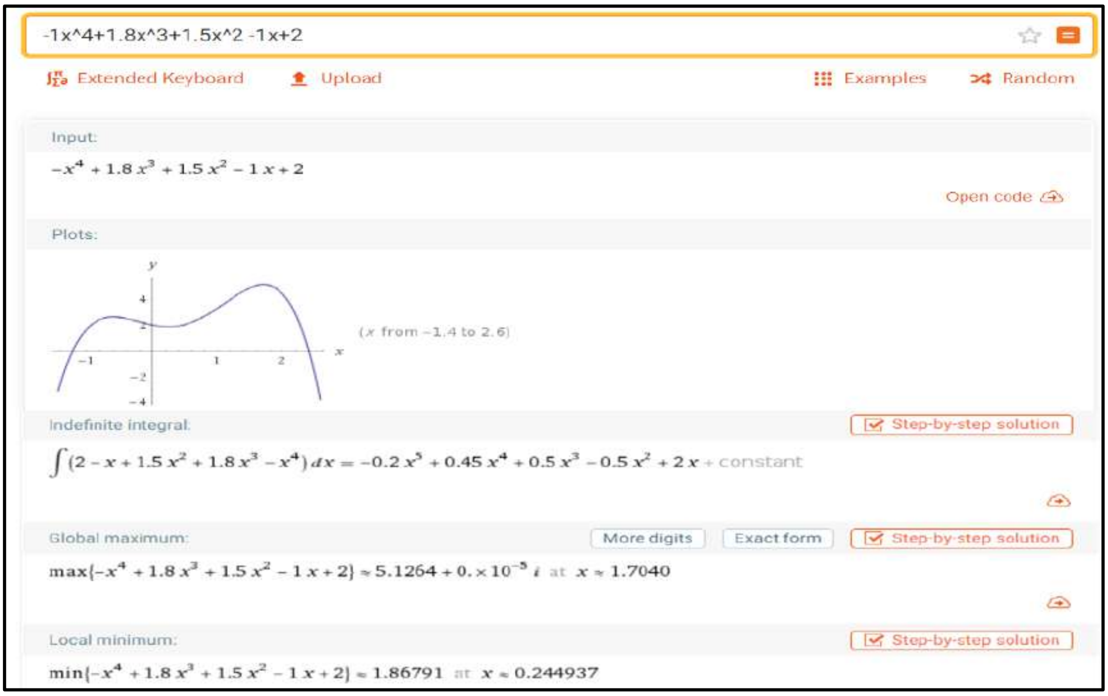


Figura N° 134 Función polinómica 16 Wolfram
Fuente: WolframAlpha

Funcion 31 $f_{(x)} = -0,2x^4 + x^2 - x + 2$

Ubica un máximo en (-1,78;4,94) y un mínimo en (0,58;1,73). **No ubica** el otro punto que corresponde a un máximo local como muestra la figura 135

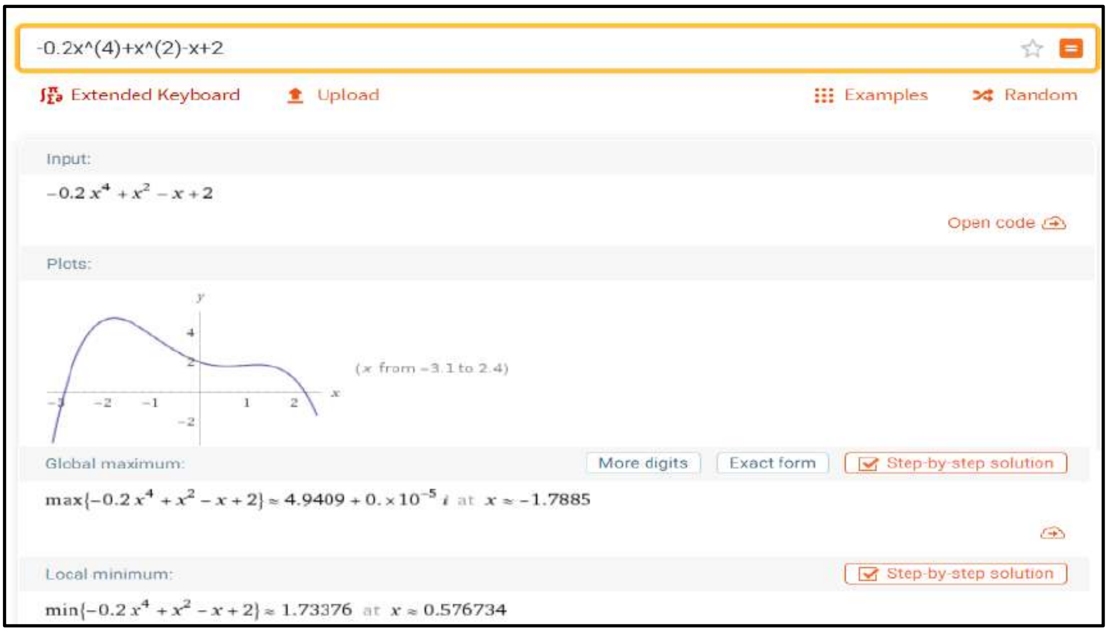


Figura N° 135 Función polinómica 17 Wolfram
Fuente: WolframAlpha

4.7.4.3 Ejecución con el software GeoGebra

Función 15 $f_{(x)} = x^4 + 1$

Ubica un mínimo en (0;1) como indica la figura 136

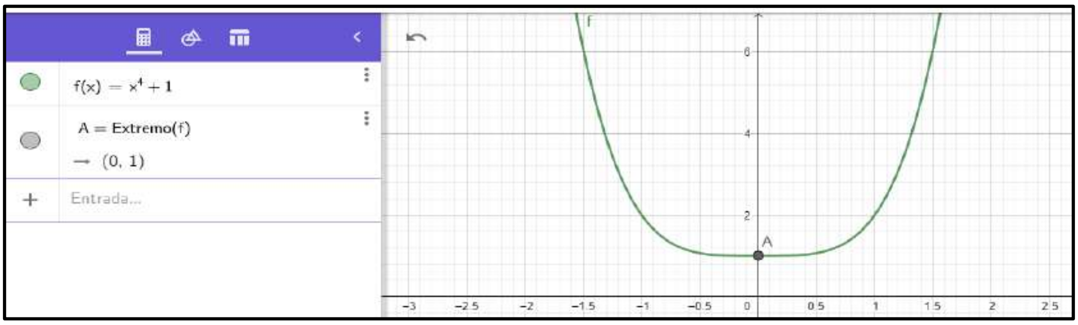


Figura N° 136 Función polinómica 1 - GeoGebra
Fuente: GeoGebra

Función 16 $f_{(x)} = x^4 + 2x^3$

Ubica un mínimo en (-1,5; -1,6875) y puntos de inflexión en (-1,-1) y en (0;0) como muestra la figura 137

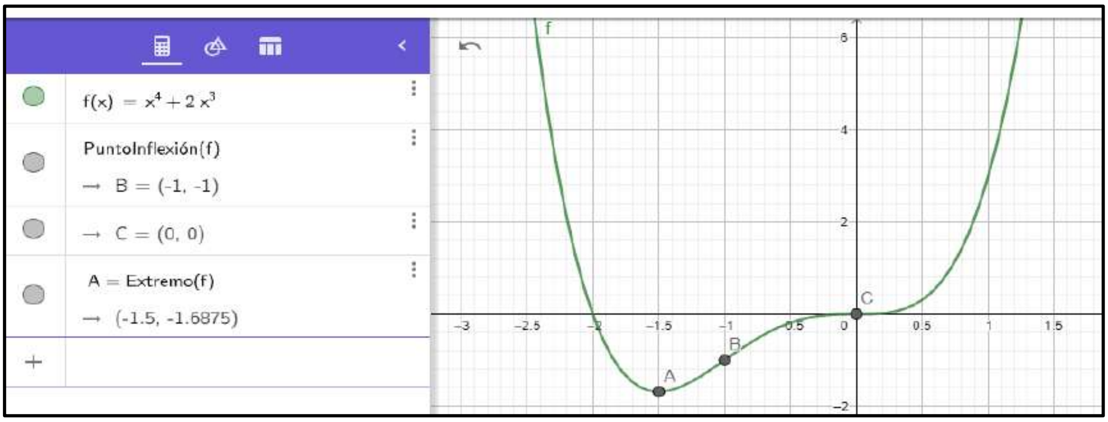


Figura N° 137 Función polinómica 2 GeoGebra
Fuente: GeoGebra

Función 17 $f_{(x)} = x^4 - 2x^3$

Ubica un mínimo en (1,5;-1,81) y dos inflexiones en (0;0) y (1;-1) como ilustra la figura 138

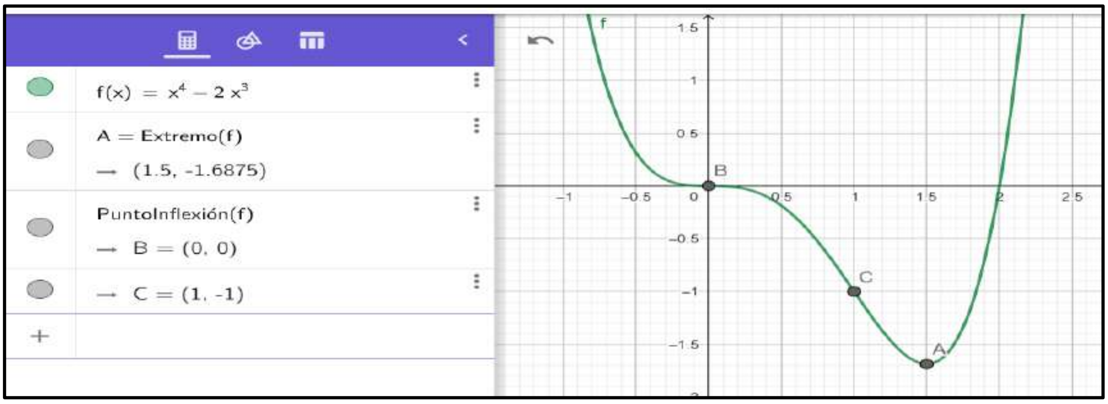


Figura N° 138 Función polinómica 3 GeoGebra
Fuente: GeoGebra

Función 18 $f_{(x)} = x^4 - 2x^2$

Ubica un máximo en (0;0) , dos mínimos en (-1;-1) , (1;-1) e inflexiones en (0,58;-0,56) y (0,58;-0,56) como indica la figura 139

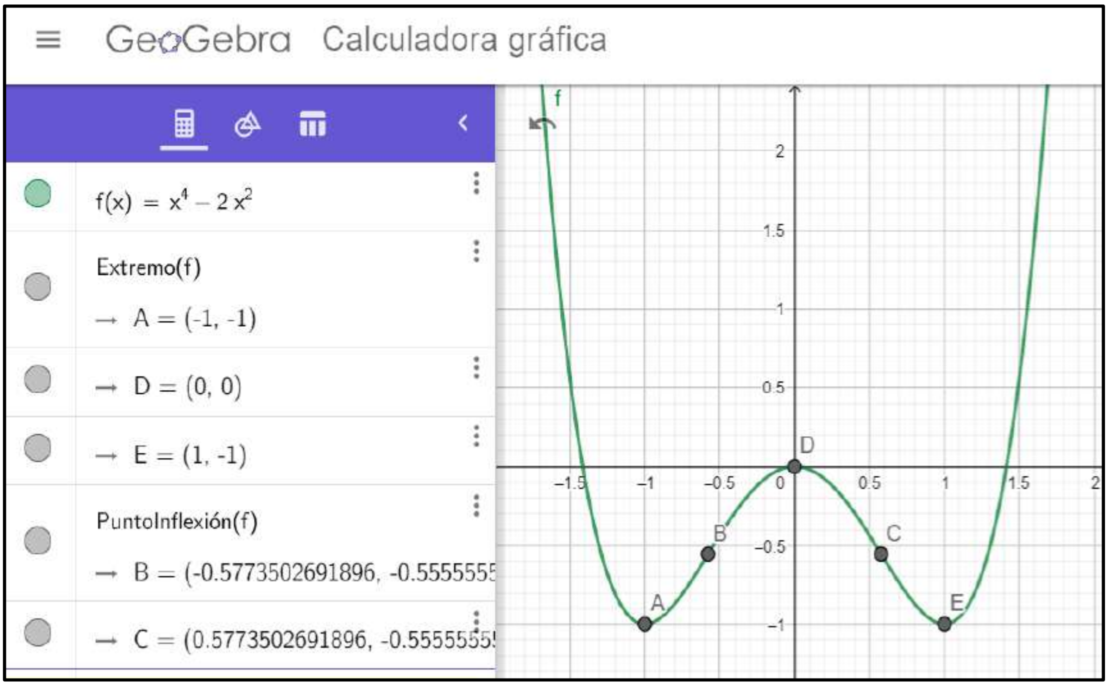


Figura N° 139 Función polinómica 4 GeoGebra
Fuente: GeoGebra

Funcion 19 $f_{(x)} = x^4 - 2x$

Ubica un mínimo en (0,79; -1,19) como muestra la figura 140

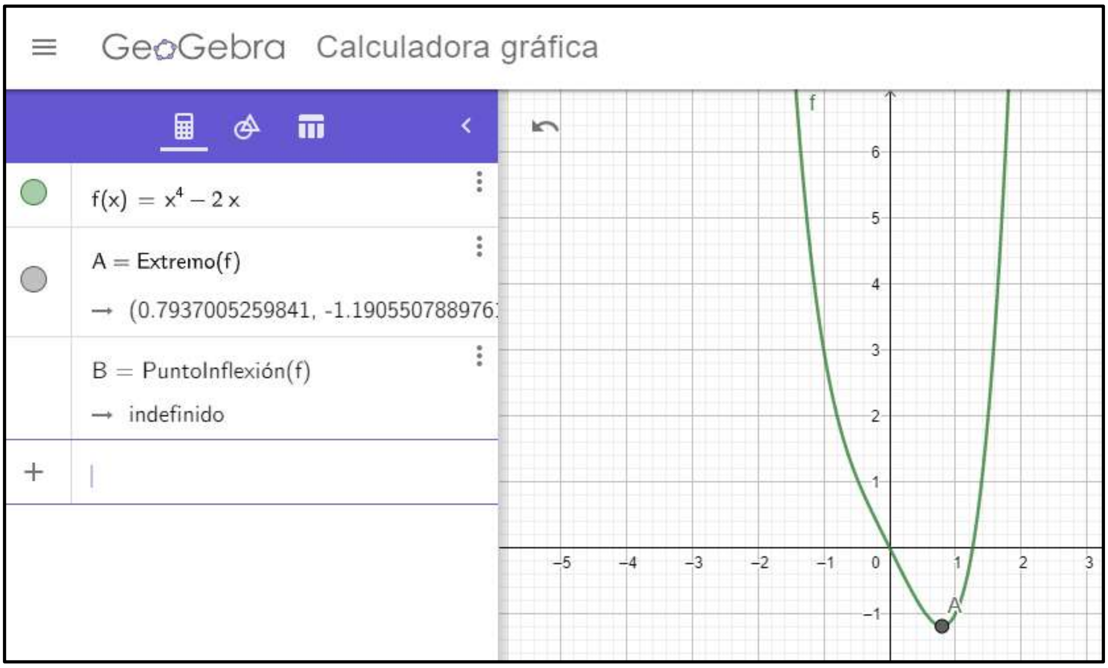


Figura N° 140 Función polinómica 5 GeoGebra
Fuente: GeoGebra

Funcion 20 $f_{(x)} = x^4 - 2x^3 - 3x^2$

Ubica mínimos en (-0,68;-0,54) y (2,18;-12,39) , un máximo en (0;0) y dos inflexiones en (1,37;-7,21) y (-0,37;-0,28) como muestra la figura 141

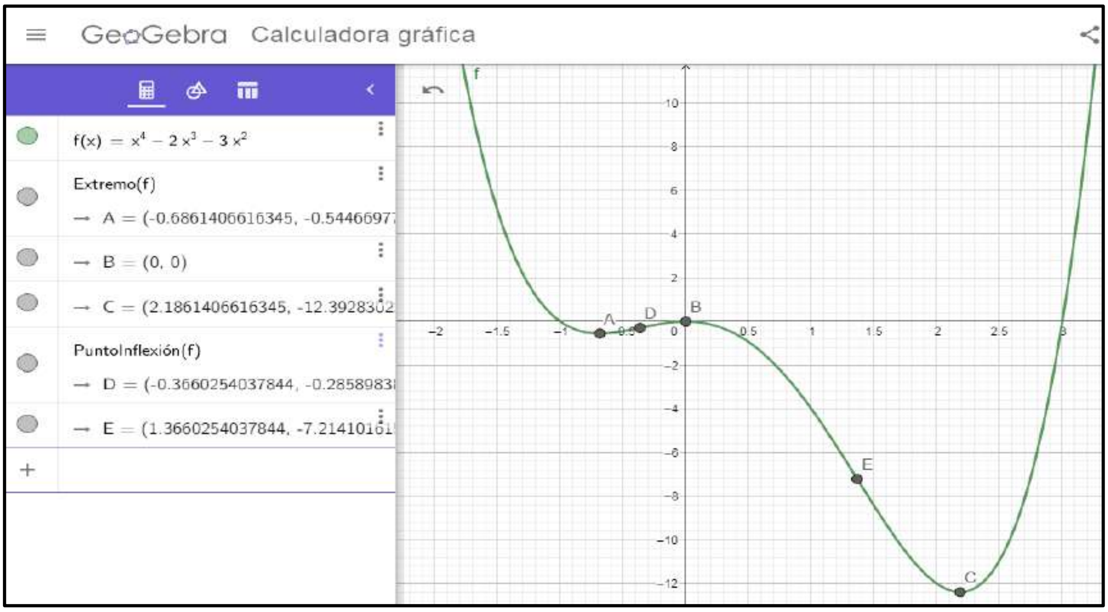


Figura N° 141 Función polinómica 6 GeoGebra
Fuente: GeoGebra

Funcion 21 $f_{(x)} = x^4 - x^3 - 3x$

Ubica un mínimo en (1,24; -3,26) e inflexines en (0;0) y (0,5;-1,56) como ilustra la figura 142

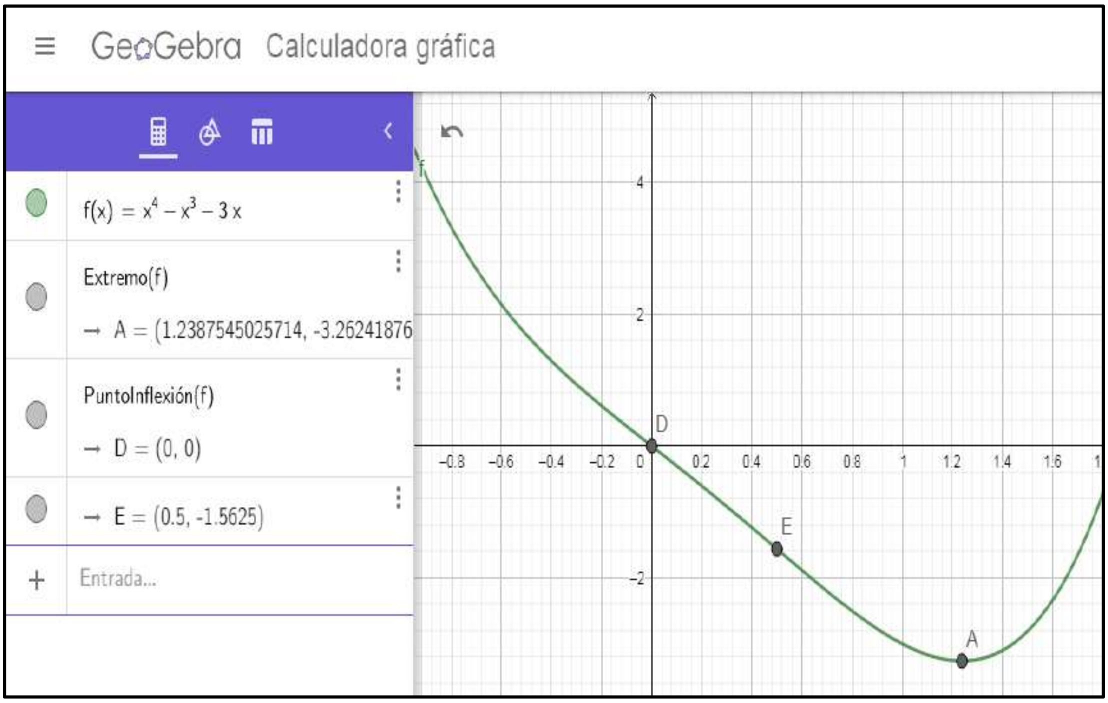


Figura N° 142 Función polinómica 7 GeoGebra
Fuente: GeoGebra

Funcion 22 $f_{(x)} = x^4 - x^3 - 2$

Ubica un mínimo en (0,75; -2,11) y dos puntos de inflexión en (0;-3) y (0,5;-2,0625) como se ilustra en la figura 143

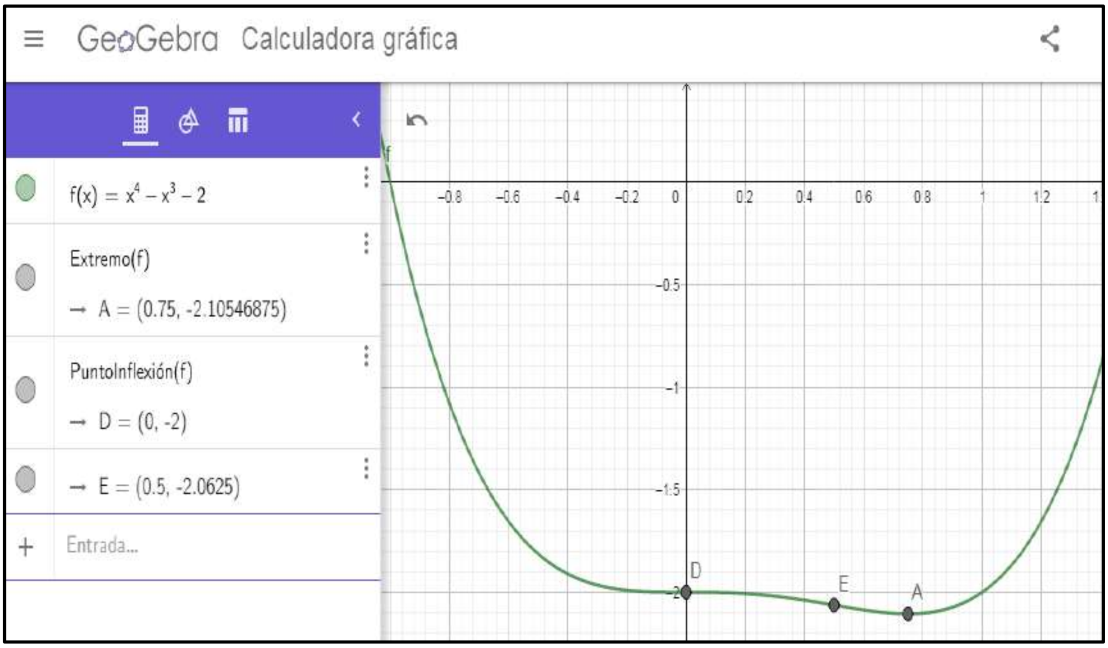


Figura N° 143 Función polinómica 8 - GeoGebra
Fuente: GeoGebra

Funcion 23 $f_{(x)} = x^4 - 2x^2 - 3x$

Ubica un mínimo en (1,26; -4,43) e inflexiones en (-0,57;1,17) y (0,57; -2,28) como muestra la figura 144

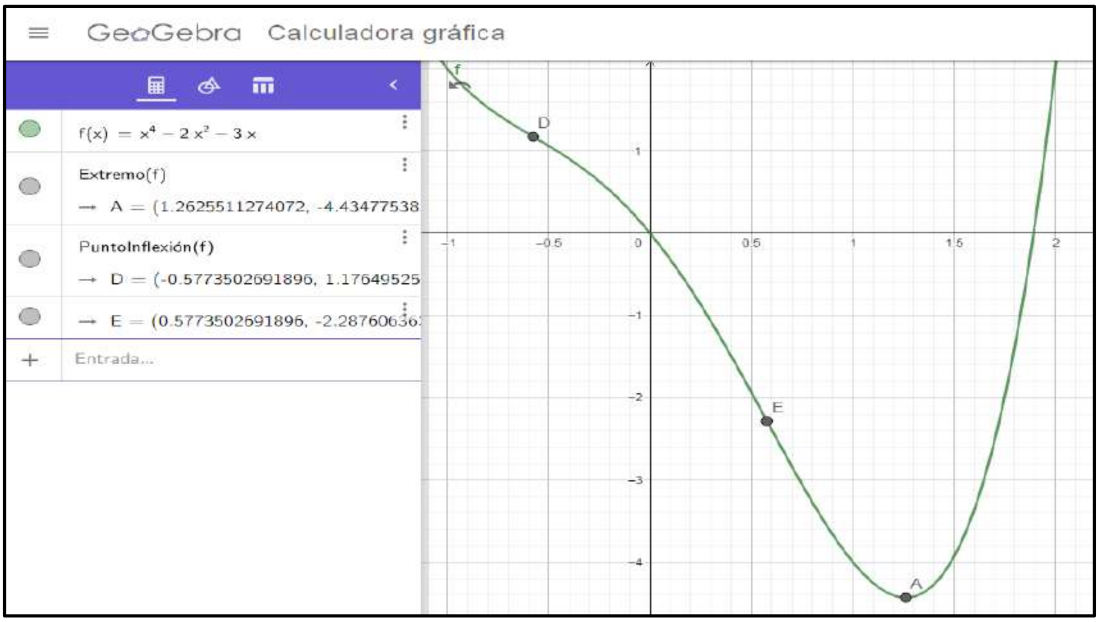


Figura N° 144 Función polinómica 9 - GeoGebra
Fuente: GeoGebra

Funcion 24 $f_{(x)} = x^4 - 3x^2 - 2$

Ubica dos mínimos en (-1,22;-4,25) , (1,22;-4,25) , un máximo en (0;-2) e inflexiones en (-0,71;-3,25) y (0,71;-3,25) como muestra la figura 145

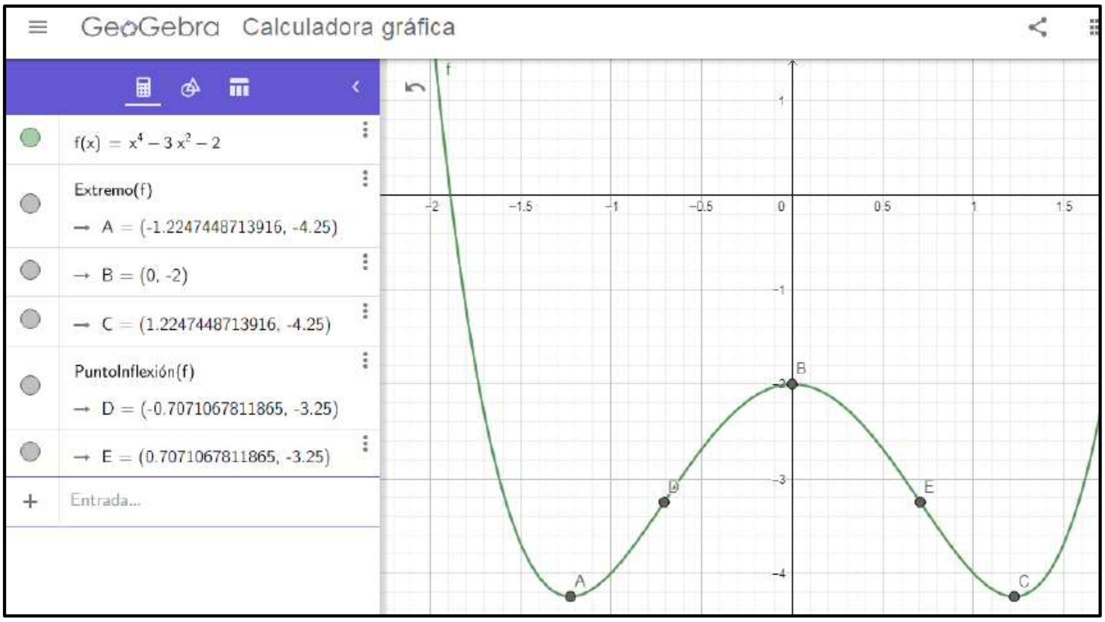


Figura N° 145 Función polinómica 10 GeoGebra
Fuente: GeoGebra

Funcion 25 $f_{(x)} = x^4 - 4x + 3$

Ubiac un mínimo en (1,0) tal como muestra la figura 146

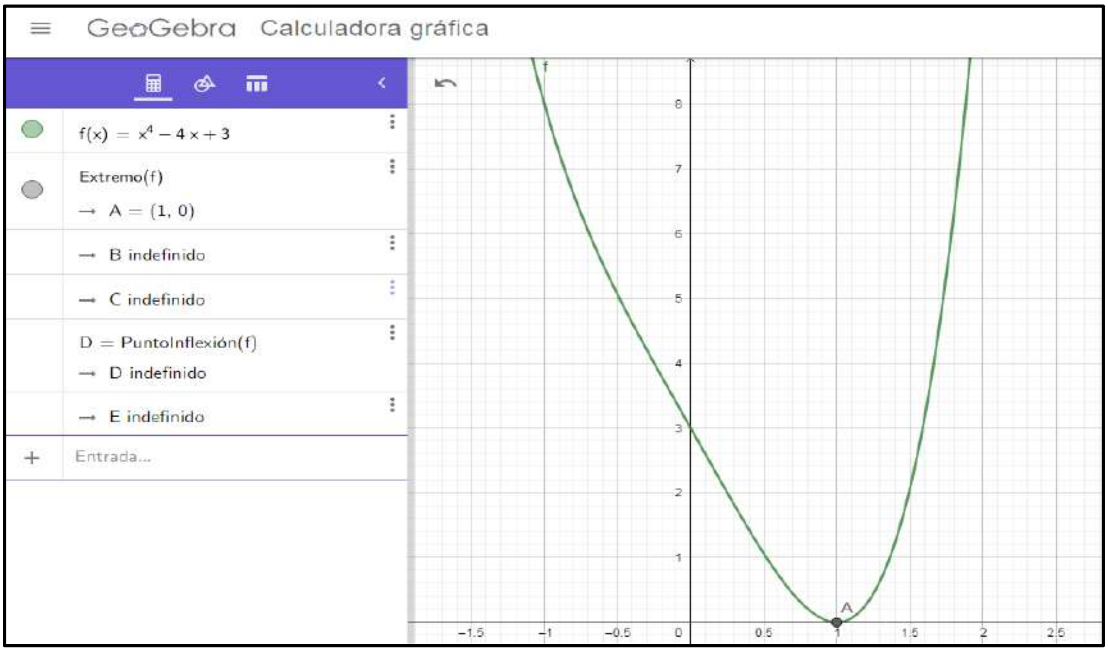


Figura N° 146 Función polinómica 11 GeoGebra
Fuente: GeoGebra

Funcion 26 $f_{(x)} = x^4 - x^3 - 2x^2 - 3x$

Ubica un mínimo en (1,64; -7,47) y dos inflexiones en (-0,38;0,93) y (0,88;-4,26) como muestra la figura 147

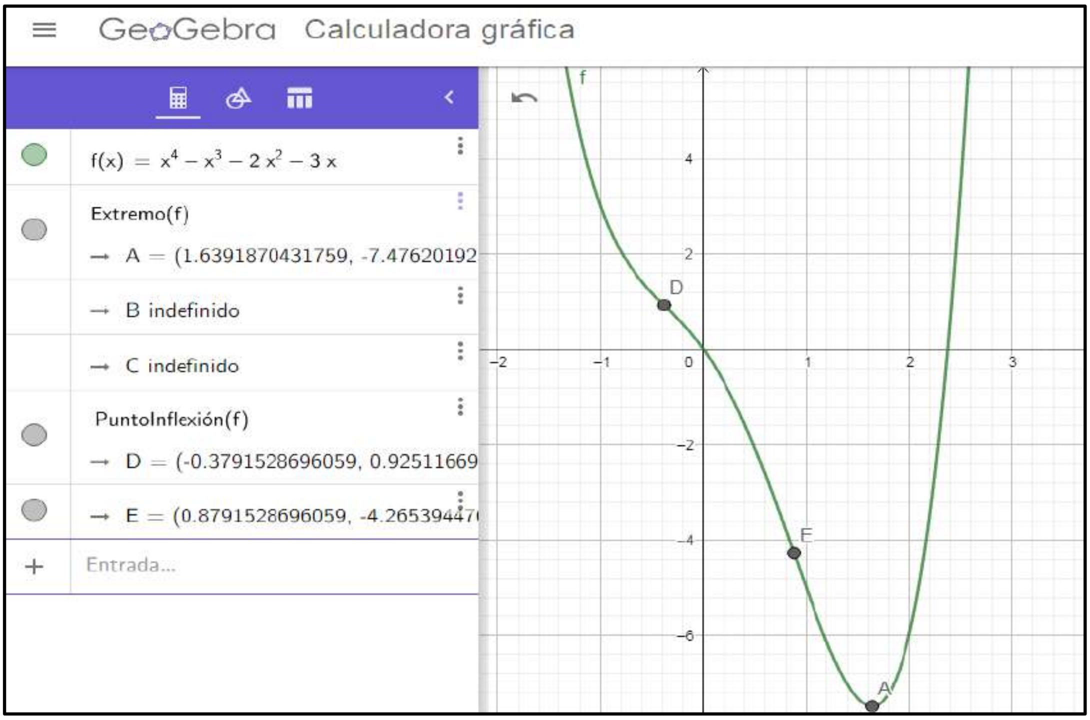


Figura N° 147 Función polinómica 12 GeoGebra
Fuente: GeoGebra

Funcion 27 $f_{(x)} = x^4 - 3x^3 - 2x - 2$

Ubica un mínimo en (2,34; -15,136) y dos puntos de inflexión en (0,-2) y (1,5;-10,0625) como ilustra la figura 148

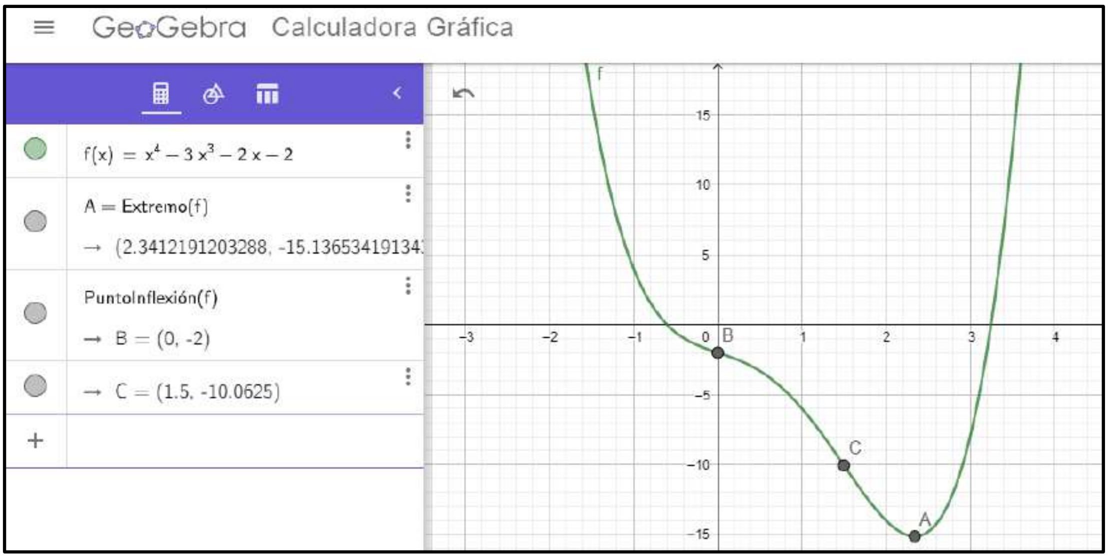


Figura N° 148 Función polinómica 13 - GeoGebra
Fuente: GeoGebra

Funcion 28 $f_{(x)} = x^4 - 3x^2 - x - 2$

Ubica tres extremos en (-1,13;-3,07) , (-0,17;-1,91) y (1,3; -5,51) así como inflexiones en (-0,71;-2,54) y (0,71;-3,95) como indica la figura 149

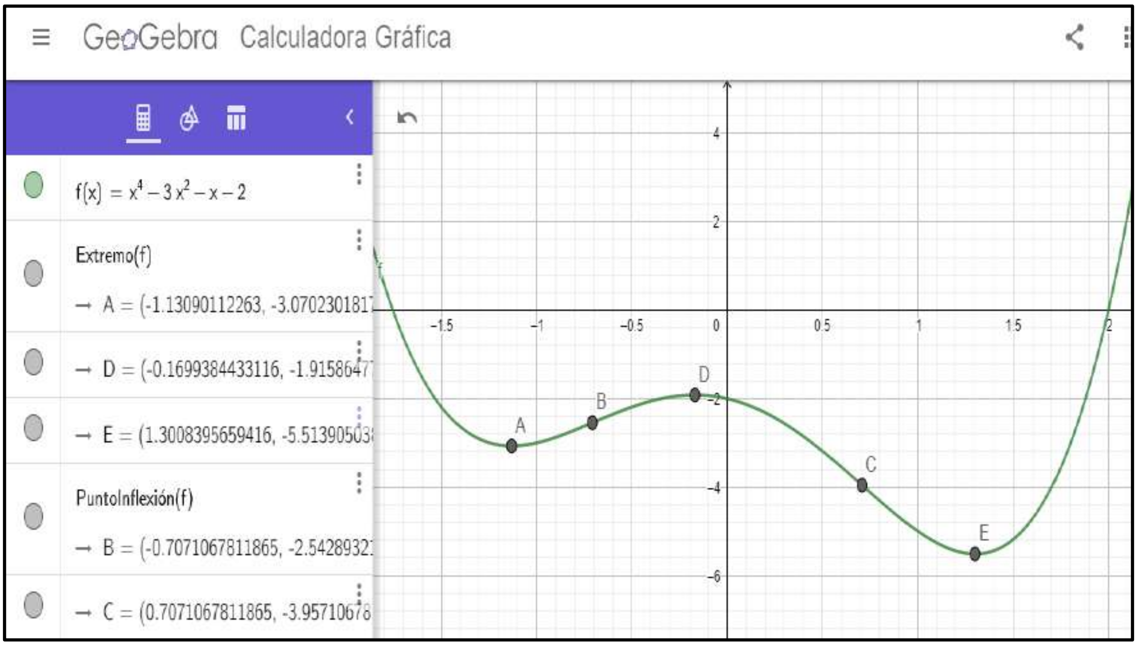


Figura N° 149 Función polinómica 14 GeoGebra
Fuente: GeoGebra

Funcion 29 $f_{(x)} = x^4 - x^3 - 2x^2 - x - 2$

Ubica un extremo en (1,52;-6,31) y dos inflexiones en (-0,38;-1,83) y (0,88;-4,51) como ilustra la figura 150

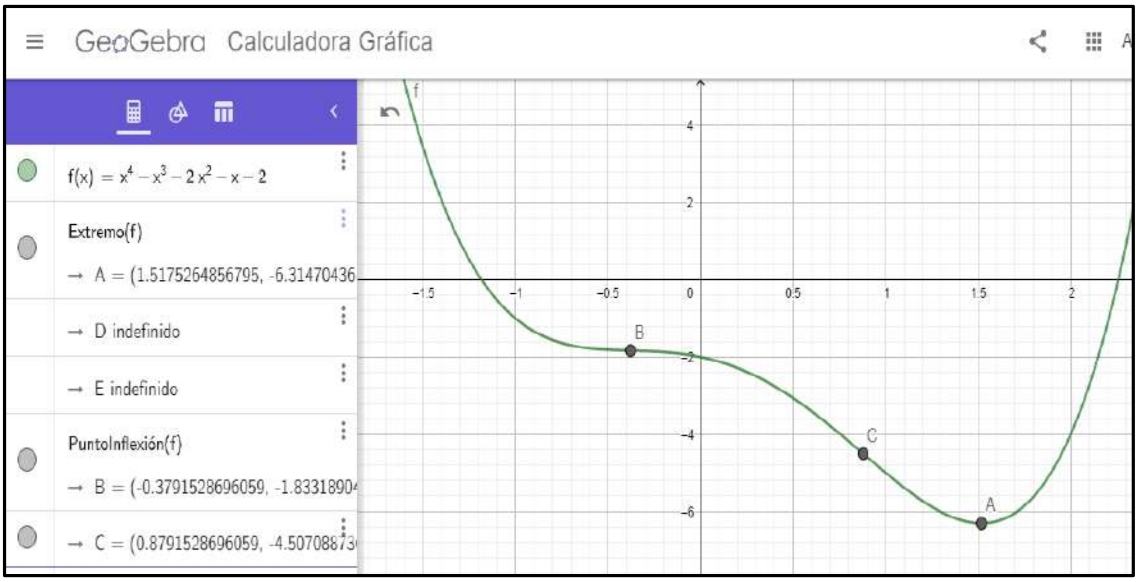


Figura N° 150 Función polinómica 15 GeoGebra
Fuente: GeoGebra

Funcion 30 $f_{(x)} = -x^4 + 1,8x^3 + 1,5x^2 - x + 2$

Ubica tres extremos en (-0,6;2,62) , (0,24; 1,87) y (1,7;5,12) así como inflexiones en (-0,22;2,27) y (1,12;3,73) como ilustra la figura 151

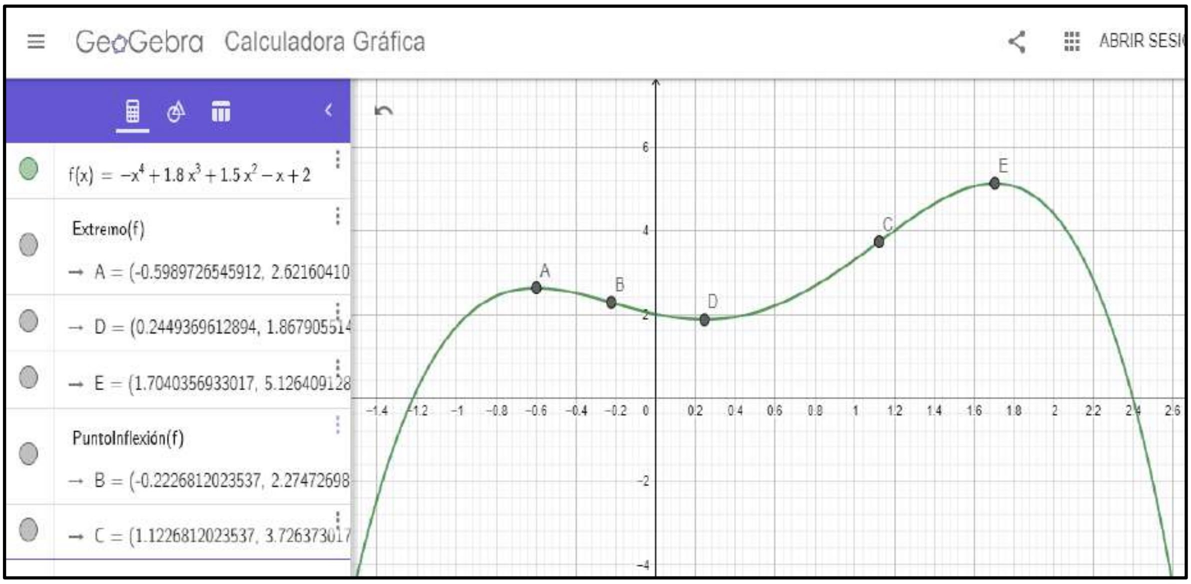


Figura N° 151 Función polinómica 16 GeoGebra
Fuente: GeoGebra

Funcion 31 $f_{(x)} = -0,2x^4 + x^2 - x + 2$

Ubica tres extremos en (-1,78;4,94) , (0,58;1,73) y (1,21;1,82) así como dos inflexiones en (-0,91;3,06) y (0,91;1,78) como ilustra la figura 152

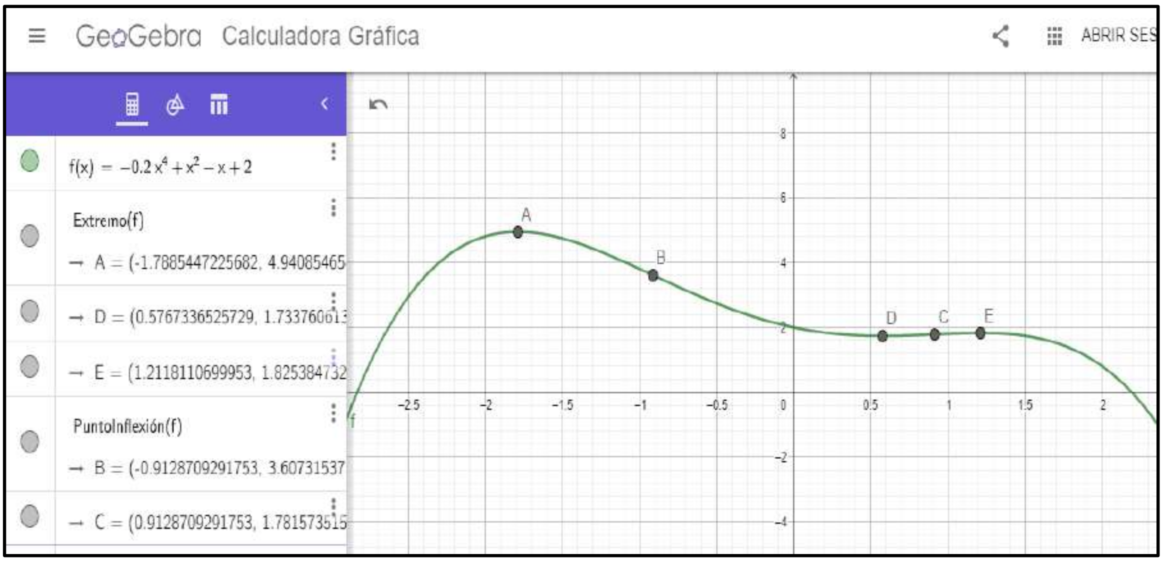


Figura N° 152 Función polinómica 17 GeoGebra
Fuente: GeoGebra

4.7.4.4 Ejecución con el software Symbolab

Función 15 $f_{(x)} = x^4 + 1$

Ubica el mínimo en (0;1) como indica la figura 153

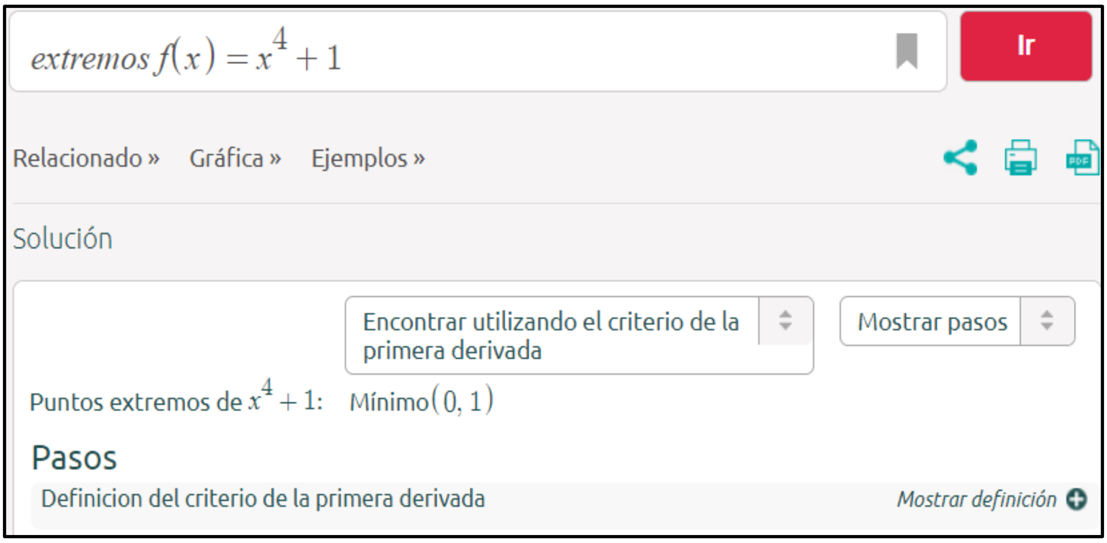


Figura N° 153 Función polinómica 1 - Symbolab
Fuente: Symbolab

Función 16 $f_{(x)} = x^4 + 2x^3$

Ubica un mínimo en (-1,5; -1,6875) y solo registra un punto de silla en (0;0) utilizando la primera derivada como indica la figura 154

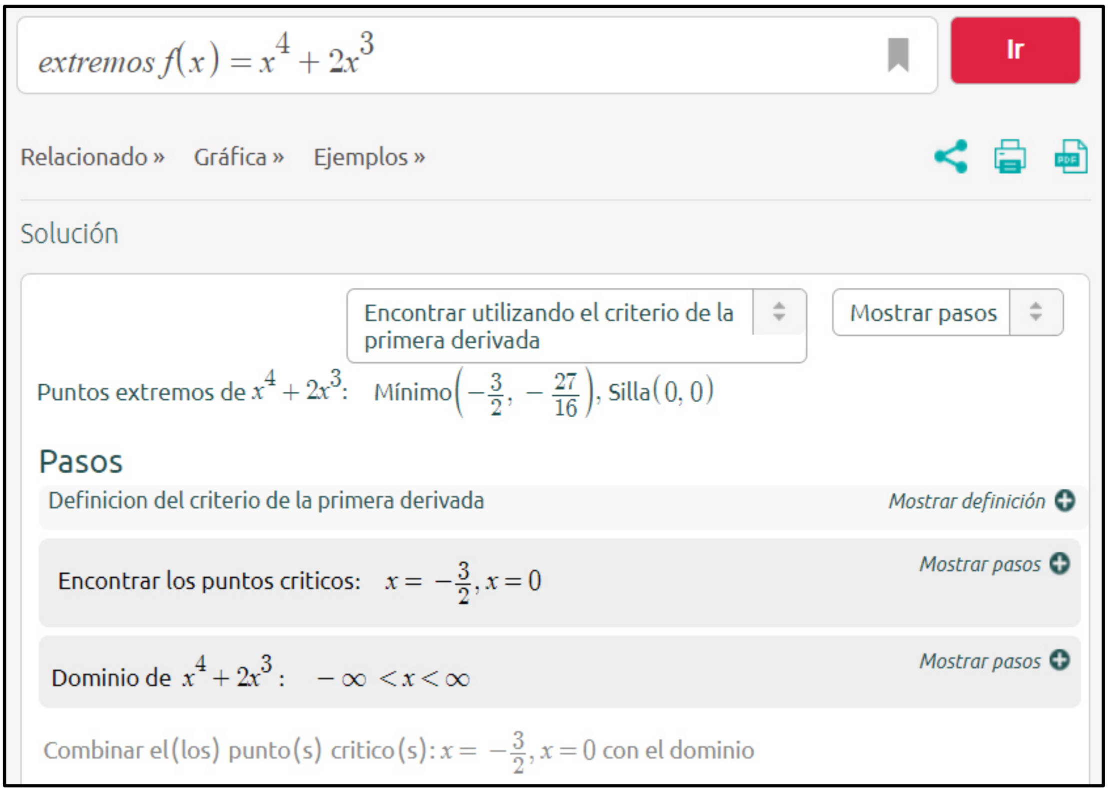


Figura N° 154 Función polinómica 2 Symbolab_1
Fuente: Symbolab

Al utilizar el criterio de la segunda derivada el punto (0,0) no es máximo ni mínimo, el criterio **es poco concluyente** como indica la figura 155

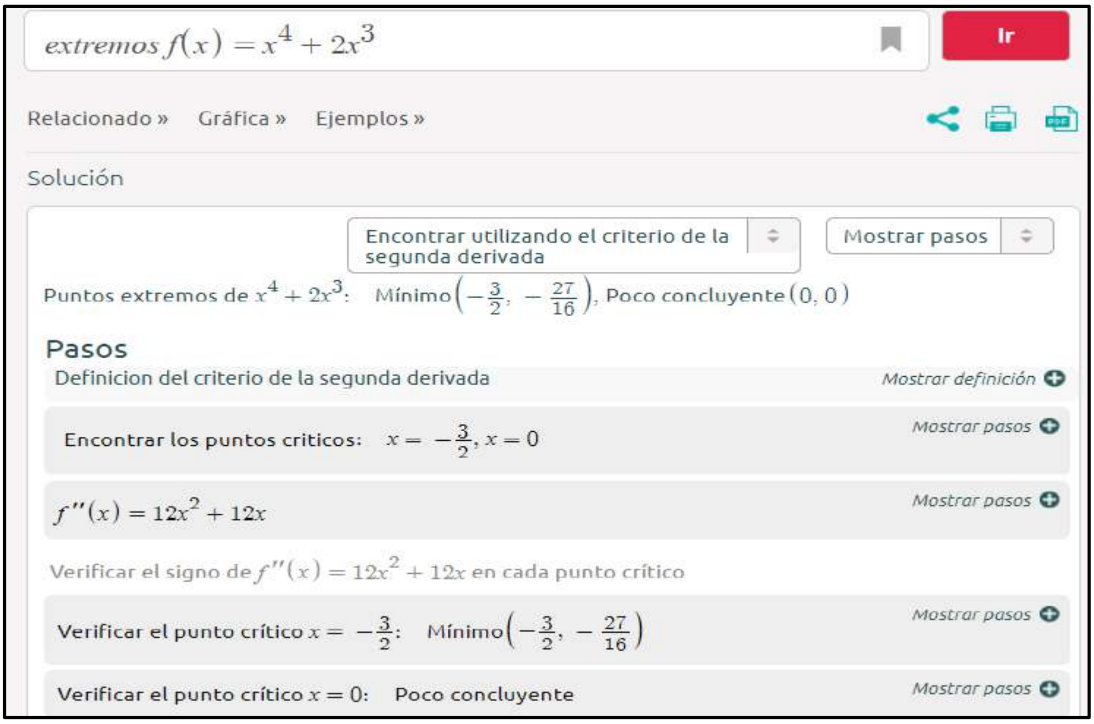


Figura N° 155 Función polinómica 2 Symbolab_2
Fuente: Symbolab

Función 17 $f_{(x)} = x^4 - 2x^3$

Ubica el punto mínimo en (1,5;-1,81) y un punto de silla en (0;0) y no reconoce punto de inflexión tal como indica la figura 156

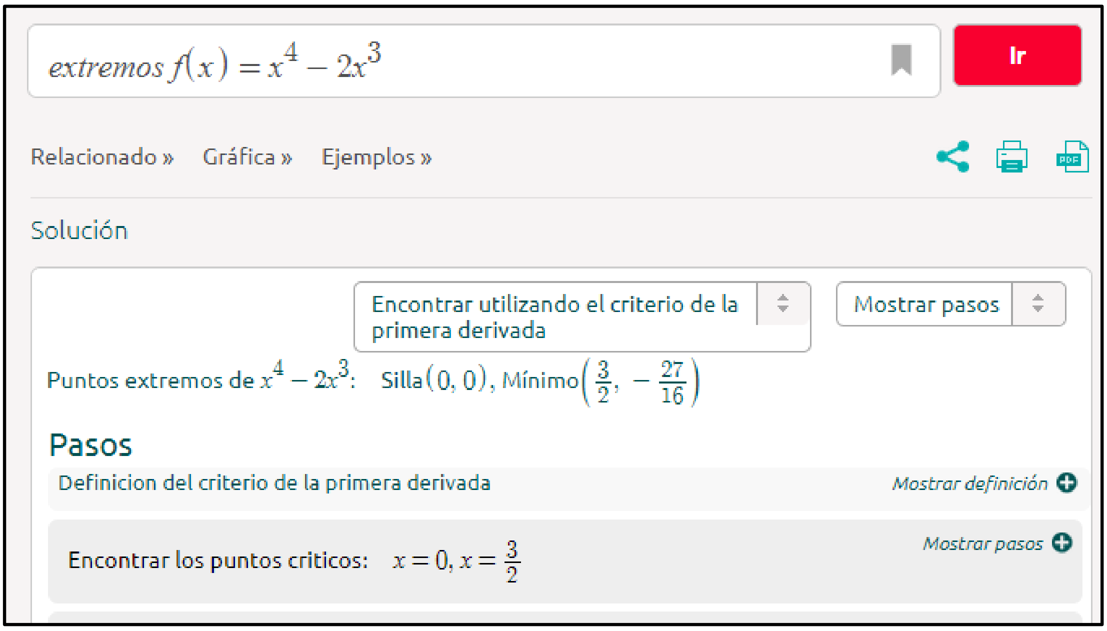


Figura N° 156 Función polinómica 3 Symbolab
Fuente: Symbolab

Función 18 $f_{(x)} = x^4 - 2x^2$

Ubica un mínimo en (-1;-1) y (1,-1) , un máximo en (0;0) como muestra la figura 157

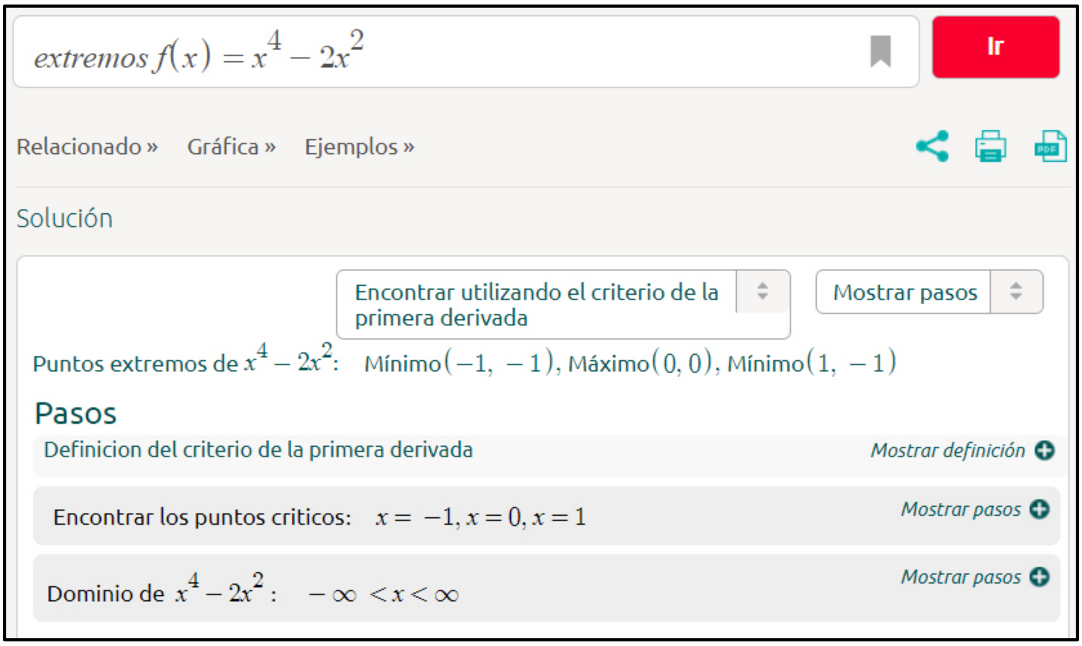


Figura N° 157 Función polinómica 4 Symbolab
Fuente: Symbolab

Funcion 19 $f_{(x)} = x^4 - 2x$

Ubica un mínimo en el punto $\left(\sqrt[3]{\frac{1}{2}}; \left(\frac{1}{2}\right)^{\frac{4}{3}} - \sqrt[3]{4}\right)$ como muestra la figura 158

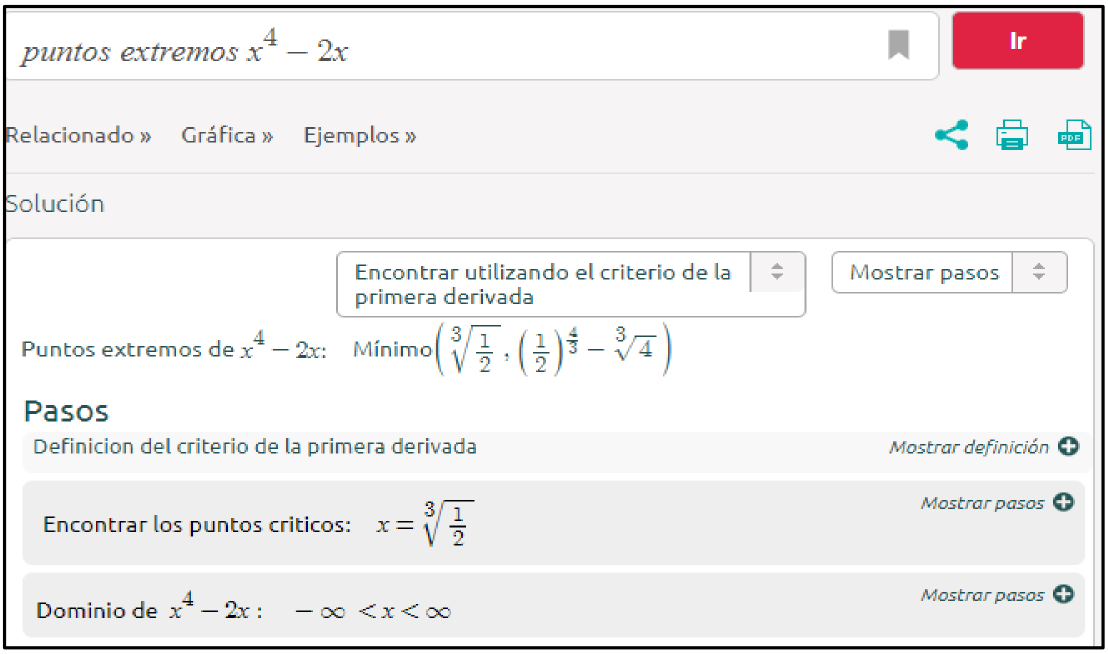


Figura N° 158 Función polinómica 5 Symbolab
Fuente: Symbolab

Funcion 20 $f_{(x)} = x^4 - 2x^3 - 3x^2$

Si se aplica el criterio de la segunda derivada muestra un máximo en (0;0) y dos mínimos en $x = \frac{3+\sqrt{33}}{4}$; $x = \frac{3-\sqrt{33}}{4}$ como indica la figura 159

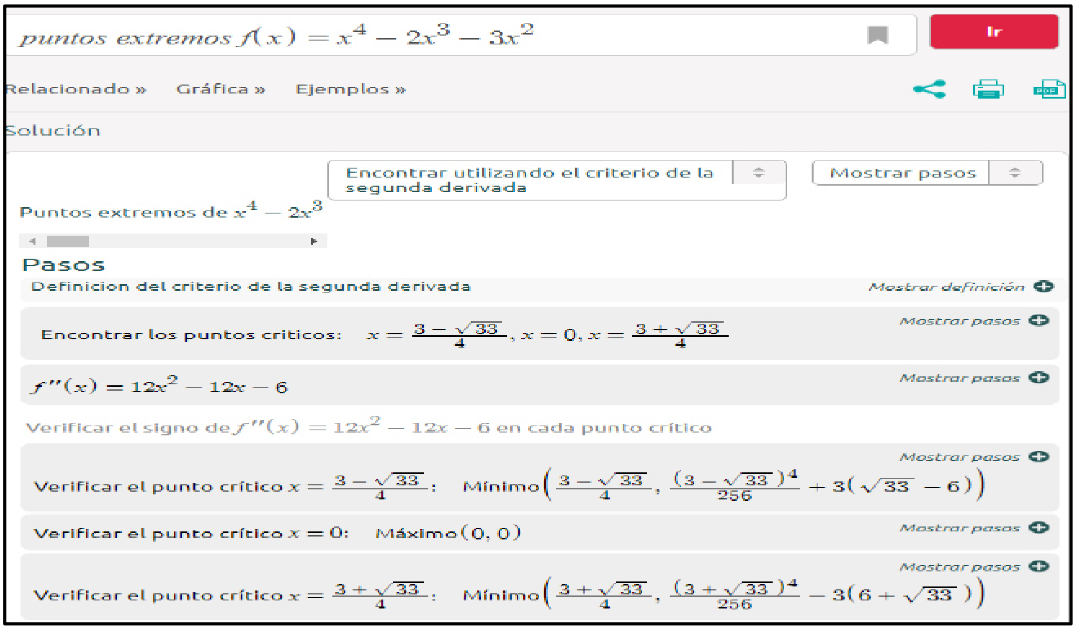


Figura N° 159 Función polinómica 6 Symbolab
Fuente: Symbolab

Funcion 21 $f_{(x)} = x^4 - x^3 - 3x$

Ubica un mínimo en (1,238;-3,26) como muestra la figura 160

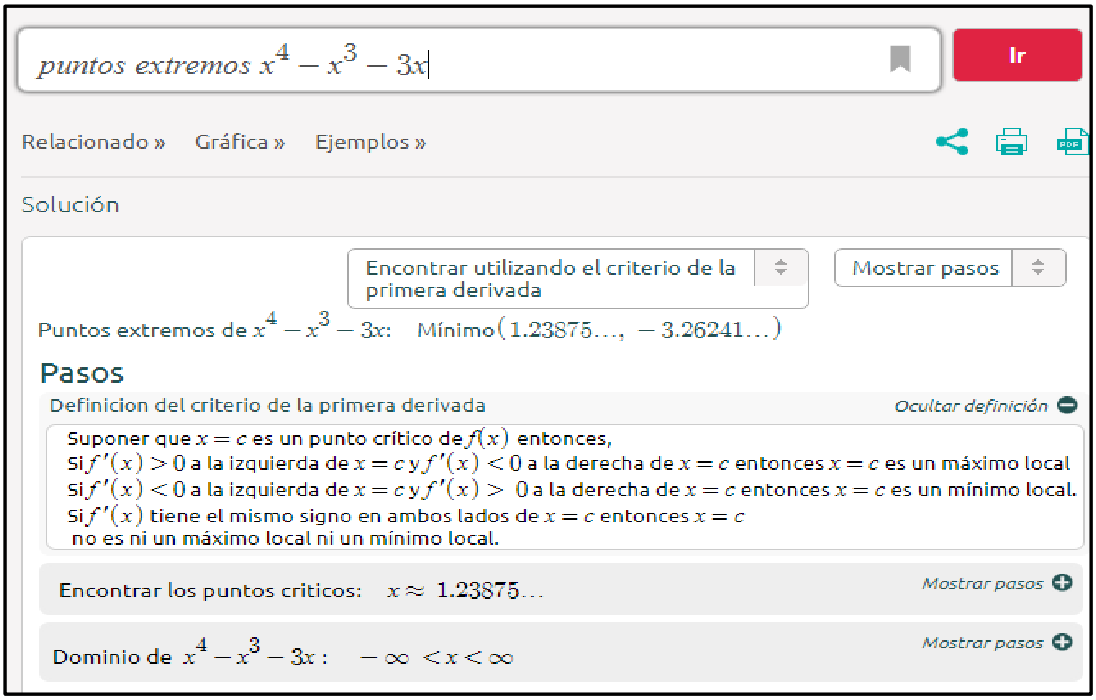


Figura N° 160 Función polinómica 7 Symbolab
Fuente: Symbolab

Funcion 22 $f_{(x)} = x^4 - x^3 - 2$

Ubica un mínimo en $\left(\frac{3}{4}; -\frac{27}{256} - 2\right)$ y un punto de silla en (0,-2) como ilustra la figura 161

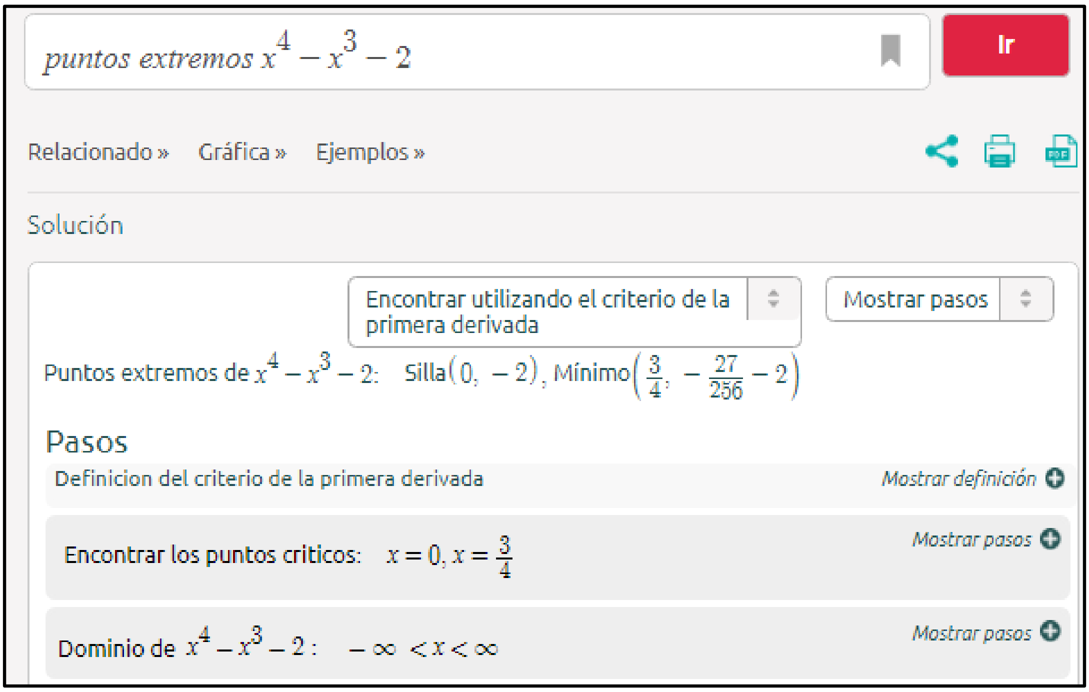


Figura N° 161 Función polinómica 8 Symbolab
Fuente: Symbolab

Funcion 23 $f_{(x)} = x^4 - 2x^2 - 3x$

Ubica un mínimo en (1,26; -4,43), como ilustra la figura 162

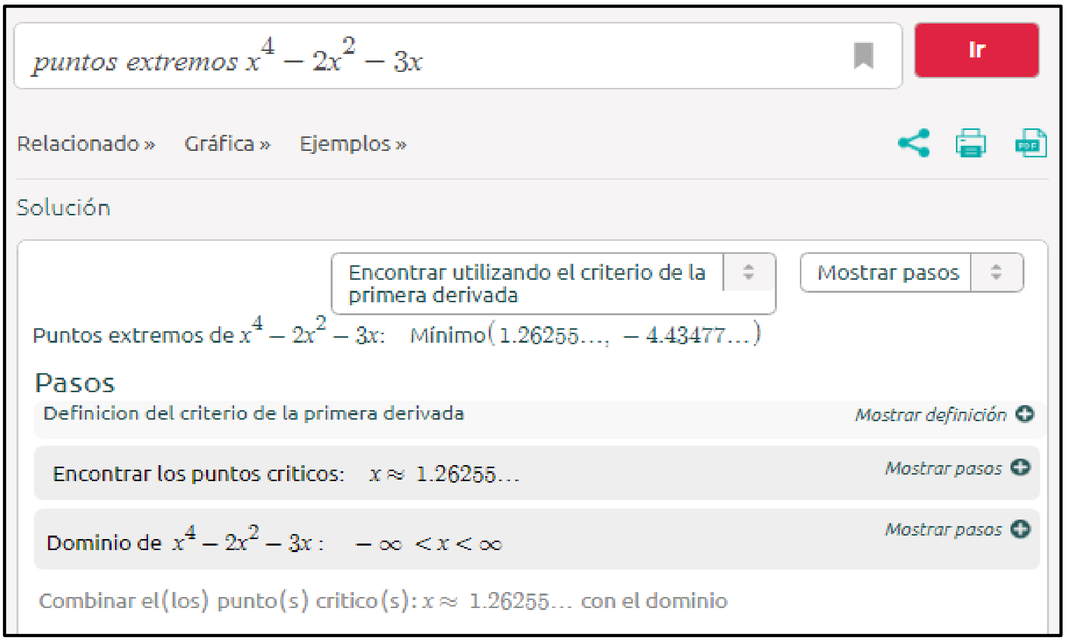


Figura N° 162 Función polinómica 9 - Symbolab
Fuente: Symbolab

Funcion 24 $f_{(x)} = x^4 - 3x^2 - 2$

Ubica un máximo en (0;-2) y dos mínimos en $\left(-\sqrt{\frac{3}{2}}; -\frac{17}{4}\right)$ y $\left(\sqrt{\frac{3}{2}}; -\frac{17}{4}\right)$ como ilustra la figura 163

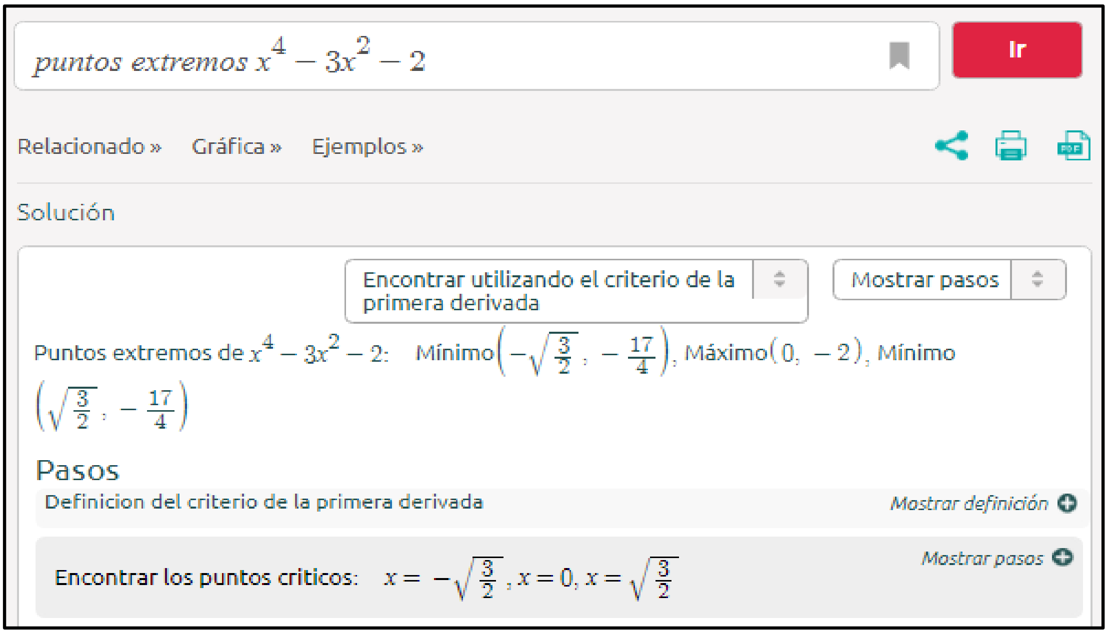


Figura N° 163 Función polinómica 10 - Symbolab
Fuente: Symbolab

Funcion 25 $f_{(x)} = x^4 - 4x + 3$

Ubica un mínimo en (1,0) como muestra la figura 164

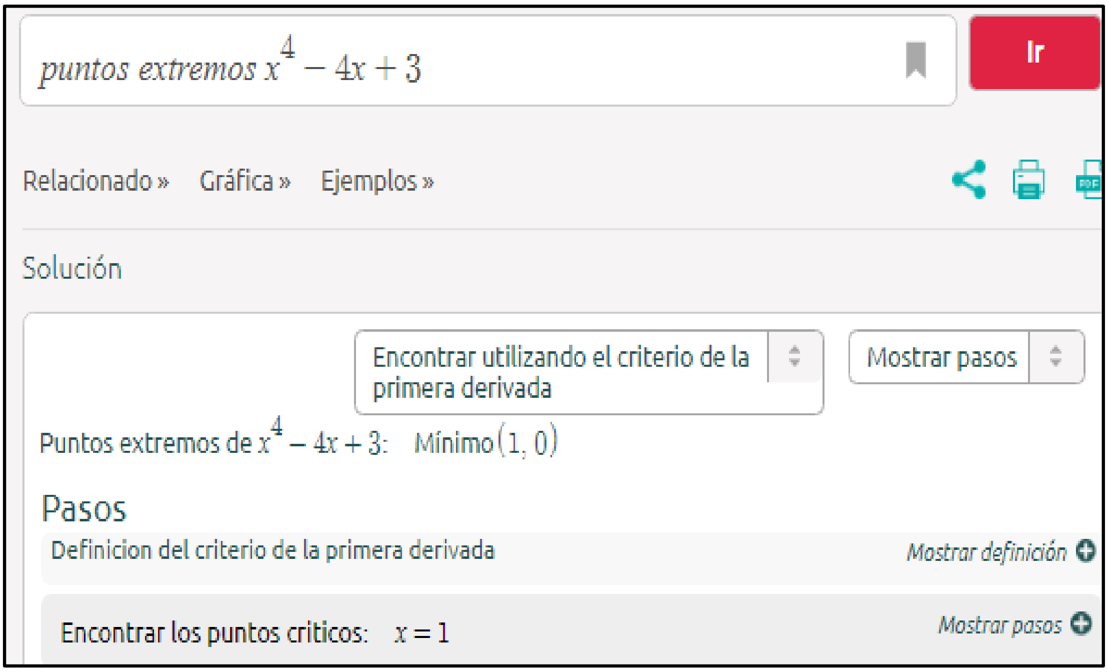


Figura N° 164 Función polinómica 11 -Symbolab
Fuente: Symbolab

Funcion 26 $f_{(x)} = x^4 - x^3 - 2x^2 - 3x$

Ubica un mínimo en (1,64; -7,47) como ilustra la figura 165

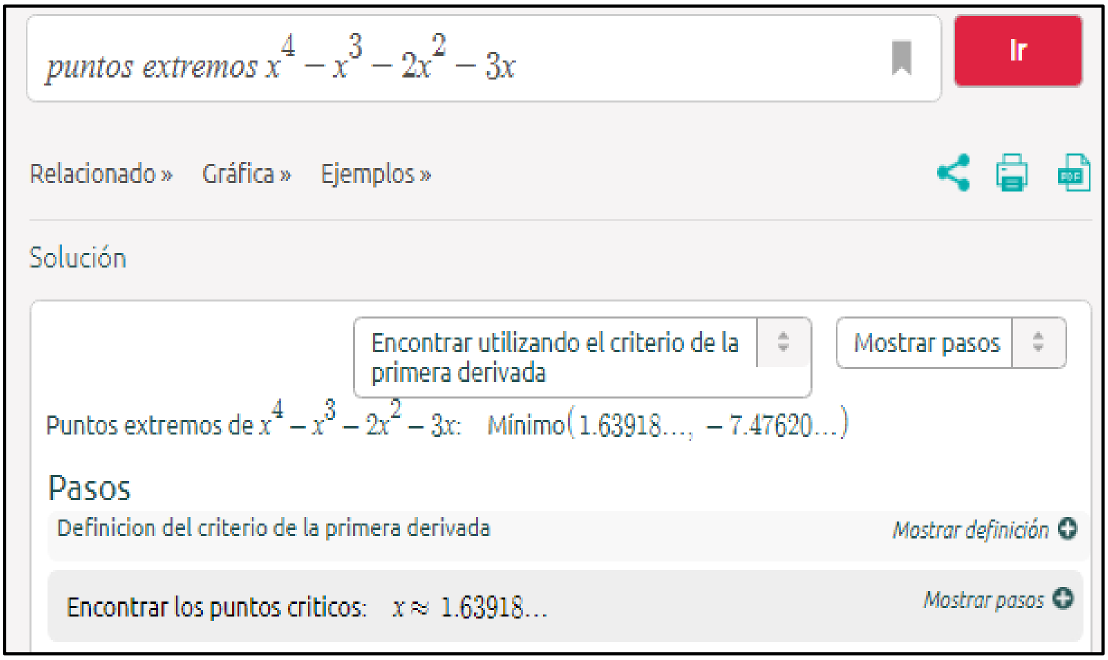


Figura N° 165 Función polinómica 12 Symbolab
Fuente: Symbolab

Funcion 27 $f_{(x)} = x^4 - 3x^3 - 2x - 2$

Ubica un mínimo en (2,34; -15,136) como muestra la figura 166

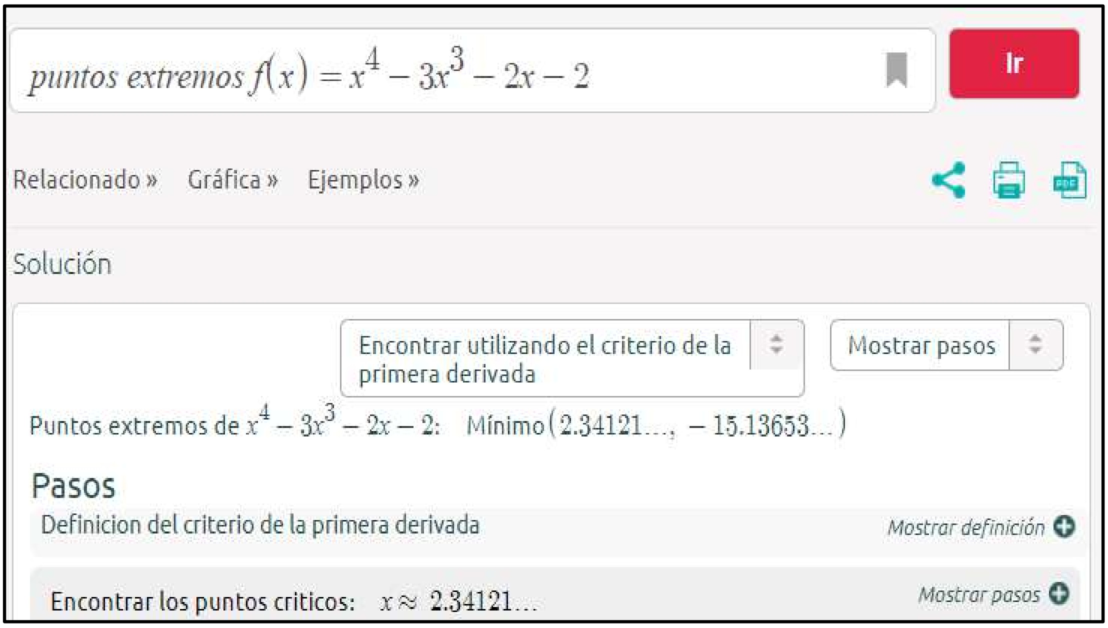


Figura N° 166 Función polinómica 13 Symbolab
Fuente: Symbolab

Funcion 28 $f_{(x)} = x^4 - 3x^2 - x - 2$

Ubica dos mínimos en (-1,13;-3,07) y (1,3; -5;51) , un máximo en (-0,17;-1,92) como muestra la figura 167.

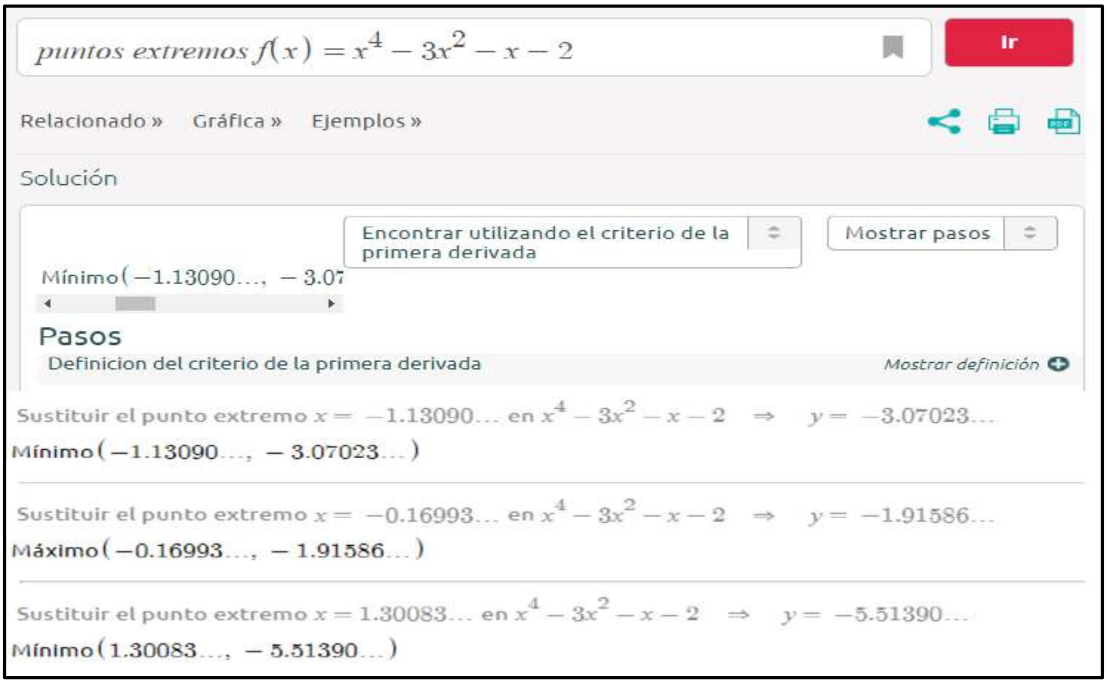


Figura N° 167 Función polinómica 14 Symbolab
Fuente: Symbolab

Funcion 29 $f_{(x)} = x^4 - x^3 - 2x^2 - x - 2$

Ubica un mínimo en (1,52;-6,31) como indica la figura 168

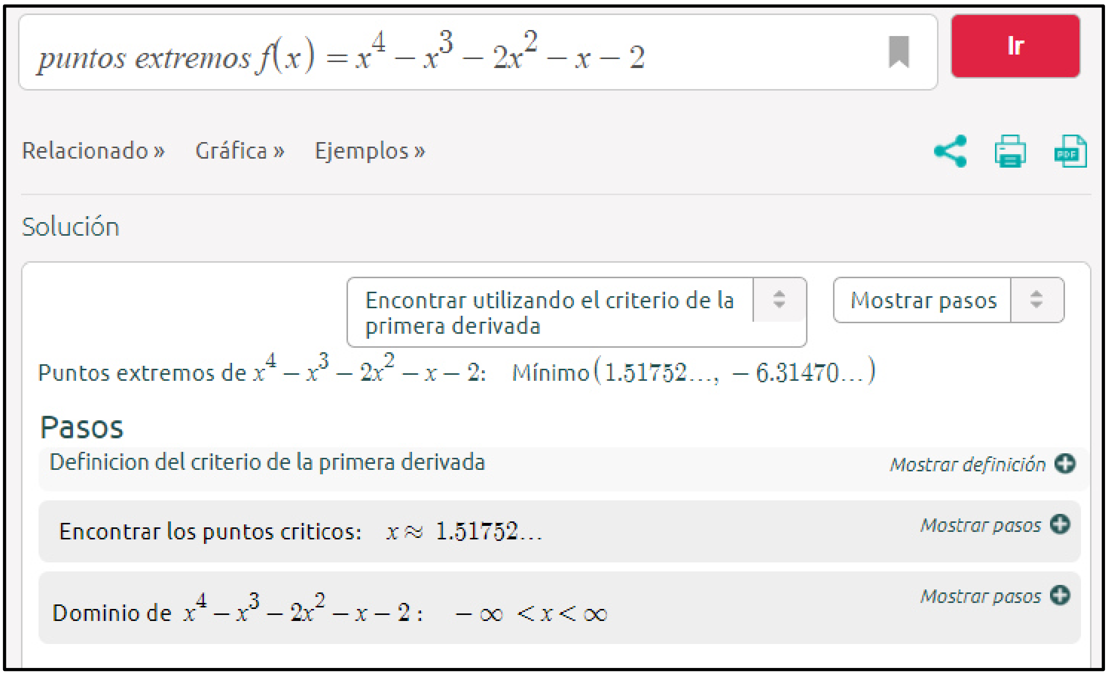


Figura N° 168 Función polinómica 15 Symbolab
Fuente: Symbolab

Funcion 30 $f_{(x)} = -x^4 + 1,8x^3 + 1,5x^2 - x + 2$

Ubica dos máximos en (-0,59;2,62) y (1,7;5,1) y un mínimo en (0,24;1,86) como ilustra la figura 169.

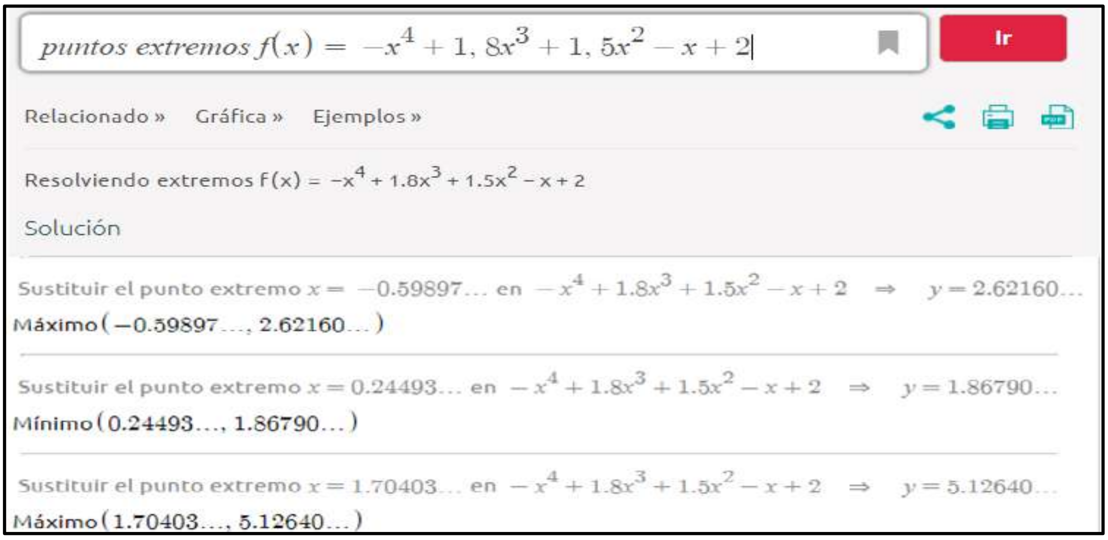


Figura N° 169 Función polinómica 16 Symbolab
Fuente: Symbolab

Funcion 31 $f_{(x)} = -0,2x^4 + x^2 - x + 2$

Ubica dos máximos en (-1,78;4,94) y (1,2;1,82) y un mínimo en (0,57;1,73) como ilustra la figura 170.

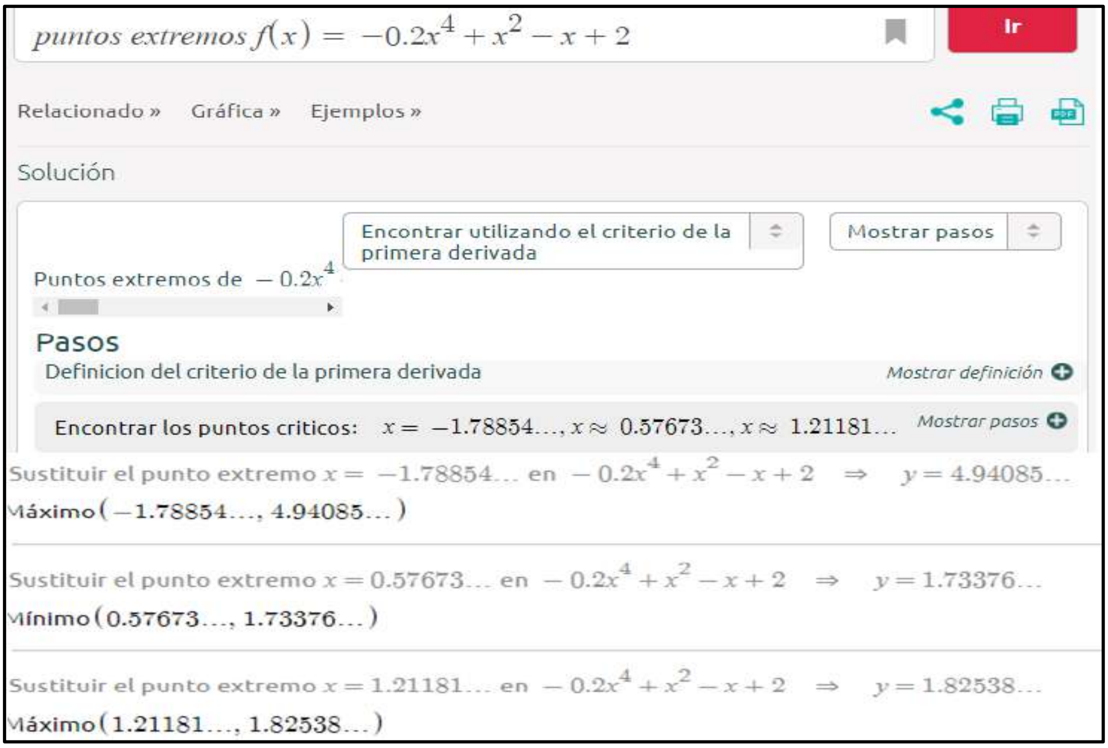


Figura N° 170 Función polinómica 17 Symbolab
Fuente: Symbolab

4.7.5 Prueba para funciones de grado 5

4.7.5.1 Ejecución con el software DERIN

Funcion 32 $f_{(x)} = 0,01x^5$

Ubica un punto de inflexión en (0;0) como muestra la figura 171

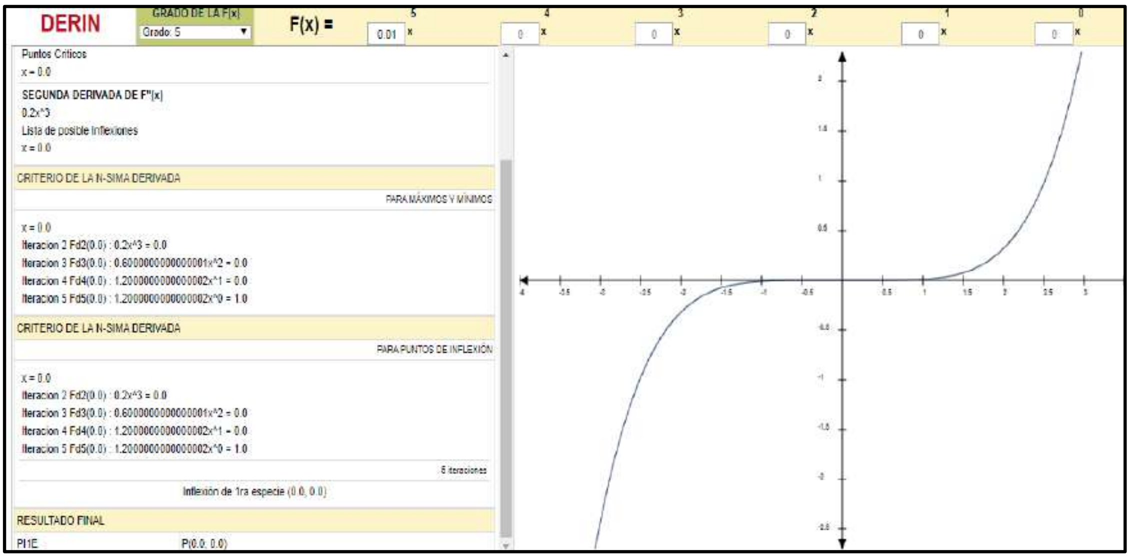


Figura N° 171 Función polinómica 18
Fuente: DERIN

Funcion 33 $f_{(x)} = x^5 - x^4$

Ubica un mínimo en (0,8; -0,0819) , el máximo local en (0;0) y una inflexión en (0,6; -0,05) como indica la figura 172

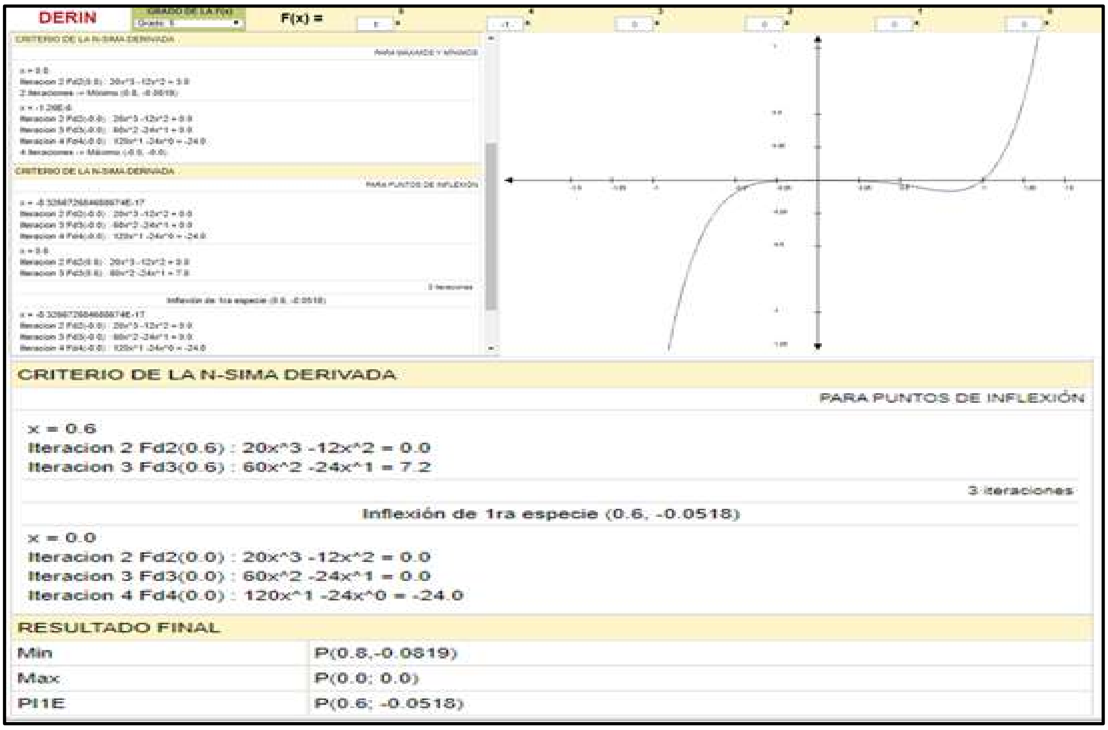


Figura N° 172 Función polinómica 19
Fuente: DERIN

Funcion 34 $f_{(x)} = 3x^5 - 5x^3$

Ubica un mínimo en (1;-2) , un máximo en (-1;2) e inflexión de primera especie en (0,707;-1,237) y (-0,707;1,237) y de segunda especie en (0;0) como muestra la figura 173

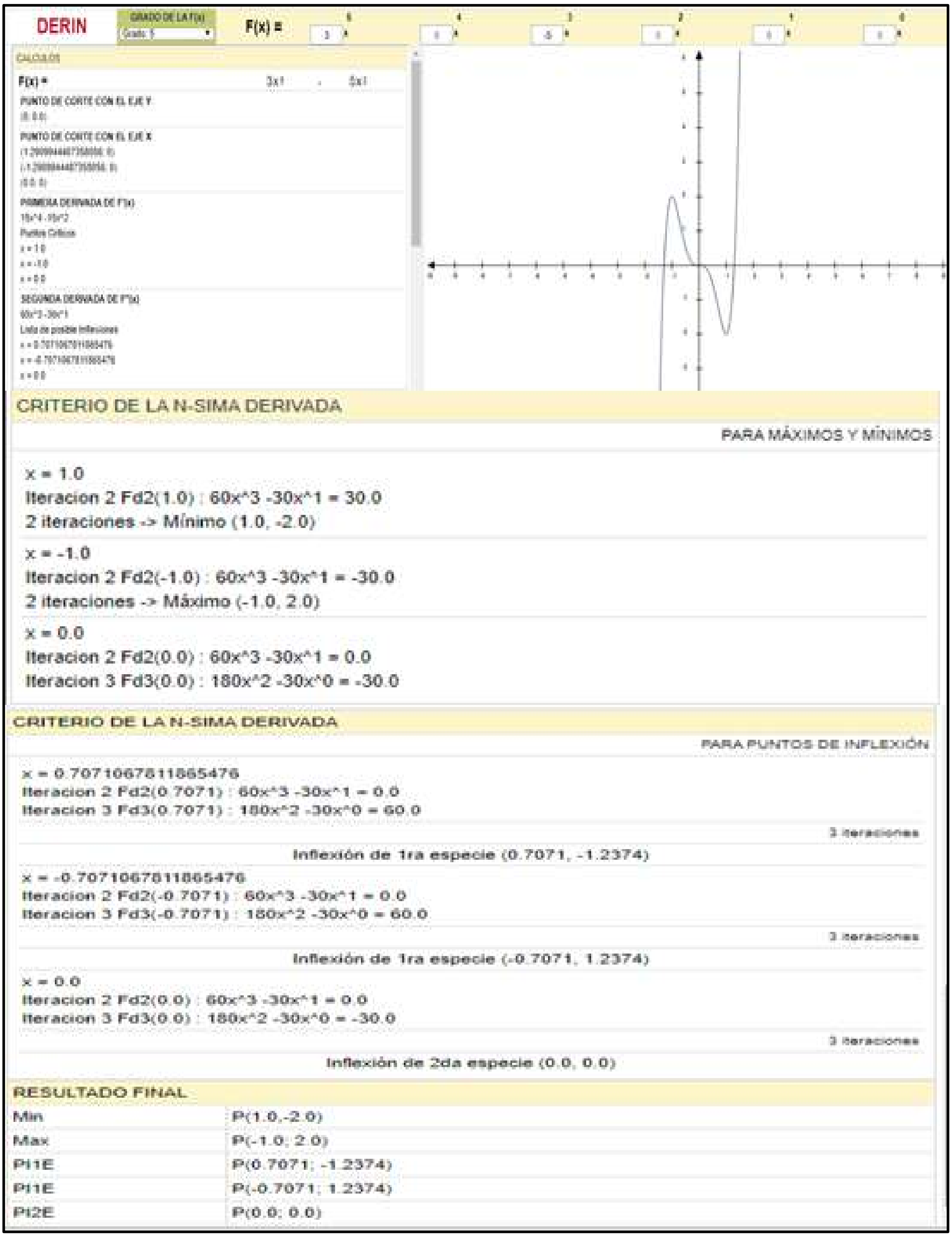


Figura N° 173 Función polinómica 20
Fuente: DERIN

Funcion 35 $f_{(x)} = x^5 - 2x^2$

Ubica un mínimo en (0,928;-1,03) y un máximo en (0;0) así como una inflexión en (0,5848;-0,6156) como indica la figura 174

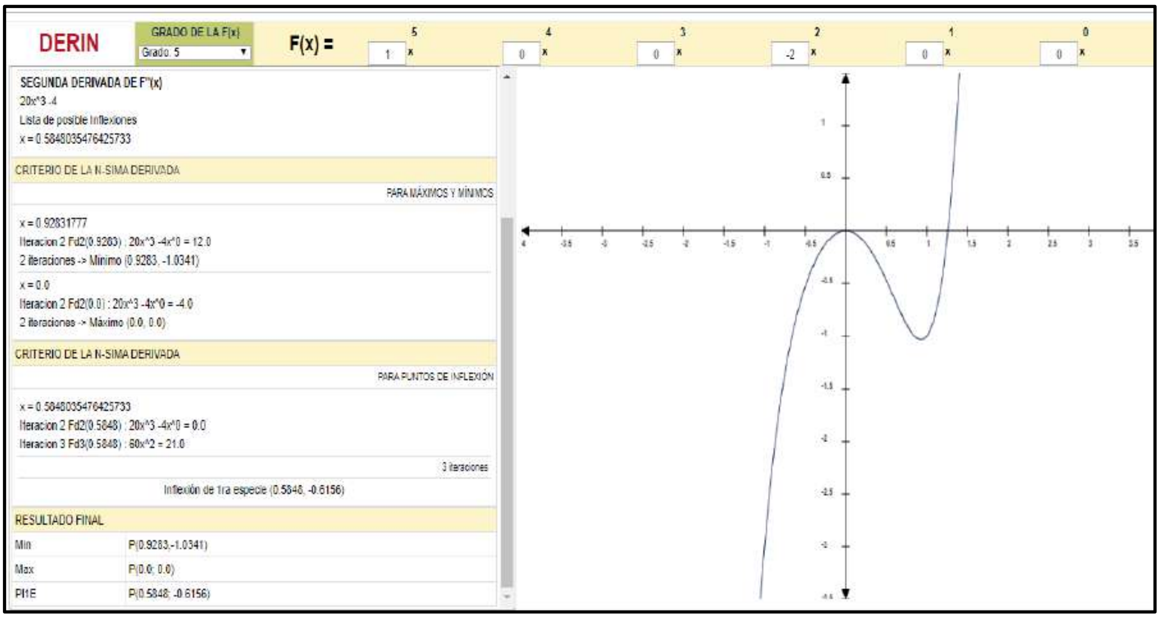


Figura N° 174 Función polinómica 21
Fuente: DERIN

Funcion 36 $f_{(x)} = x^5 - 5x$

Ubica un mínimo en (1;-4) y un máximo en (-1;4) asi como una inflexión de primera especie en (0;0) como ilustra la figura 175

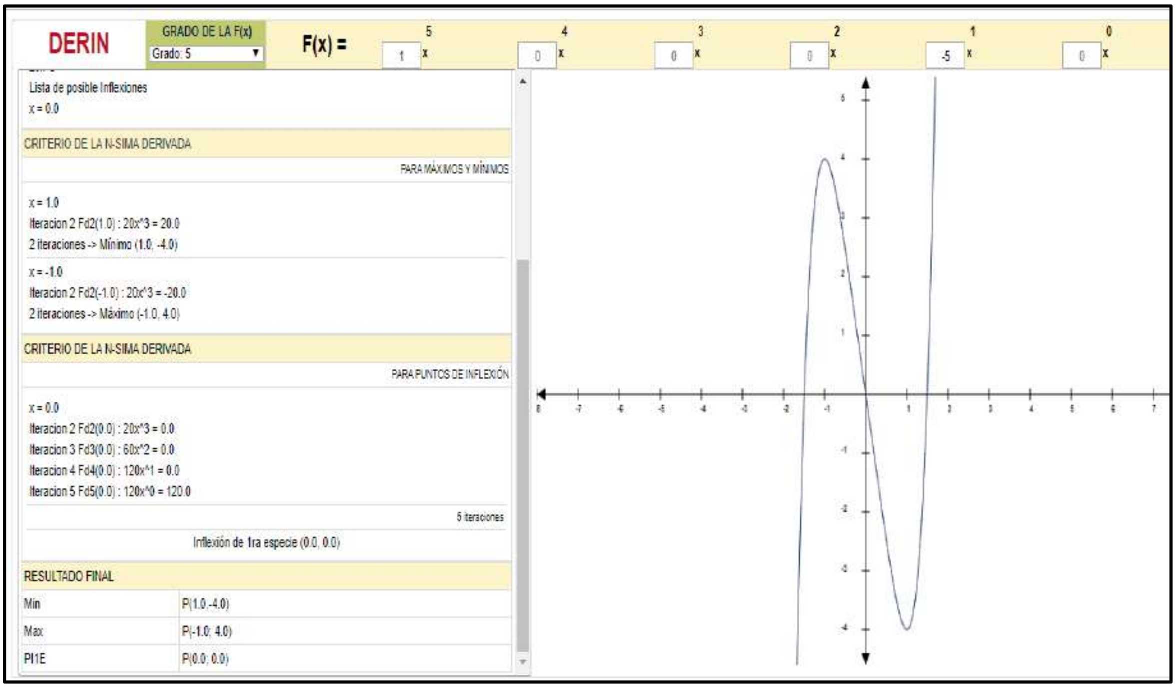


Figura N° 175 Función polinómica 22
Fuente: DERIN

Funcion 37 $f_{(x)} = x^5 + 0,5x$

Ubica un punto de inflexión en (0;0) como ilustra la figura 176

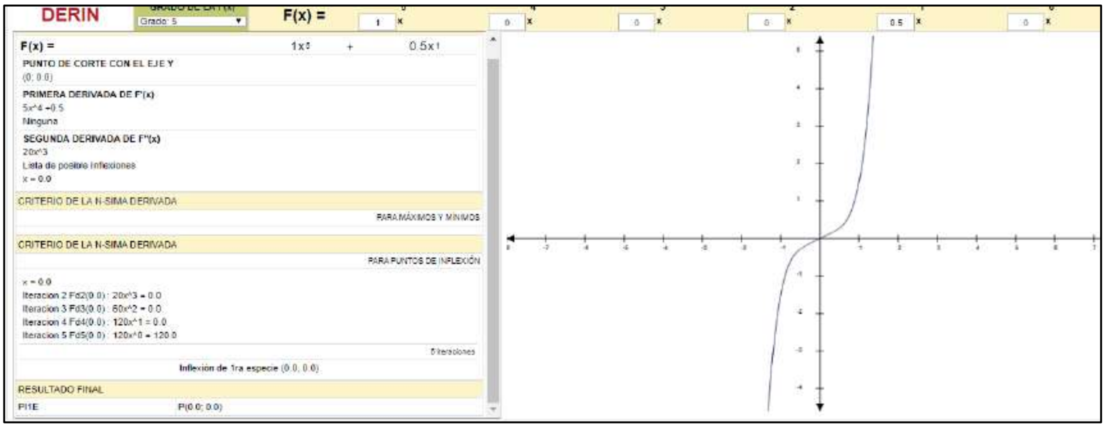


Figura N° 176 Función polinómica 23
Fuente: DERIN

Funcion 38 $f_{(x)} = x^5 - 4x^4 + 4x^3$

Ubica un mínimo en (2;0) , un máximo en (1,2;1,11) e inflexión de primera especie en (0;0) y (1,69; 0,46) , en (0,71; 0,59) de segunda especie como ilustra la figura 177

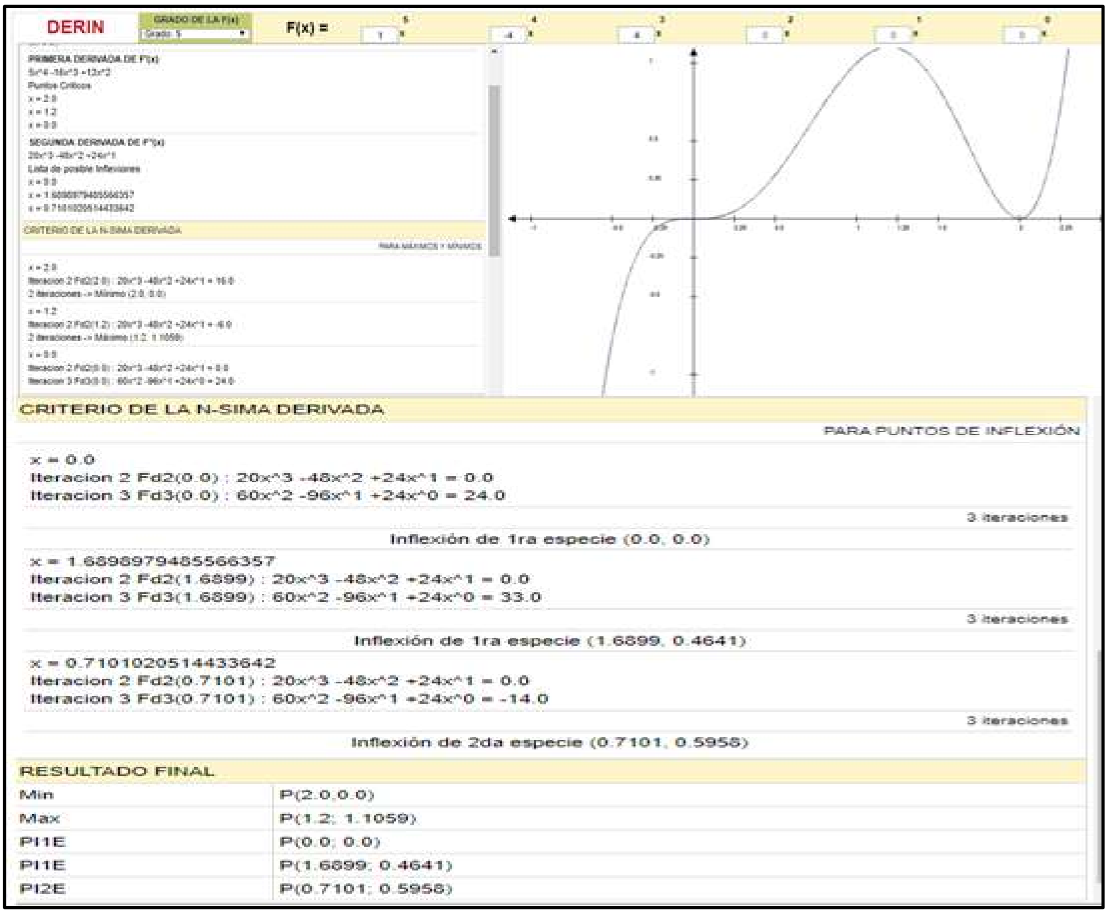


Figura N° 177 Función polinómica 24
Fuente: DERIN

Funcion 39 $f_{(x)} = x^5 - 3x^4 + 3x^2$

Ubica dos máximos en (-0,629;0,618) y en (0,892;1,052) , dos mínimos en (0;0) y (2,137;-4,297) e inflexión de primera especie en (-0,372;0,35) y (1,696;-2,157) y de segunda especie en (0,476;0,55) como muestra en las figuras 178 y 179

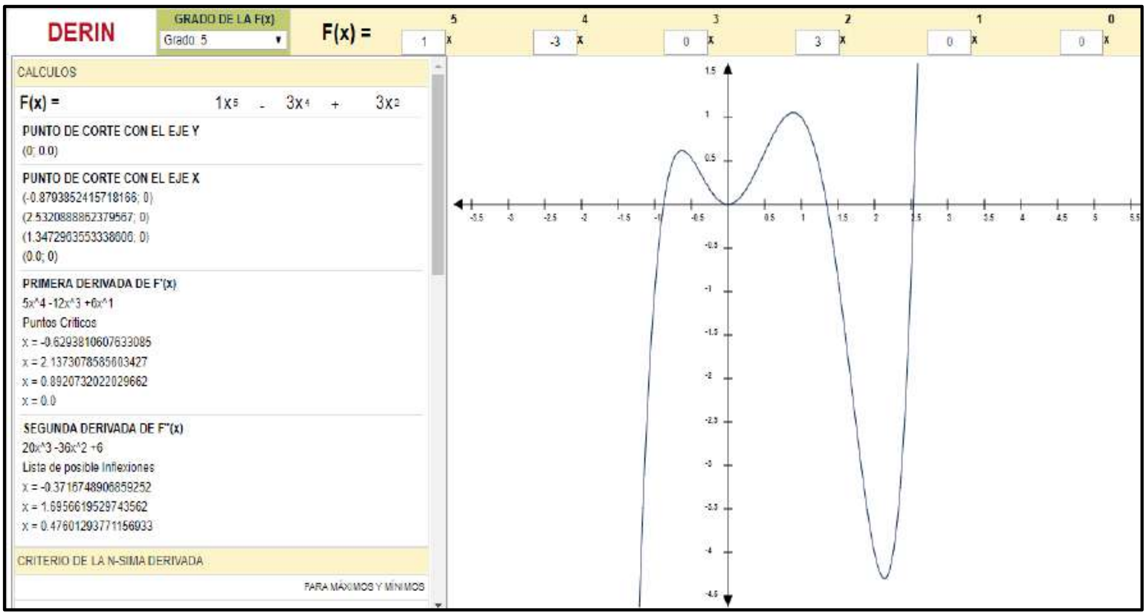
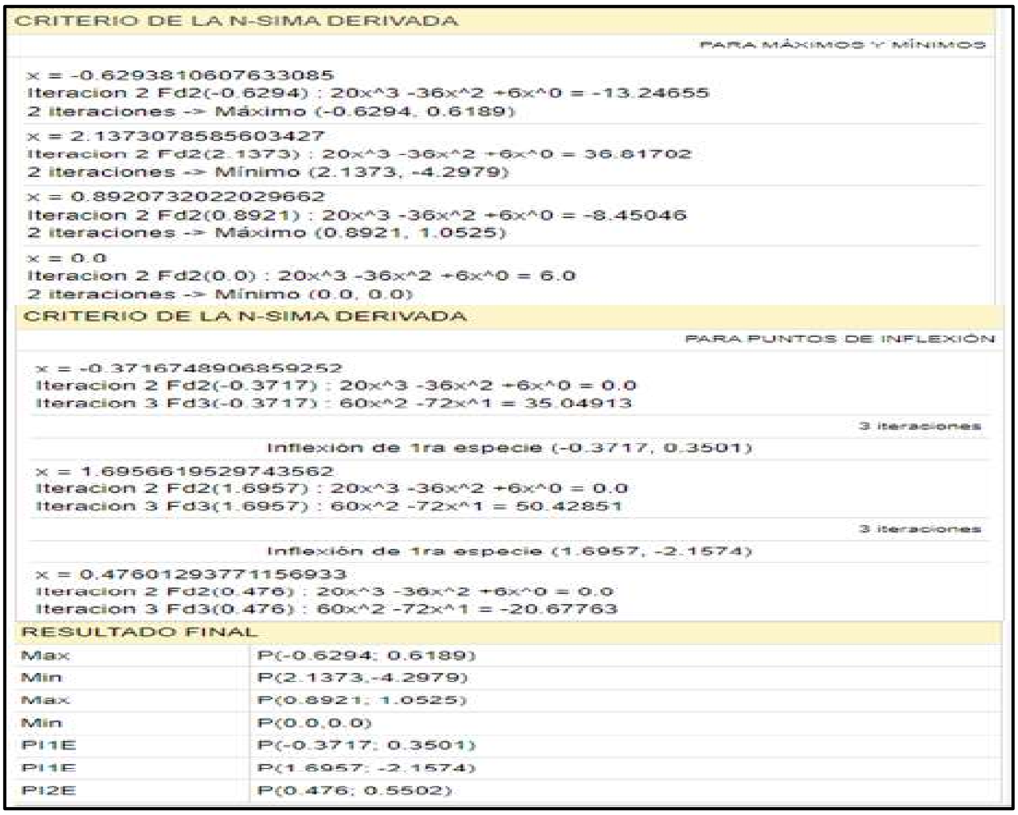


Figura N° 178 Función polinómica 25A
Fuente: DERIN



CRITERIO DE LA N-SIMA DERIVADA

PARA MÁXIMOS Y MÍNIMOS

x = -0.6293810607633085
Iteracion 2 Fd2(-0.6294) : 20x^3 -36x^2 +6x^0 = -13.24655
2 iteraciones -> Máximo (-0.6294, 0.6189)

x = 2.1373078585603427
Iteracion 2 Fd2(2.1373) : 20x^3 -36x^2 +6x^0 = 36.81702
2 iteraciones -> Mínimo (2.1373, -4.2979)

x = 0.8920732022029662
Iteracion 2 Fd2(0.8921) : 20x^3 -36x^2 +6x^0 = -8.45046
2 iteraciones -> Máximo (0.8921, 1.0525)

x = 0.0
Iteracion 2 Fd2(0.0) : 20x^3 -36x^2 +6x^0 = 6.0
2 iteraciones -> Mínimo (0.0, 0.0)

CRITERIO DE LA N-SIMA DERIVADA

PARA PUNTOS DE INFLEXIÓN

x = -0.3716748906859252
Iteracion 2 Fd2(-0.3717) : 20x^3 -36x^2 +6x^0 = 0.0
Iteracion 3 Fd3(-0.3717) : 60x^2 -72x^1 = 35.04913

3 iteraciones

Inflexión de 1ra especie (-0.3717, 0.3501)

x = 1.6956619529743562
Iteracion 2 Fd2(1.6957) : 20x^3 -36x^2 +6x^0 = 0.0
Iteracion 3 Fd3(1.6957) : 60x^2 -72x^1 = 50.42851

3 iteraciones

Inflexión de 1ra especie (1.6957, -2.1574)

x = 0.4760129377115693
Iteracion 2 Fd2(0.476) : 20x^3 -36x^2 +6x^0 = 0.0
Iteracion 3 Fd3(0.476) : 60x^2 -72x^1 = -20.67763

RESULTADO FINAL

Max	P(-0.6294; 0.6189)
Min	P(2.1373,-4.2979)
Max	P(0.8921; 1.0525)
Min	P(0.0,0.0)
PI1E	P(-0.3717; 0.3501)
PI1E	P(1.6957; -2.1574)
PI2E	P(0.476; 0.5502)

Figura N° 179 Función polinómica 25B
Fuente: DERIN

Funcion 40 $f_{(x)} = x^5 - 2x^4 - 2x$

Ubica un mínimo en (1,68; -5,91) , un máximo en (-0,57;0,87) y una inflexión de primera especie en (1,2;-4,06) como muestran las figuras 180 y 181

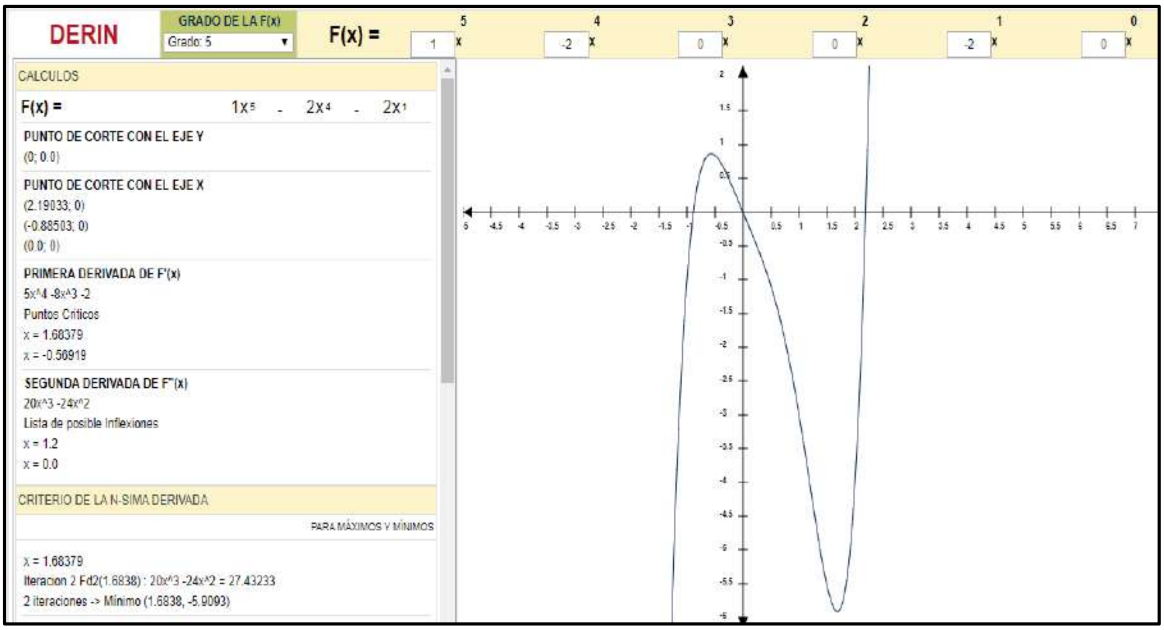


Figura N° 180 Función polinómica 26A
Fuente: DERIN

CRITERIO DE LA N-SIMA DERIVADA

PARA MÁXIMOS Y MÍNIMOS

x = 1.68379
Iteracion 2 Fd2(1.6838) : 20x^3 -24x^2 = 27.43233
2 iteraciones -> Mínimo (1.6838, -5.9093)

x = -0.56919
Iteracion 2 Fd2(-0.5692) : 20x^3 -24x^2 = -11.46355
2 iteraciones -> Máximo (-0.5692, 0.8687)

CRITERIO DE LA N-SIMA DERIVADA

PARA PUNTOS DE INFLEXIÓN

x = 1.2
Iteracion 2 Fd2(1.2) : 20x^3 -24x^2 = 0.0
Iteracion 3 Fd3(1.2) : 60x^2 -48x^1 = 28.8

3 iteraciones

Inflexión de 1ra especie (1.2, -4.0589)

x = 0.0
Iteracion 2 Fd2(0.0) : 20x^3 -24x^2 = 0.0
Iteracion 3 Fd3(0.0) : 60x^2 -48x^1 = 0.0
Iteracion 4 Fd4(0.0) : 120x^1 -48x^0 = -48.0

RESULTADO FINAL

Min	P(1.6838,-5.9093)
Max	P(-0.5692; 0.8687)
PI1E	P(1.2; -4.0589)

Figura N° 181 Función polinómica 26B
Fuente: DERIN

Funcion 41 $f_{(x)} = 0,2x^5 - 0,5x^4 - 1$

Ubica un máximo en (0;-1) , un mínimo en (2;-2,6) asi como una inflexión de primera especie en (1,5; -2,0125) tal como muestran las figuras 182 y 183

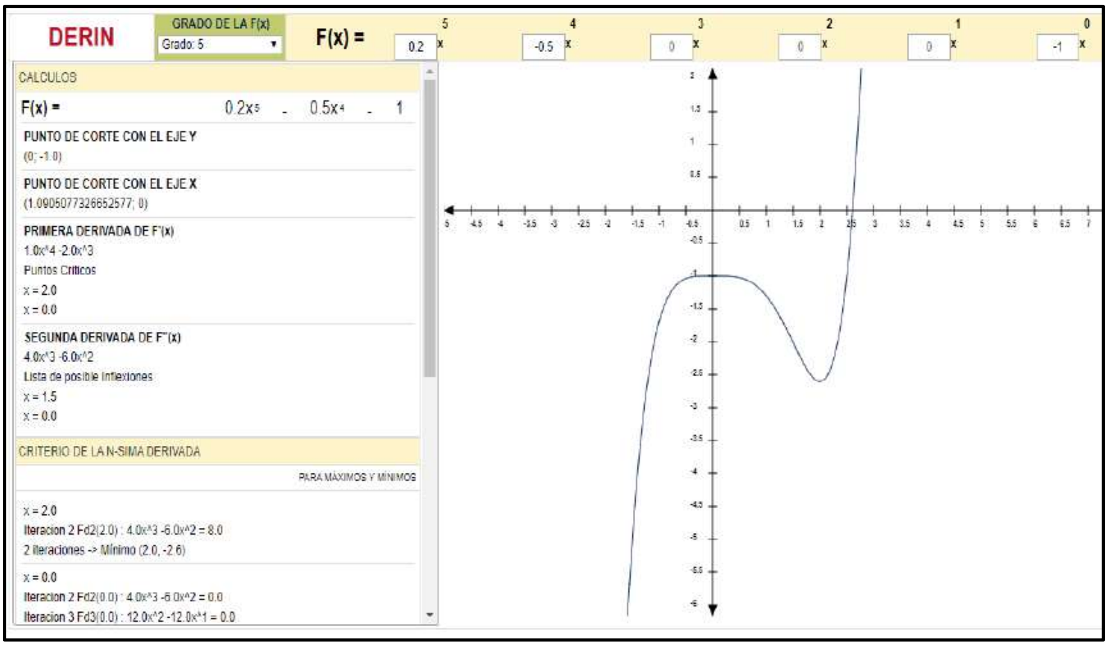
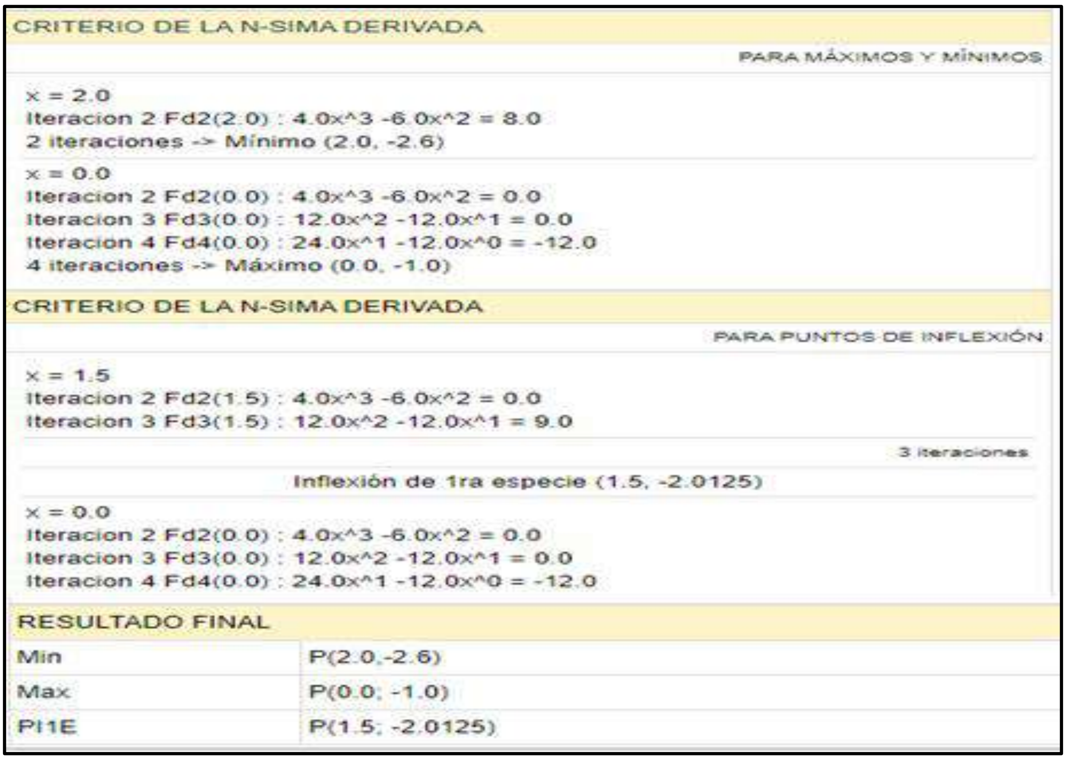


Figura N° 182 Función polinómica 27A
Fuente: DERIN



CRITERIO DE LA N-SIMA DERIVADA

PARA MÁXIMOS Y MÍNIMOS

x = 2.0
Iteracion 2 Fd2(2.0) : 4.0x^3 -6.0x^2 = 8.0
2 iteraciones -> Mínimo (2.0, -2.6)

x = 0.0
Iteracion 2 Fd2(0.0) : 4.0x^3 -6.0x^2 = 0.0
Iteracion 3 Fd3(0.0) : 12.0x^2 -12.0x^1 = 0.0
Iteracion 4 Fd4(0.0) : 24.0x^1 -12.0x^0 = -12.0
4 iteraciones -> Máximo (0.0, -1.0)

CRITERIO DE LA N-SIMA DERIVADA

PARA PUNTOS DE INFLEXIÓN

x = 1.5
Iteracion 2 Fd2(1.5) : 4.0x^3 -6.0x^2 = 0.0
Iteracion 3 Fd3(1.5) : 12.0x^2 -12.0x^1 = 9.0

3 iteraciones

Inflexión de 1ra especie (1.5, -2.0125)

x = 0.0
Iteracion 2 Fd2(0.0) : 4.0x^3 -6.0x^2 = 0.0
Iteracion 3 Fd3(0.0) : 12.0x^2 -12.0x^1 = 0.0
Iteracion 4 Fd4(0.0) : 24.0x^1 -12.0x^0 = -12.0

RESULTADO FINAL

Min	P(2.0,-2.6)
Max	P(0.0; -1.0)
PI1E	P(1.5; -2.0125)

Figura N° 183 Función polinómica 27B
Fuente: DERIN

Función 42 $f_{(x)} = x^5 - 2x^3 - 3x^2$

Ubica un máximo en (0;0) ,un mínimo en (1,428;-6,005) y una inflexión de primera especie en (0,955;-3;69) como muestra la figura 184

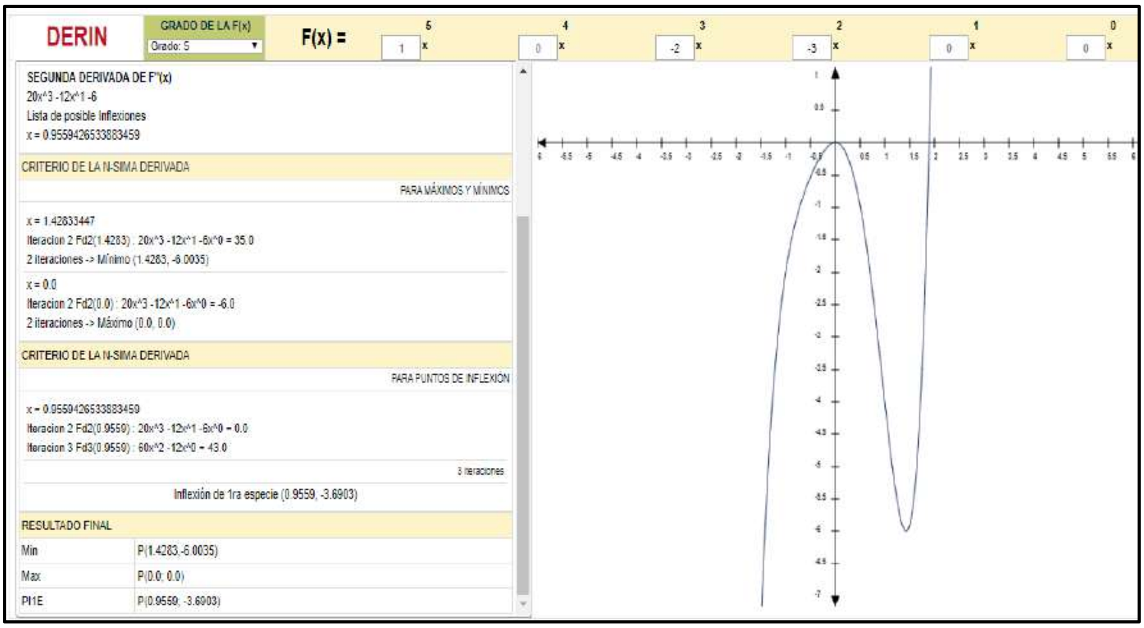


Figura N° 184 Función polinómica 28
Fuente: DERIN

Funcion 43 $f_{(x)} = x^5 - 1.5x^3 + 1.2x^2$

Ubica un máximo en (-1,148;1,857) , un mínimo en (0;0) y una inflexión de primera especie en (-0,7774;1,146) como muestra la figura 185

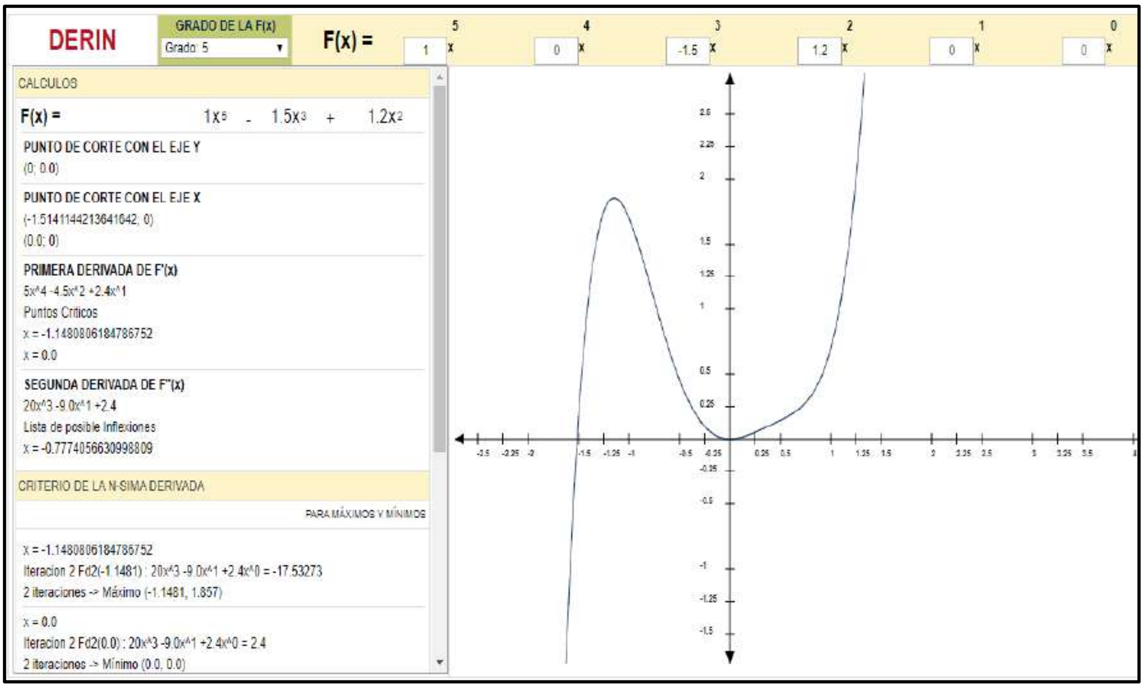


Figura N° 185 Función polinómica 29A
Fuente: DERIN

En la figura 174 se presenta el resumen mediante la derivada enésima para alcanzar los resultados presentados.

CRITERIO DE LA N-SIMA DERIVADA

PARA MÁXIMOS Y MÍNIMOS

x = -1.1480806184786752
Iteracion 2 Fd2(-1.1481) : 20x^3 -9.0x^1 +2.4x^0 = -17.53273
2 iteraciones -> Máximo (-1.1481, 1.857)

x = 0.0
Iteracion 2 Fd2(0.0) : 20x^3 -9.0x^1 +2.4x^0 = 2.4
2 iteraciones -> Mínimo (0.0, 0.0)

CRITERIO DE LA N-SIMA DERIVADA

PARA PUNTOS DE INFLEXIÓN

x = -0.7774056630998809
Iteracion 2 Fd2(-0.7774) : 20x^3 -9.0x^1 +2.4x^0 = 0.0
Iteracion 3 Fd3(-0.7774) : 60x^2 -9.0x^0 = 27.26157

3 iteraciones

Inflexión de 1ra especie (-0.7774, 1.146)

RESULTADO FINAL	
Max	P(-1.1481; 1.857)
Min	P(0.0,0.0)
PI1E	P(-0.7774; 1.146)

Figura N° 186 Función polinómica 29B
Fuente: DERIN

Funcion 44 $f_{(x)} = x^5 - 2x^3 + 2x$

Ubica inflexiones de primera especie en (0,7746,0,8985) y en (-0,7746,-0,8985) y en (0;0) una de segunda especie como muestra la figura 187.

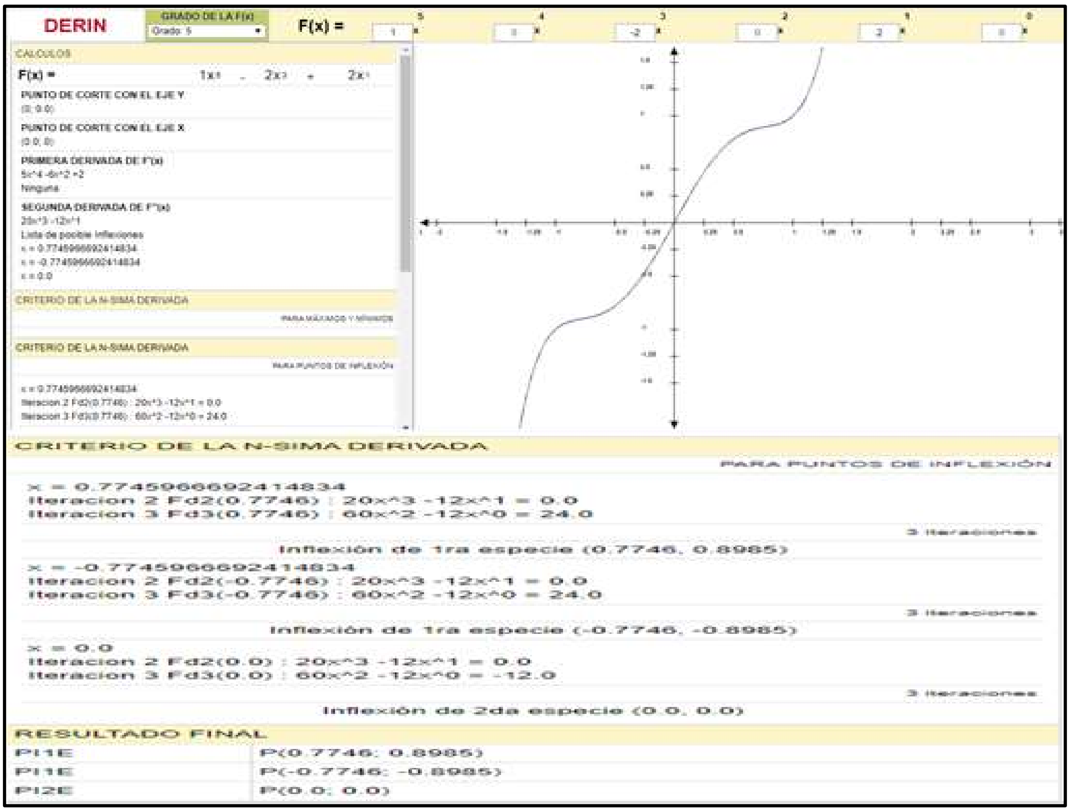


Figura N° 187 Función polinómica 30
Fuente: DERIN

Funcion 45 $f_{(x)} = x^5 - 2x^3 + x$

Ubica dos mínimos en (-1;0) y en (-0,447;-0,286) , dos máximos en (1;0) y en (0,447;0,286) e inflexiones de primerea especie en (0,7746;0,1239) y (-0,7746;-0,1239) y en (0;0) una de segunda especie como muestra las figuras 188 y 189

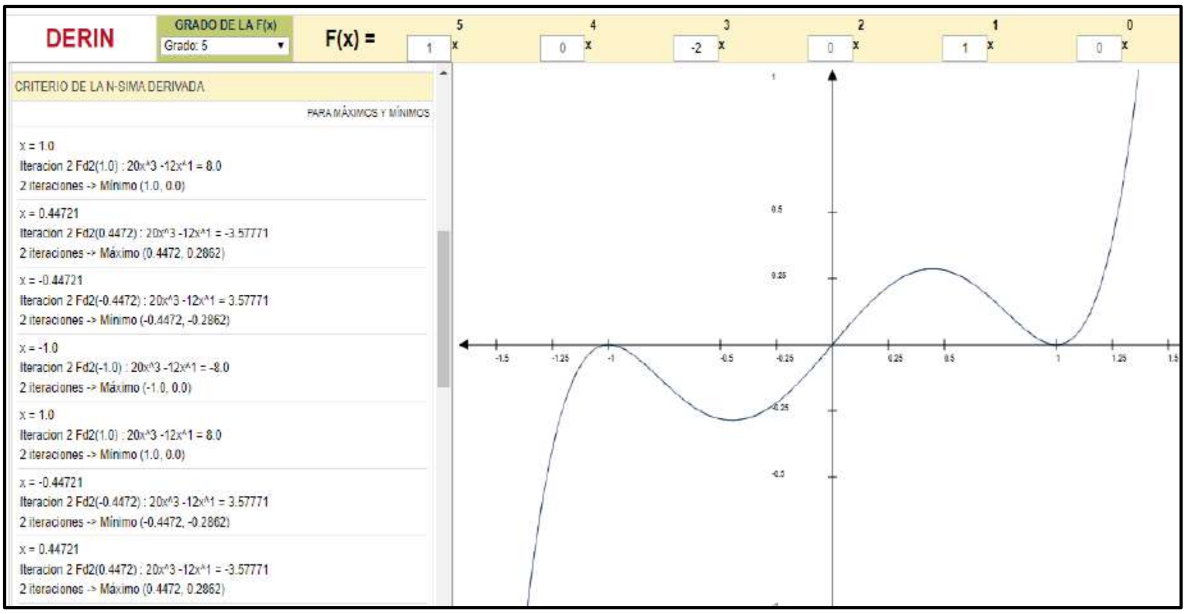


Figura N° 188 Función polinómica 31A
Fuente: DERIN

Figura N° 189 Función polinómica 31B
Fuente: DERIN

Funcion 46 $f_{(x)} = x^5 - x^3 + 1$

Ubica un mínimo en (0,7746;0,8141) y un máximo en (-0,7746;1,1859) , dos inlfexiones de primer orden en (0,5477;0,885) y (-0,5477; 1,115) y una inflexión de segundo orden en (0;1) como muestra la figura 190

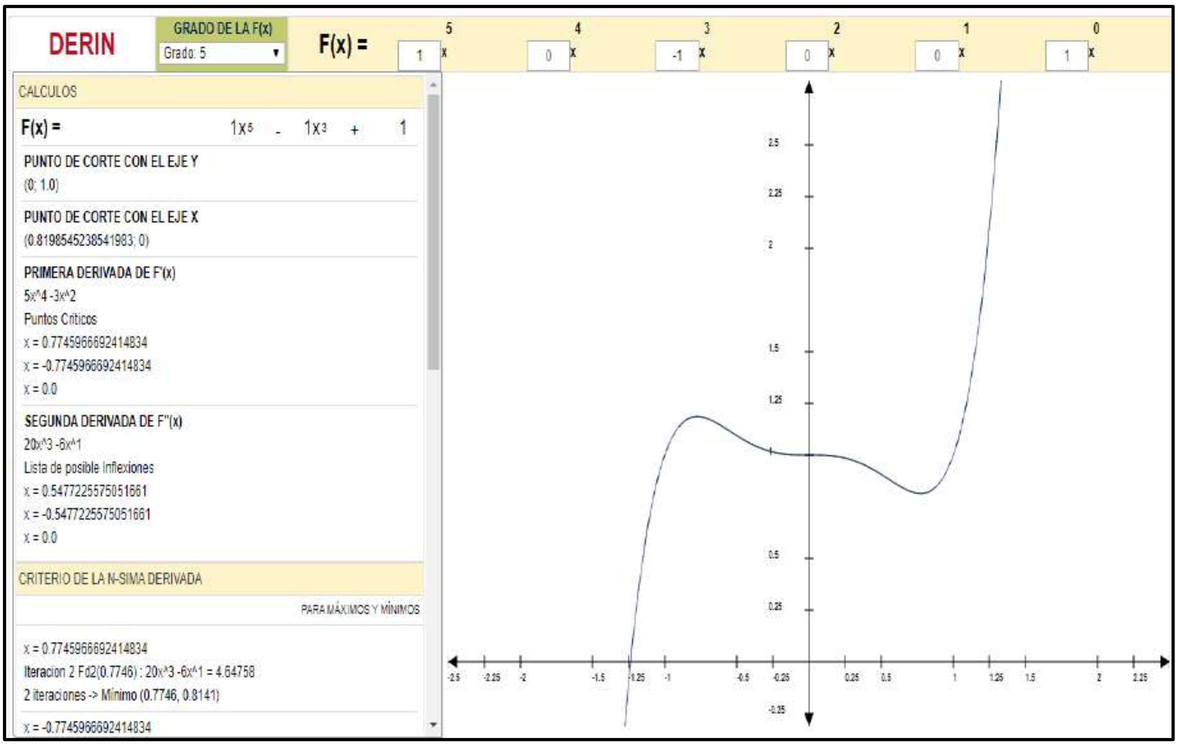


Figura N° 190 Función polinómica 32A
Fuente: DERIN

En la figura 191 se presenta el resumen de proceso de la derivada enesima aplicada a la funcion

CRITERIO DE LA N-SIMA DERIVADA

PARA MÁXIMOS Y MÍNIMOS

x = 0.7745966692414834
Iteracion 2 Fd2(0.7746) : 20x^3 -6x^1 = 4.64758
2 iteraciones -> Mínimo (0.7746, 0.8141)

x = -0.7745966692414834
Iteracion 2 Fd2(-0.7746) : 20x^3 -6x^1 = -4.64758
2 iteraciones -> Máximo (-0.7746, 1.1859)

x = 0.0
Iteracion 2 Fd2(0.0) : 20x^3 -6x^1 = 0.0
Iteracion 3 Fd3(0.0) : 60x^2 -6x^0 = -6.0

PARA PUNTOS DE INFLEXIÓN

x = 0.5477225575051661
Iteracion 2 Fd2(0.5477) : 20x^3 -6x^1 = 0.0
Iteracion 3 Fd3(0.5477) : 60x^2 -6x^0 = 12.0

3 iteraciones

Inflexión de 1ra especie (0.5477, 0.885)

x = -0.5477225575051661
Iteracion 2 Fd2(-0.5477) : 20x^3 -6x^1 = 0.0
Iteracion 3 Fd3(-0.5477) : 60x^2 -6x^0 = 12.0

3 iteraciones

Inflexión de 1ra especie (-0.5477, 1.115)

x = 0.0
Iteracion 2 Fd2(0.0) : 20x^3 -6x^1 = 0.0
Iteracion 3 Fd3(0.0) : 60x^2 -6x^0 = -6.0

3 iteraciones

Inflexión de 2da especie (0.0, 1.0)

RESULTADO FINAL

Min	P(0.7746,0.8141)
Max	P(-0.7746, 1.1859)
PI1E	P(0.5477; 0.885)
PI1E	P(-0.5477, 1.115)
PI2E	P(0.0; 1.0)

Figura N° 191 Función polinómica 32B
Fuente: DERIN

Funcion 47 $f_{(x)} = x^5 + x^2 - 2x$

Ubica un mínimo en (0,6231;-0,764), un máximo en (-0,9384;2,0297) e inflexión de primera especie en (-0,4642;1,1222) como muestra la figura 192

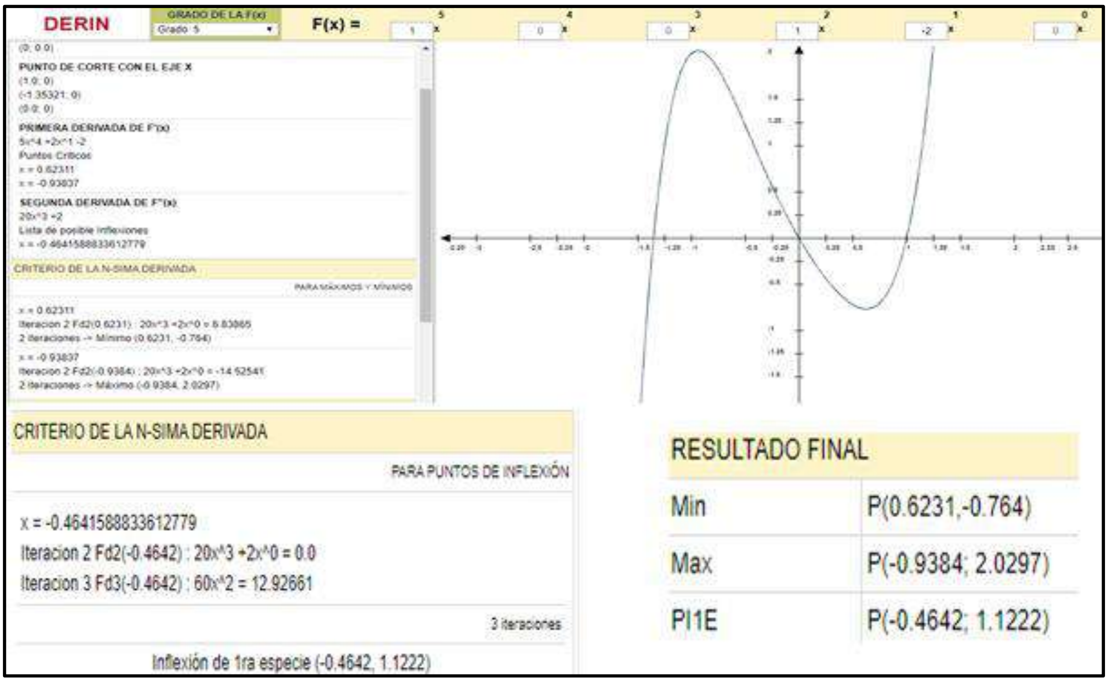


Figura N° 192 Función polinómica 33
Fuente: DERIN

Funcion 48 $f_{(x)} = x^5 + x^2 - 2$

Ubica un máximo en (-0,7368;-1,6743) , un mínimo en (0;-2) e inflexión de primera especie en (-0,4642; -1,8061) como se muestra en la figura 193

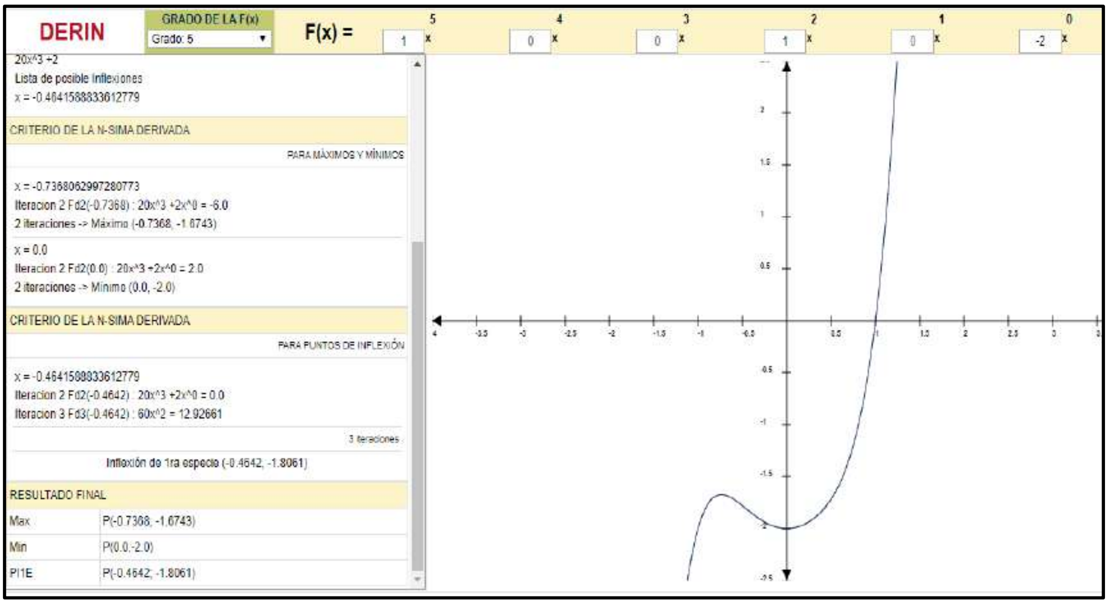


Figura N° 193 Función polinómica 34
Fuente: DERIN

Funcion 49 $f_{(x)} = x^5 - 3x + 1$

Ubica un mínimo en (0,8801;-1,1123), un máximo en (-0,8801;3,1123) así como un punto de inflexión en (0;1) como muestra la figura 194

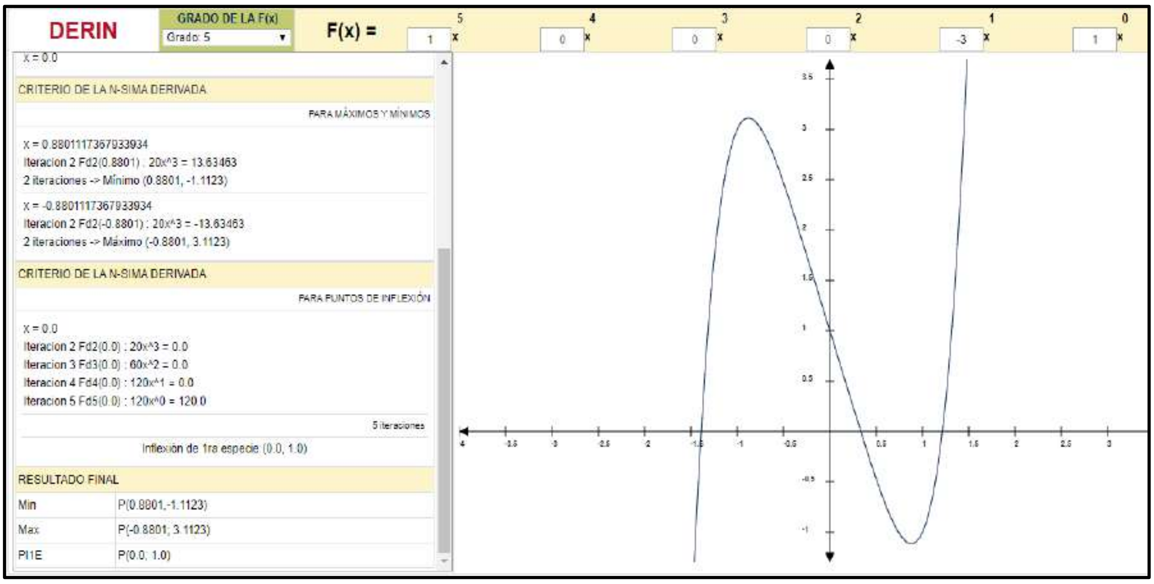


Figura N° 194 Función polinómica 35
Fuente: DERIN

Funcion 50 $f_{(x)} = x^5 - x^4 + 2x^3 - 4x^2$

Ubica un mínimo en (1,0723;-2,0378) , un máximo en (0;0) asì como una inflexión de primera especie en (0,6396; -1,1735) como muesra la figura 195

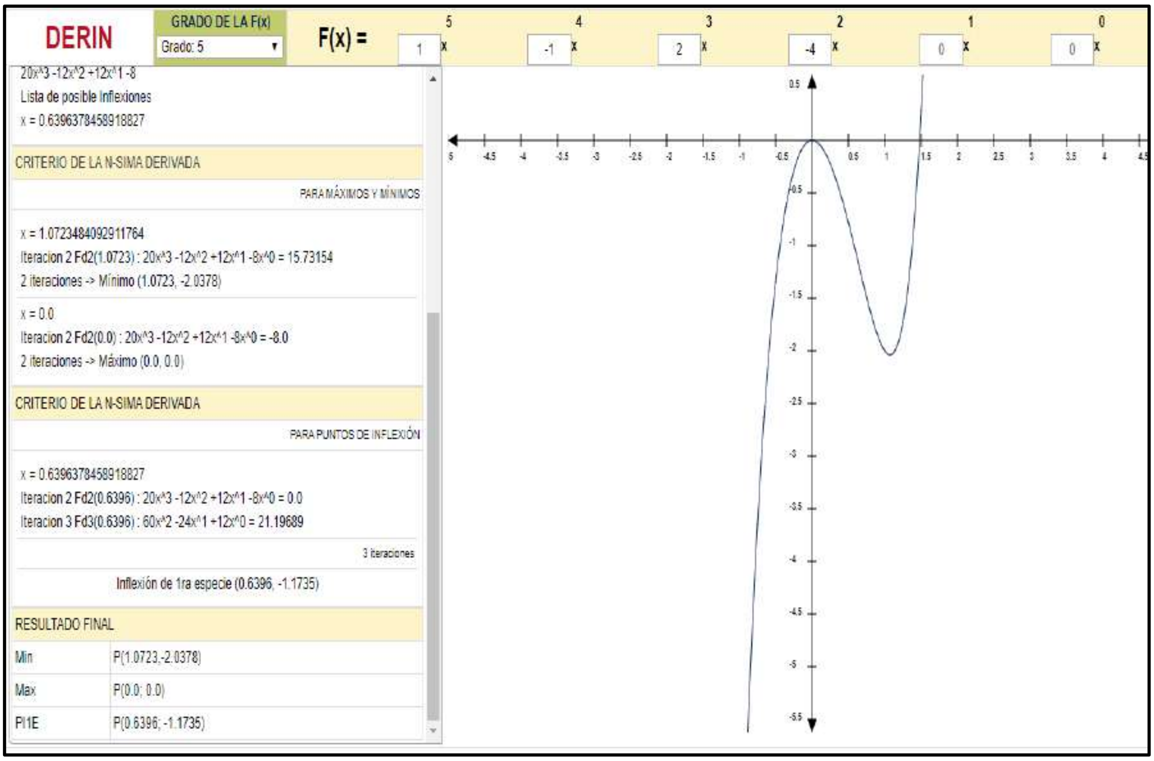


Figura N° 195 Función polinómica 36
Fuente: DERIN

Funcion 51 $f_{(x)} = x^5 - x^4 + 0.5x^2 - 4x$

Ubica un mínimo en (1,1619;-3,6775) ,un máximo en (-0,839;2,7967) y una inflexiòn de primera especie en (-0,2435; 0,9992) como muestra la figura 196

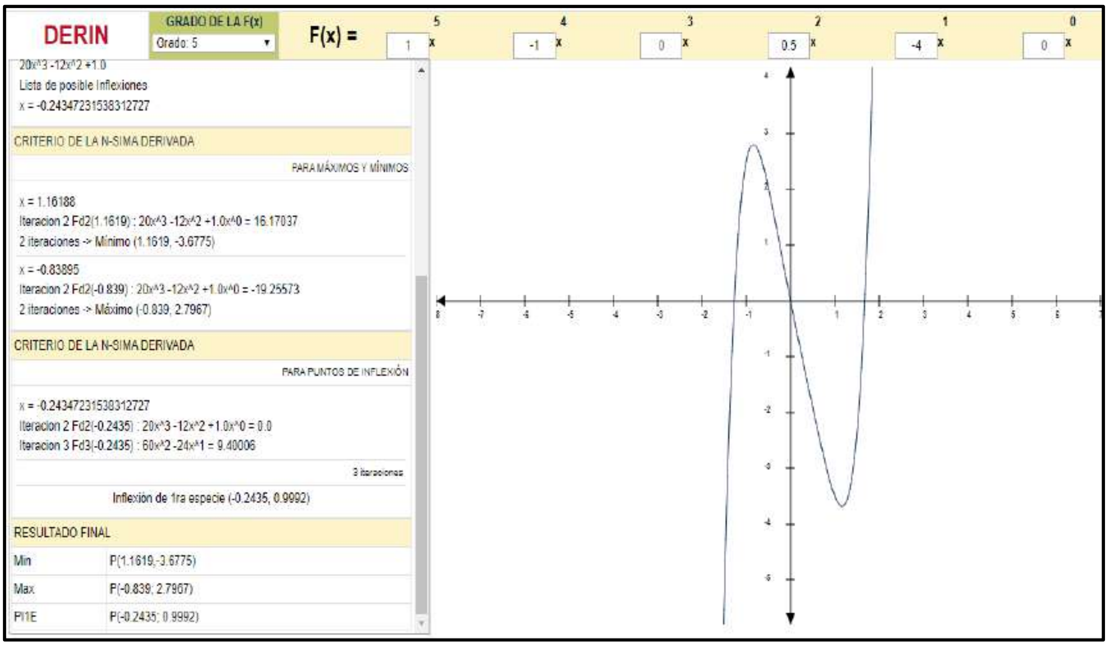


Figura N° 196 Función polinómica 37
Fuente: DERIN

Funcion 52 $f_{(x)} = x^5 - x^4 - 2x^3 - x$

Ubica un mínimo en (1,5992; -5,8598), un máximo en (-0,8736; 1,1158) e inflexiones de primera especie en (1,1307; -3,808) y (-0,5307; 0,7082) y una de segunda especie en (0;0) tal como muestran las figuras 197 y 198

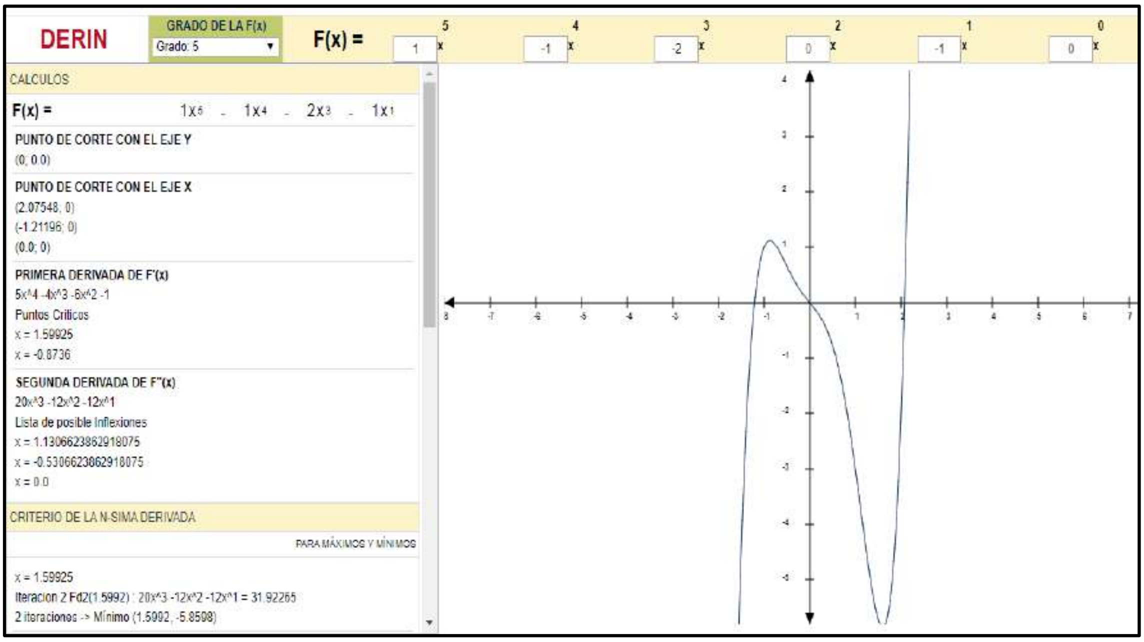


Figura N° 197 Función polinómica 38A
Fuente: DERIN


CRITERIO DE LA N-SIMA DERIVADA

PARA MÁXIMOS Y MÍNIMOS

x = 1.59925
Iteracion 2 Fd2(1.5992) : 20x^3 -12x^2 -12x^1 = 31.92265
2 iteraciones -> Mínimo (1.5992, -5.8598)

x = -0.8736
Iteracion 2 Fd2(-0.8736) : 20x^3 -12x^2 -12x^1 = -12.00915
2 iteraciones -> Máximo (-0.8736, 1.1158)

PARA PUNTOS DE INFLEXIÓN

x = 1.1306623862918075
Iteracion 2 Fd2(1.1307) : 20x^3 -12x^2 -12x^1 = 0.0
Iteracion 3 Fd3(1.1307) : 60x^2 -24x^1 -12x^0 = 37.56795

3 iteraciones

Inflexión de 1ra especie (1.1307, -3.808)

x = -0.5306623862918075
Iteracion 2 Fd2(-0.5307) : 20x^3 -12x^2 -12x^1 = 0.0
Iteracion 3 Fd3(-0.5307) : 60x^2 -24x^1 -12x^0 = 17.63205

3 iteraciones

Inflexión de 1ra especie (-0.5307, 0.7082)

x = 0.0
Iteracion 2 Fd2(0.0) : 20x^3 -12x^2 -12x^1 = 0.0
Iteracion 3 Fd3(0.0) : 60x^2 -24x^1 -12x^0 = -12.0

3 iteraciones

Inflexión de 2da especie (0.0, 0.0)

RESULTADO FINAL

Min	P(1.5992,-5.8598)
Max	P(-0.8736; 1.1158)
PI1E	P(1.1307; -3.808)
PI1E	P(-0.5307; 0.7082)
PI2E	P(0.0; 0.0)

Figura N° 198 Función polinómica 38B
Fuente: DERIN

Funcion 53 $f_{(x)} = 0.2x^5 - 0.25x^4 - x^2 - 2x$

Ubica un mínimo en (1,8737; -5,7207), un máximo en (-0,7035; 0,8164) y una inflexión en (1,1369; -3,604) al como muestran las figuras 199 y 200

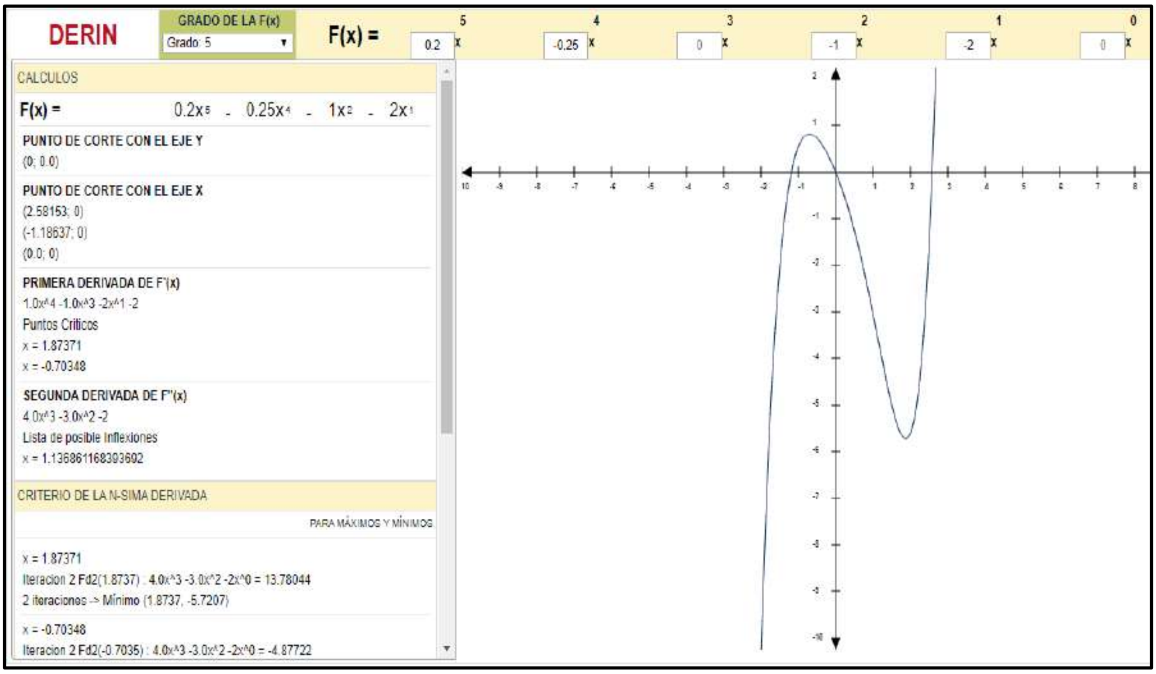


Figura N° 199 Función polinómica 39 A
Fuente: DERIN

CRITERIO DE LA N-SIMA DERIVADA

PARA MÁXIMOS Y MÍNIMOS

x = 1.87371
Iteracion 2 Fd2(1.8737) : 4.0x^3 -3.0x^2 -2x^0 = 13.78044
2 iteraciones -> Mínimo (1.8737, -5.7207)

x = -0.70348
Iteracion 2 Fd2(-0.7035) : 4.0x^3 -3.0x^2 -2x^0 = -4.87722
2 iteraciones -> Máximo (-0.7035, 0.8164)

CRITERIO DE LA N-SIMA DERIVADA

PARA PUNTOS DE INFLEXIÓN

x = 1.136861168393692
Iteracion 2 Fd2(1.1369) : 4.0x^3 -3.0x^2 -2x^0 = 0.0
Iteracion 3 Fd3(1.1369) : 12.0x^2 -6.0x^1 = 8.68827

3 iteraciones

Inflexión de 1ra especie (1.1369, -3.604)

RESULTADO FINAL

Min	P(1.8737, -5.7207)
Max	P(-0.7035; 0.8164)
PI1E	P(1.1369; -3.604)

Figura N° 200 Función polinómica 39 B
Fuente: DERIN

Funcion 54 $f_{(x)} = 0.4x^5 - 2x^3 - 2x^2 - x$

Ubica mínimo en (2,0269; -13,2136), máximo en (-1,3351, 0,8329) e inflexión de primera especie en (-1; 0,6) y (1,366; -8,2935) y de segunda especie en (-0,366; 0,1935) como muestran las figuras 201 y 202.

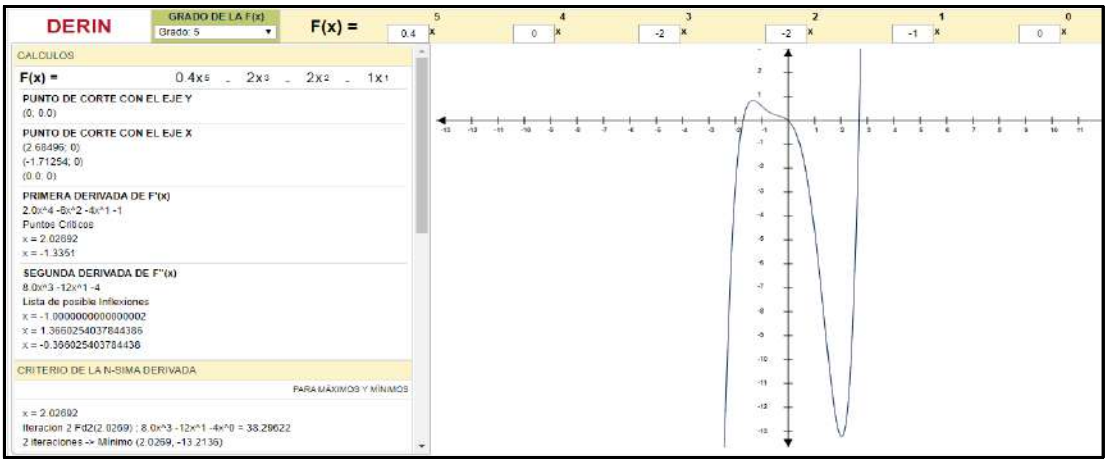


Figura N° 201 Función polinómica 40A
Fuente: DERIN

CRITERIO DE LA N-SIMA DERIVADA

PARA MÁXIMOS Y MÍNIMOS

x = 2.02692
Iteracion 2 Fd2(2.0269) : 8.0x^3 -12x^1 -4x^0 = 38.29622
2 iteraciones -> Mínimo (2.0269, -13.2136)

x = -1.3351
Iteracion 2 Fd2(-1.3351) : 8.0x^3 -12x^1 -4x^0 = -7.01724
2 iteraciones -> Máximo (-1.3351, 0.8329)

PARA PUNTOS DE INFLEXIÓN

x = -1.0000000000000002
Iteracion 2 Fd2(-1.0) : 8.0x^3 -12x^1 -4x^0 = 0.0
Iteracion 3 Fd3(-1.0) : 24.0x^2 -12x^0 = 12.0

3 iteraciones

Inflexión de 1ra especie (-1.0, 0.6)

x = 1.3660254037844386
Iteracion 2 Fd2(1.366) : 8.0x^3 -12x^1 -4x^0 = 0.0
Iteracion 3 Fd3(1.366) : 24.0x^2 -12x^0 = 32.78461

3 iteraciones

Inflexión de 1ra especie (1.366, -8.2935)

x = -0.3660254037844438
Iteracion 2 Fd2(-0.366) : 8.0x^3 -12x^1 -4x^0 = 0.0
Iteracion 3 Fd3(-0.366) : 24.0x^2 -12x^0 = -8.78461

3 iteraciones

Inflexión de 2da especie (-0.366, 0.1935)

RESULTADO FINAL

Min	P(2.0269,-13.2136)
Max	P(-1.3351; 0.8329)
PI1E	P(-1.0; 0.6)
PI1E	P(1.366; -8.2935)
PI2E	P(-0.366; 0.1935)

Figura N° 202 Función polinómica 40B
Fuente: DERIN

Funcion 55 $f_{(x)} = f_{(x)} = \mathbf{0.2x^5 - 0.25x^4 - x^3 - 0.5x^2 - 1}$

Ubica máximos en (-1; -0,95) y (0; -1), mínimos en (2,414; -10,075) y (-0,414;-1,024) e inflexiones de primera especie en (-0,7726; -0,981) y (1,7117; -6,687) y de segunda especie en (-0,189; -1,0115) lo que se evidencia en las figuras 203 y 204

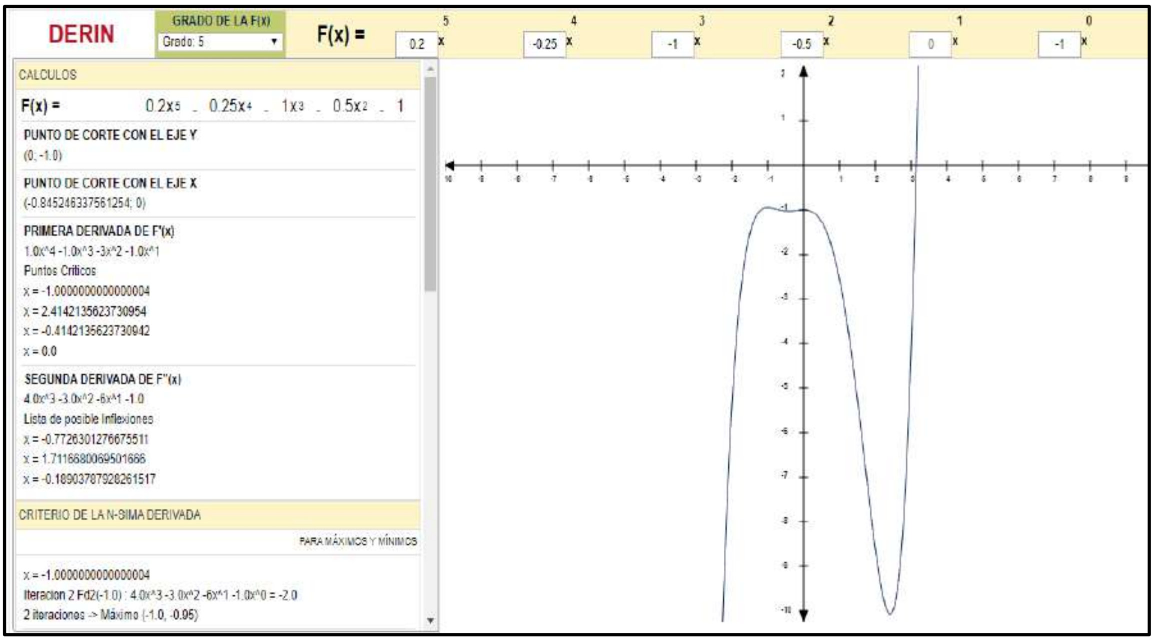


Figura N°203 Función polinómica 41A
Fuente: DERIN

CRITERIO DE LA N-SIMA DERIVADA

PARA MÁXIMOS Y MÍNIMOS

x = -1.0000000000000004
Iteracion 2 Fd2(-1.0) : 4.0x^3 -3.0x^2 -6x^1 -1.0x^0 = -2.0
2 iteraciones -> Máximo (-1.0, -0.95)

x = 2.4142135623730954
Iteracion 2 Fd2(2.4142) : 4.0x^3 -3.0x^2 -6x^1 -1.0x^0 = 23.31371
2 iteraciones -> Mínimo (2.4142, -10.0755)

x = -0.4142135623730942
Iteracion 2 Fd2(-0.4142) : 4.0x^3 -3.0x^2 -6x^1 -1.0x^0 = 0.68629
2 iteraciones -> Mínimo (-0.4142, -1.0245)

x = 0.0
Iteracion 2 Fd2(0.0) : 4.0x^3 -3.0x^2 -6x^1 -1.0x^0 = -1.0
2 iteraciones -> Máximo (0.0, -1.0)

CRITERIO DE LA N-SIMA DERIVADA

PARA PUNTOS DE INFLEXIÓN

x = -0.7726301276675511
Iteracion 2 Fd2(-0.7726) : 4.0x^3 -3.0x^2 -6x^1 -1.0x^0 = 0.0
Iteracion 3 Fd3(-0.7726) : 12.0x^2 -6.0x^1 -6x^0 = 5.79927

3 iteraciones

Inflexión de 1ra especie (-0.7726, -0.9814)

x = 1.7116680069501666
Iteracion 2 Fd2(1.7117) : 4.0x^3 -3.0x^2 -6x^1 -1.0x^0 = 0.0
Iteracion 3 Fd3(1.7117) : 12.0x^2 -6.0x^1 -6x^0 = 18.88768

3 iteraciones

Inflexión de 1ra especie (1.7117, -6.6872)

x = -0.18903787928261517
Iteracion 2 Fd2(-0.189) : 4.0x^3 -3.0x^2 -6x^1 -1.0x^0 = 0.0
Iteracion 3 Fd3(-0.189) : 12.0x^2 -6.0x^1 -6x^0 = -4.43695

3 iteraciones

Inflexión de 2da especie (-0.189, -1.0115)

RESULTADO FINAL

Max	P(-1.0; -0.95)
Min	P(2.4142,-10.0755)
Min	P(-0.4142,-1.0245)
Max	P(0.0; -1.0)
PI1E	P(-0.7726; -0.9814)
PI1E	P(1.7117; -6.6872)
PI2E	P(-0.189; -1.0115)

Figura N° 204 Función polinómica 41B
Fuente: DERIN

Funcion 56 $f_{(x)} = \mathbf{0.2x^5 - x^3 - 0.5x^2 + 2x - 1}$

Ubica un mínimo en (1,702; -1,118), un máximo en (0,718;-0,1537) e inflexiones de primera especie en (-1,1309; -2,8249) y (1,301; -0,701) y una ide segunda en (-0,1699; -1,3494) como muestran las figuras 205 y 206

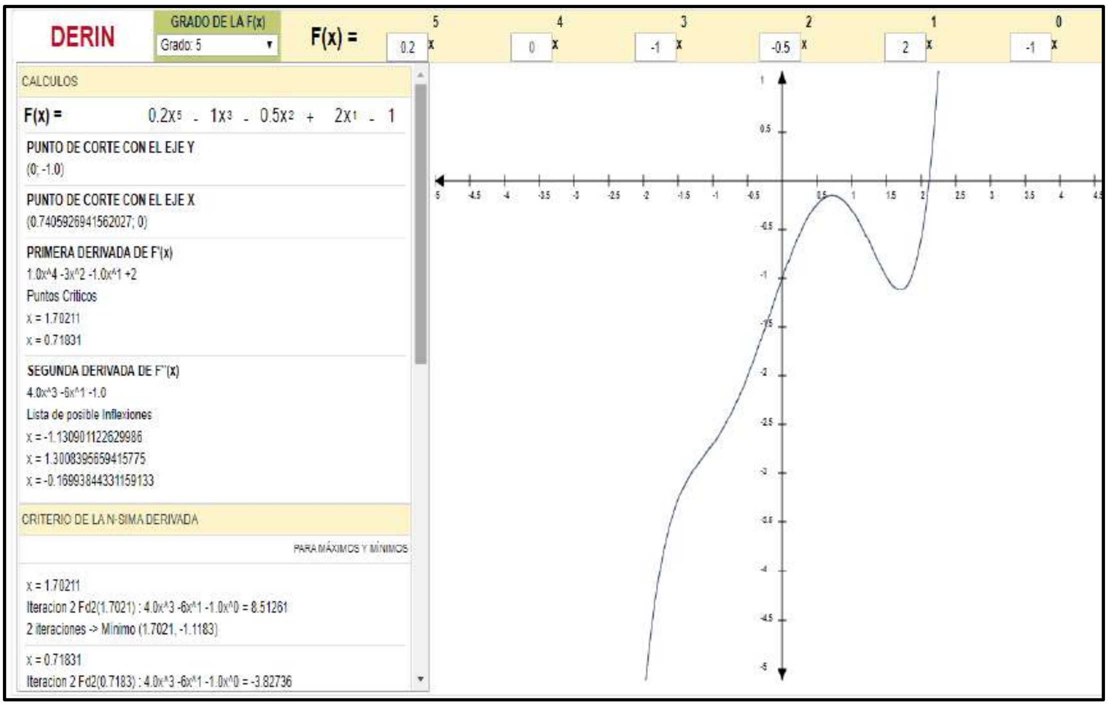


Figura N° 204 Función polinómica 42A
Fuente: DERIN

CRITERIO DE LA N-SIMA DERIVADA

PARA MÁXIMOS Y MÍNIMOS

x = 1.70211
Iteracion 2 Fd2(1.7021) : 4.0x^3 -6x^1 -1.0x^0 = 8.51261
2 iteraciones -> Mínimo (1.7021, -1.1183)

x = 0.71831
Iteracion 2 Fd2(0.7183) : 4.0x^3 -6x^1 -1.0x^0 = -3.82736
2 iteraciones -> Máximo (0.7183, -0.1537)

PARA PUNTOS DE INFLEXIÓN

x = -1.130901122629986
Iteracion 2 Fd2(-1.1309) : 4.0x^3 -6x^1 -1.0x^0 = 0.0
Iteracion 3 Fd3(-1.1309) : 12.0x^2 -6x^0 = 9.34725

3 iteraciones

Inflexión de 1ra especie (-1.1309, -2.8249)

x = 1.3008395659415775
Iteracion 2 Fd2(1.3008) : 4.0x^3 -6x^1 -1.0x^0 = 0.0
Iteracion 3 Fd3(1.3008) : 12.0x^2 -6x^0 = 14.3062

3 iteraciones

Inflexión de 1ra especie (1.3008, -0.7007)

x = -0.16993844331159133
Iteracion 2 Fd2(-0.1699) : 4.0x^3 -6x^1 -1.0x^0 = 0.0
Iteracion 3 Fd3(-0.1699) : 12.0x^2 -6x^0 = -5.65345

3 iteraciones

Inflexión de 2da especie (-0.1699, -1.3494)

RESULTADO FINAL

Min	P(1.7021,-1.1183)
Max	P(0.7183; -0.1537)
PI1E	P(-1.1309; -2.8249)
PI1E	P(1.3008; -0.7007)
PI2E	P(-0.1699; -1.3494)

Figura N° 206 Función polinómica 42B
Fuente: DERIN

Funcion 57 $f_{(x)} = \mathbf{0.2x^5 - 0.5x^4 - x^3 + 0.25x^2 + 1.5x - 1}$

Ubica un mínimo en (2,9108; -13,28) , un máximo en (0,6874; -0,2565) e inflexiones de primera especie en (-0,7435; -1,7643) y (2,1659; -8,2094) y de segunda en (0,776; -0,8825) como muestra en las figuras 207 y 208.

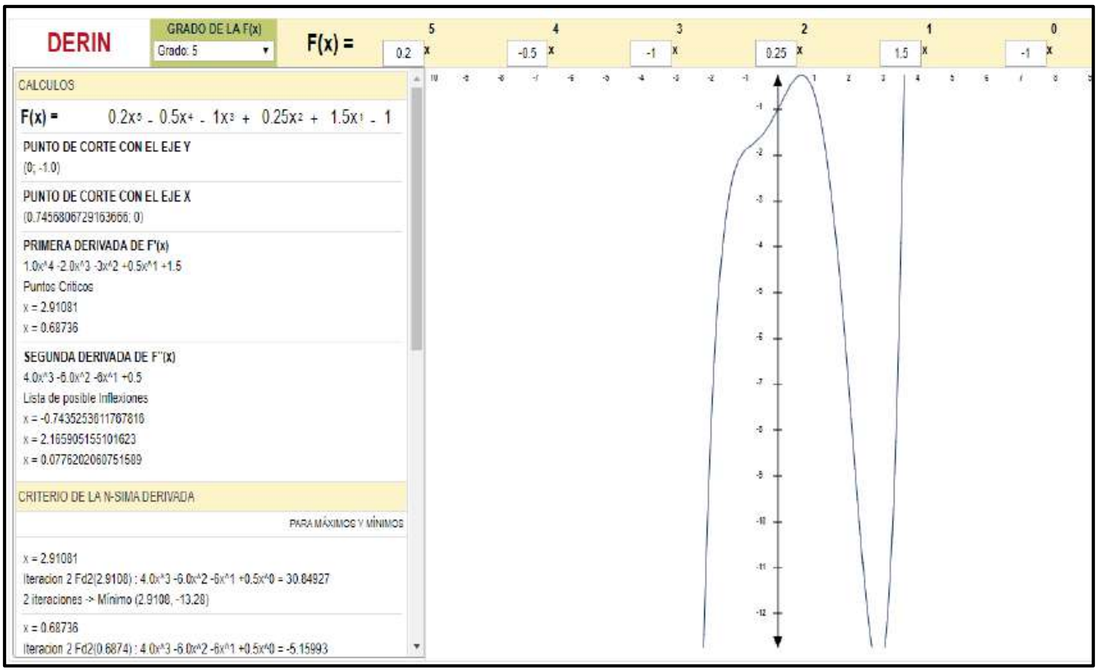


Figura N° 207 Función polinómica 43A
Fuente: DERIN

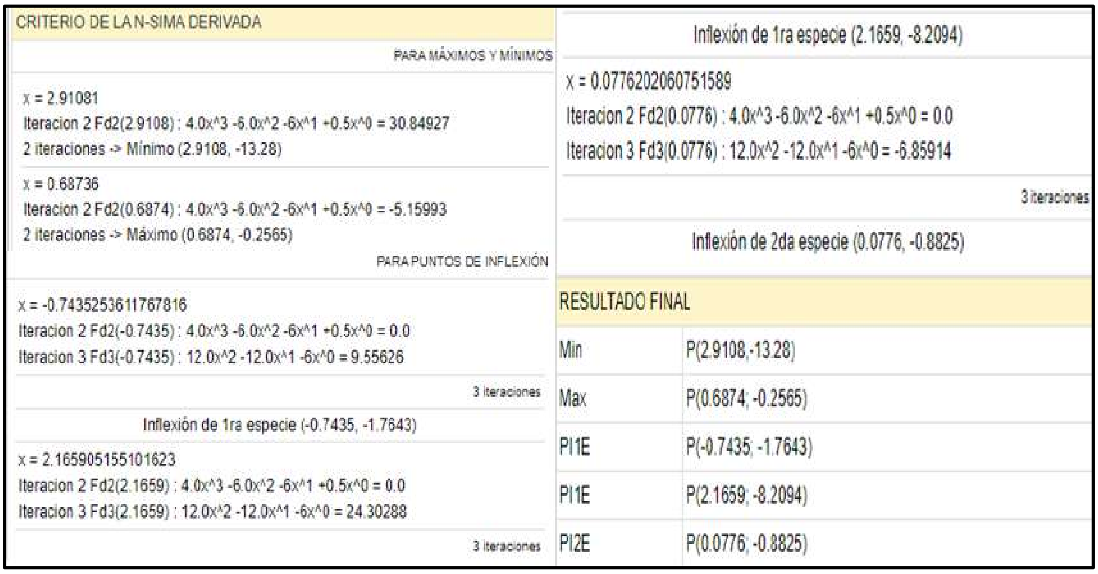


Figura N° 208 Función polinómica 43B
Fuente: DERIN

4.7.5.2 Ejecución con el software WolframAlpha

Funcion 32 $f_{(x)} = 0,01x^5$

No encuentra extremos relativos ni inflexiones como muestra la figura 209

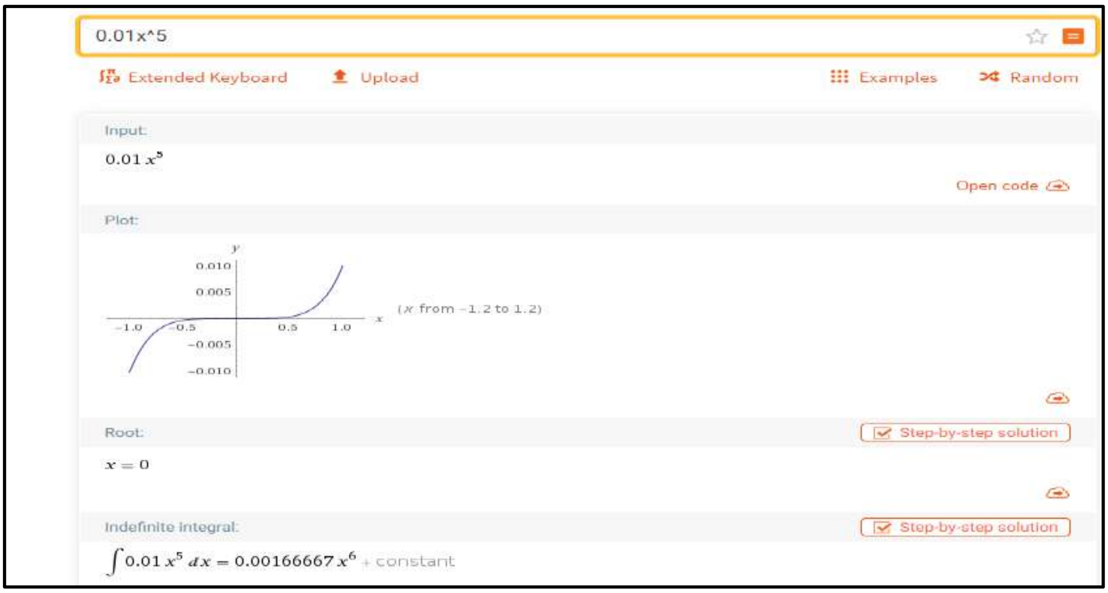


Figura N° 209 Función polinómica 18 Wolfram
Fuente: WolframAlpha

Funcion 33 $f_{(x)} = x^5 - x^4$

Ubica un máximo en x=0 y un mínimo en x = 4/5 como muestra la figura 210

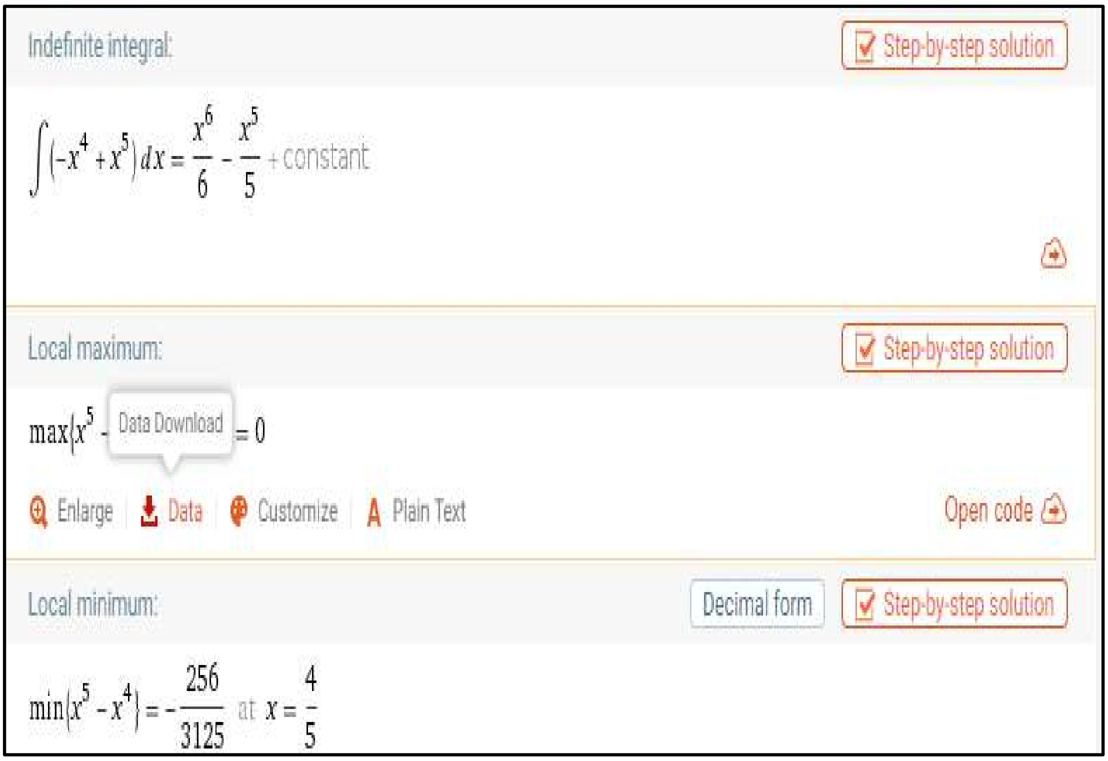


Figura N° 210 Función polinómica 19 Wolfram
Fuente: WolframAlpha

Funcion 34 $f_{(x)} = 3x^5 - 5x^3$

Ubica un máximo en (-1;2) , un mínimo en (1;-2) como muestra la figura 211

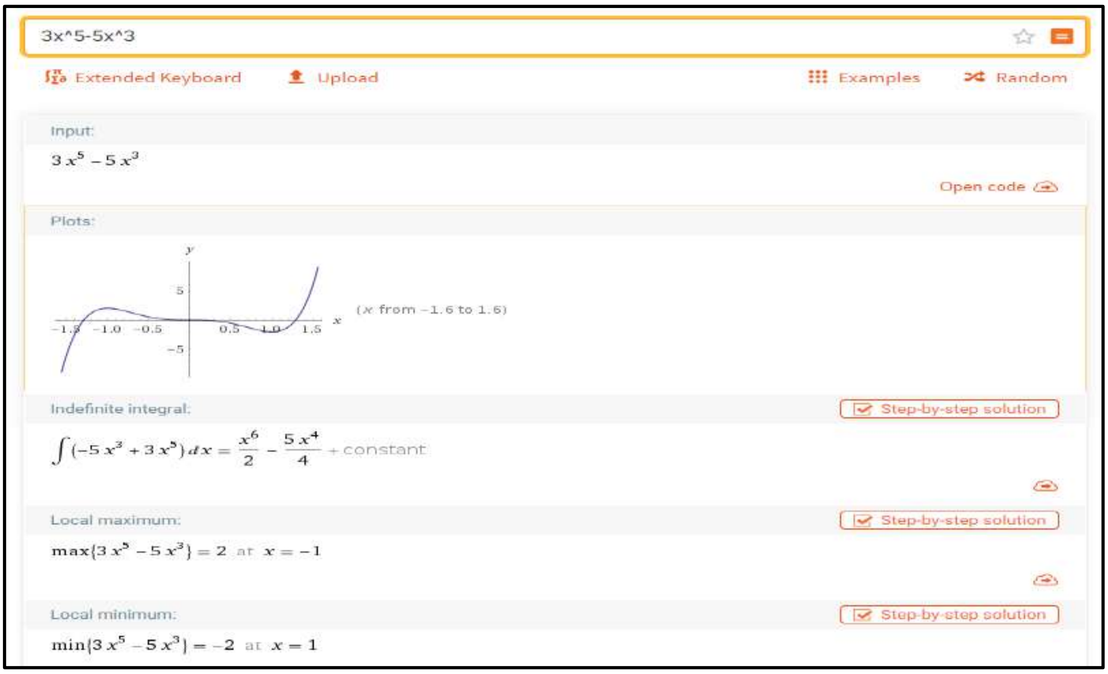


Figura N° 211 Función polinómica 20 Wolfram
Fuente: WolframAlpha

Funcion 35 $f_{(x)} = x^5 - 2x^2$

Ubica un máximo en (0,0) , un mínimo en (0,928;-1,03) como muestra la figura 212

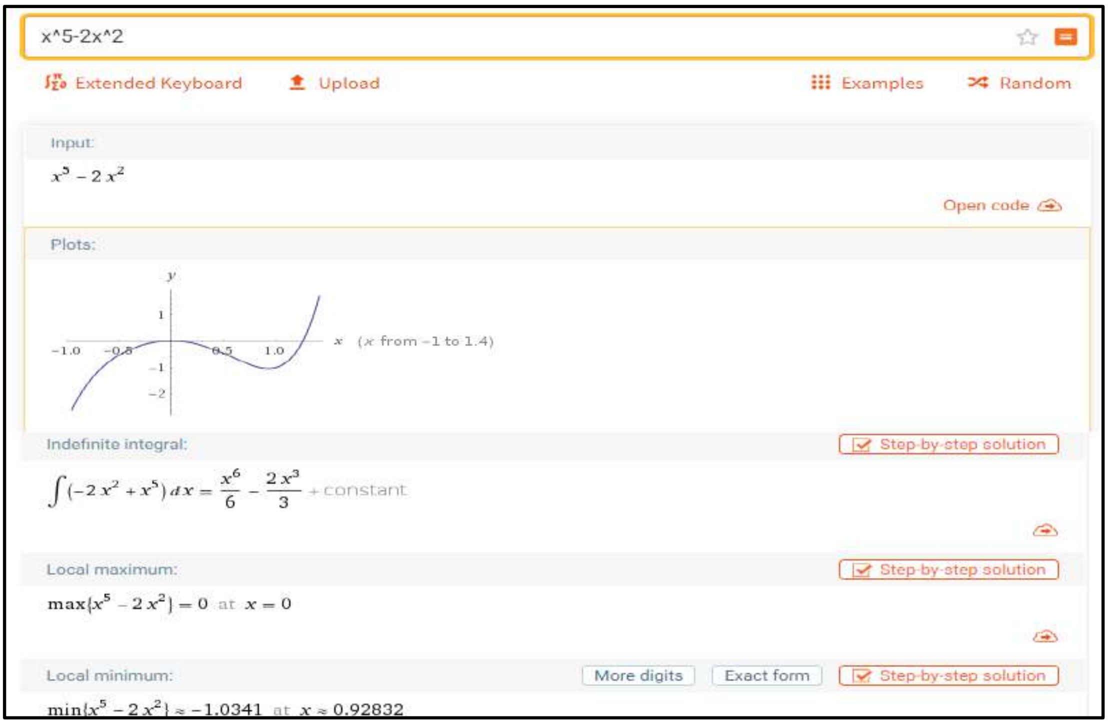


Figura N° 212 Función polinómica 21 Wolfram
Fuente: WolframAlpha

Funcion 36 $f_{(x)} = x^5 - 5x$

Ubica un máximo en (-1;4) y un mínimo en (1;-4) como muestra la figura 213

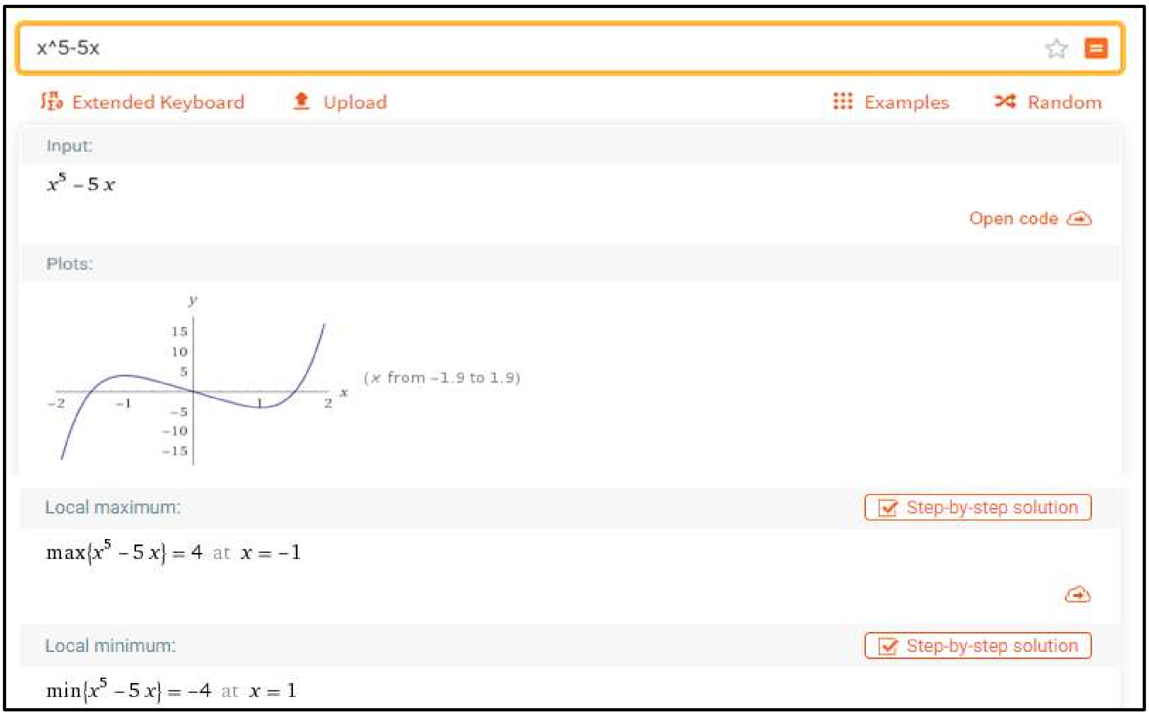


Figura N° 213 Función polinómica 22 Wolfram
Fuente: WolframAlpha

Funcion 37 $f_{(x)} = x^5 + 0,5x$

No ubica extremos relativos como muestra la figura 214

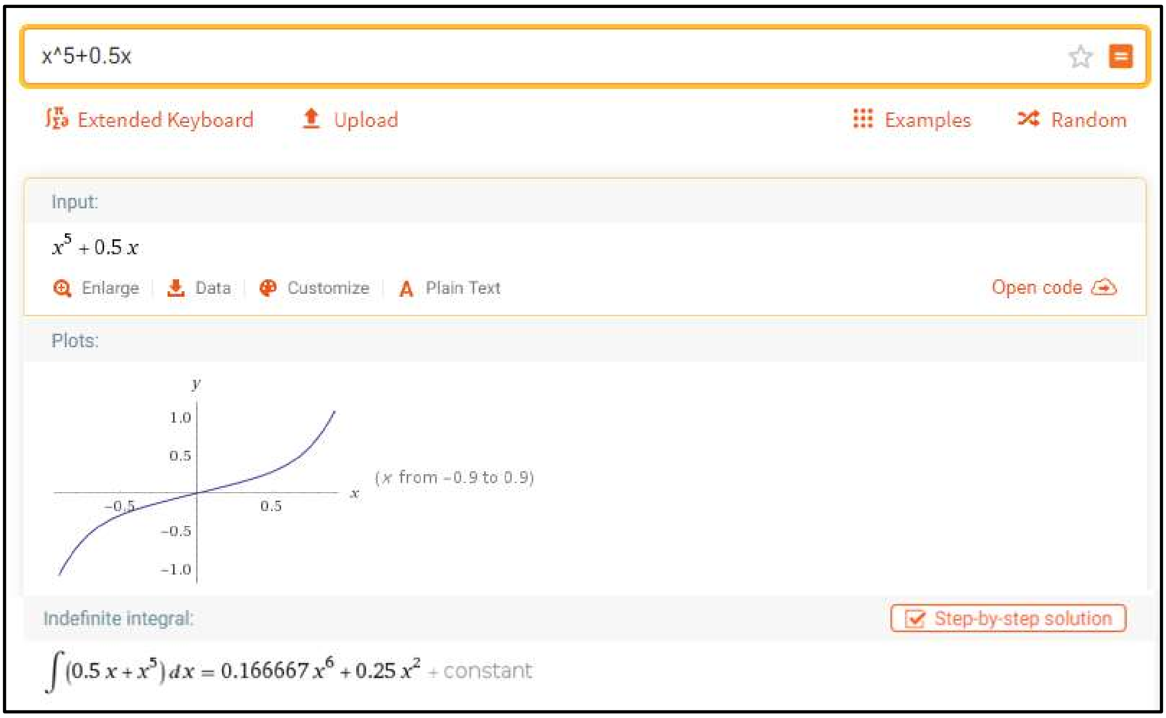


Figura N° 214 Función polinómica 23 Wolfram
Fuente: WolframAlpha

Funcion 38 $f_{(x)} = x^5 - 4x^4 + 4x^3$

Uibica un mínimo en (2;0), un máximo en $\left(\frac{6}{5};\frac{3456}{3125}\right)$ como muestra la figura 215

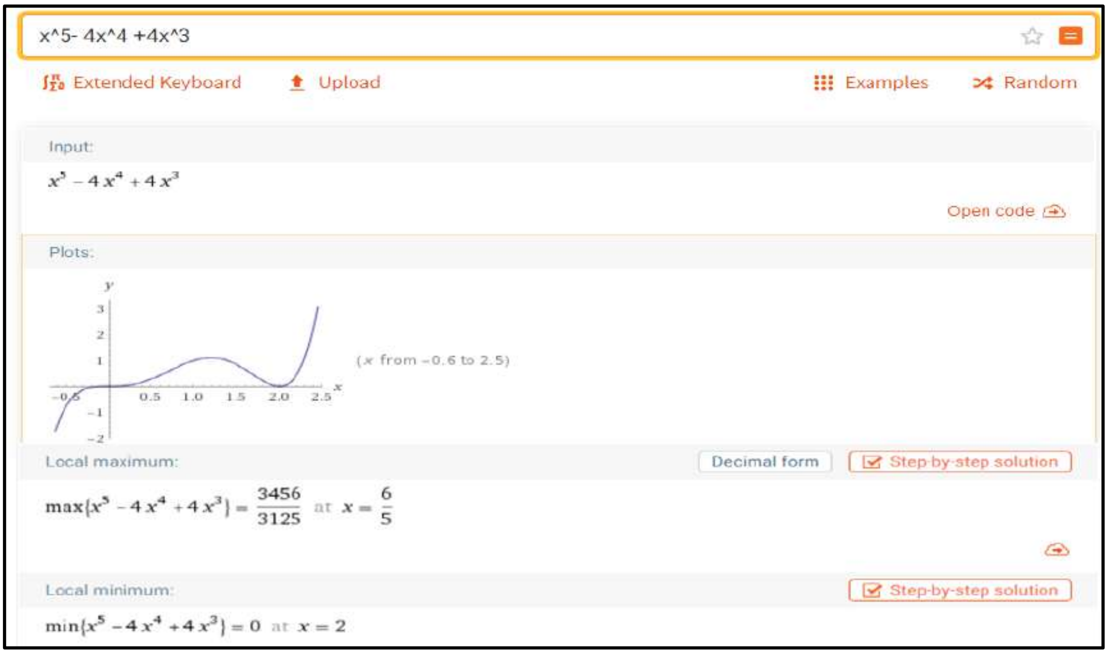


Figura N° 215 Función polinómica 24 Wolfram
Fuente: WolframAlpha

Funcion 39 $f_{(x)} = x^5 - 3x^4 + 3x^2$

Ubica dos máximos en (-0,629;0,618) y en (0,892;1,052) así como dos mínimos en (0;0) y en (2,137;-4,297) como muestra la figura 216

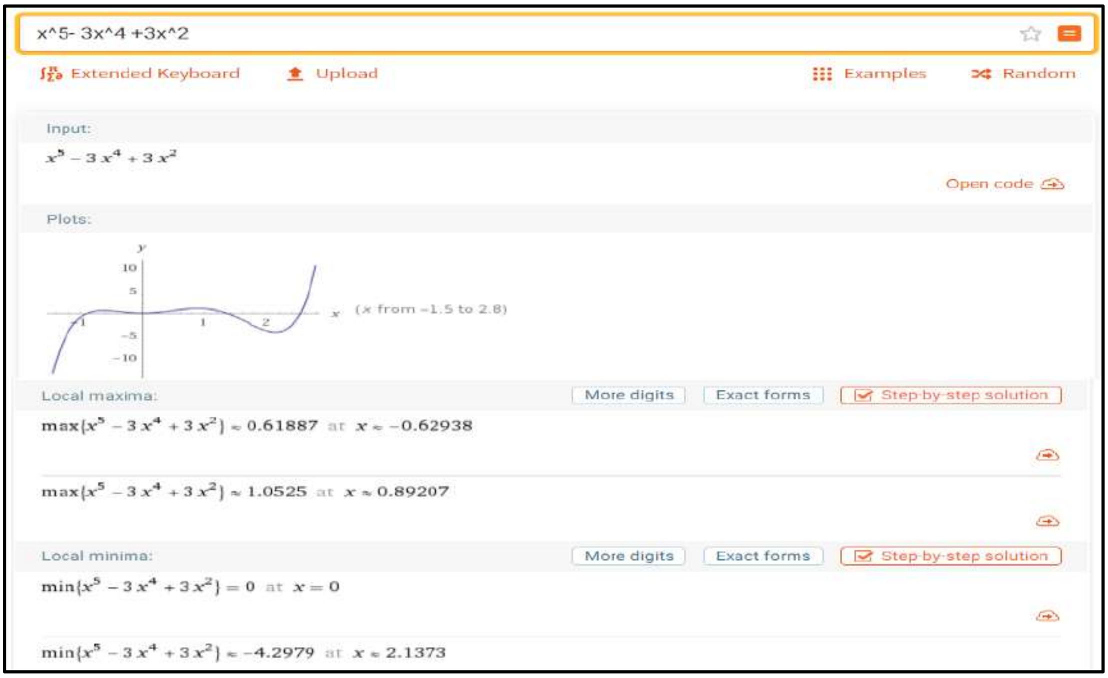


Figura N° 216 Función polinómica 25 Wolfram
Fuente: WolframAlpha

Funcion 40 $f_{(x)} = x^5 - 2x^4 - 2x$

Ubica un mínimo en (1,6838; -5,9093), un máximo en (-0,569;0,868) como se muestra en la figura 217

Figura N° 217 Función polinómica 26 Wolfram
Fuente: WolframAlpha

Funcion 41 $f_{(x)} = 0,2x^5 - 0,5x^4 - 1$

Ubica un mínimo en (2;-2,6) y un máximo en (0;-1) como ilustra la fgura 218

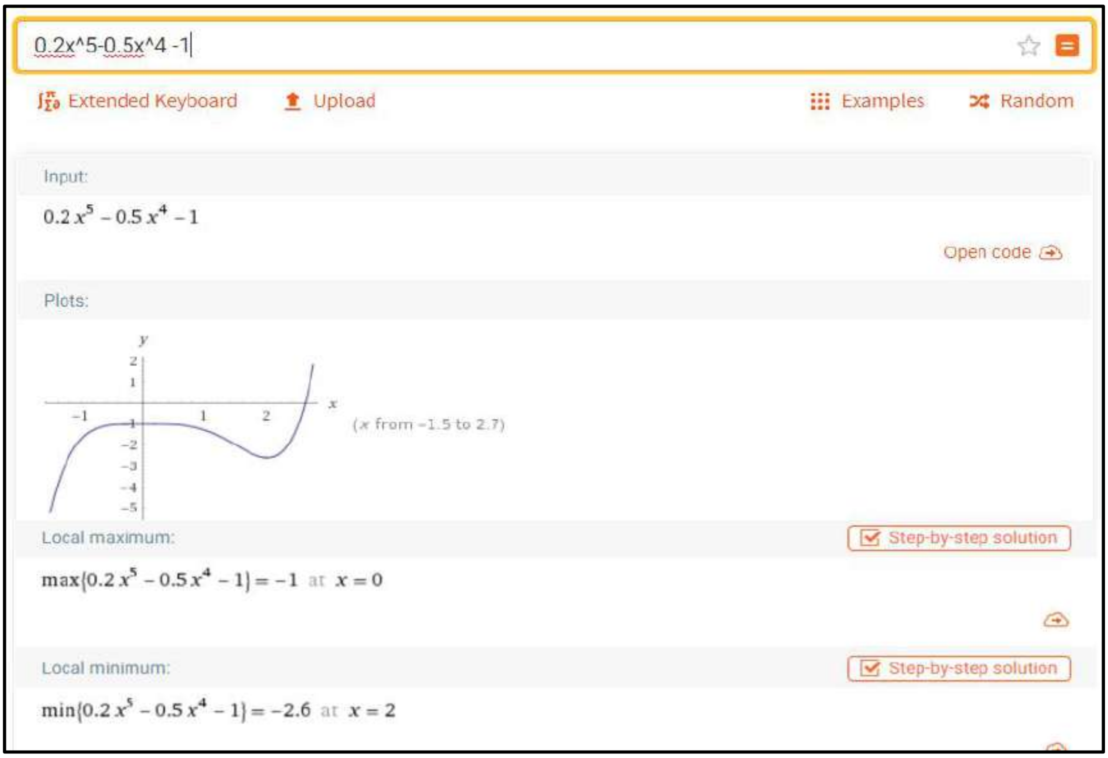


Figura N° 218 Función polinómica 27 Wolfram
Fuente: WolframAlpha

Función 42 $f_{(x)} = x^5 - 2x^3 - 3x^2$

Ubica mínimo en (1,43;-6,005) ,un máximo en (0;0) como muestra la figura 219

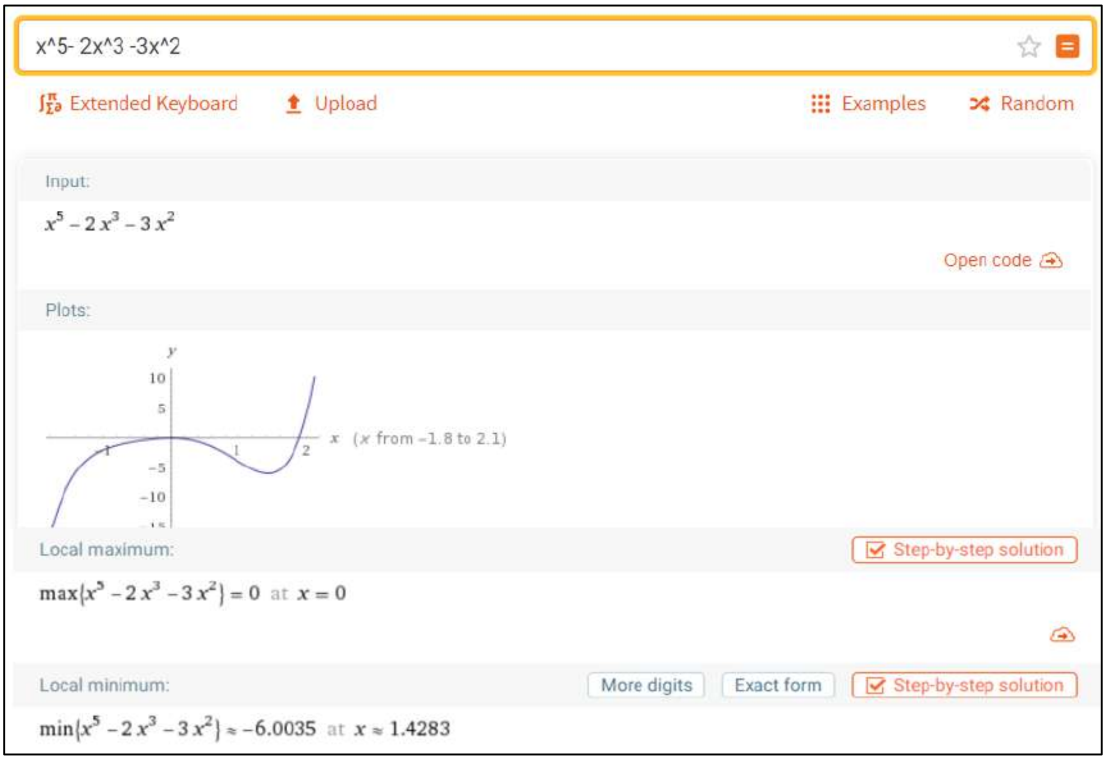


Figura N° 219 Función polinómica 28 Wolfram
Fuente: WolframAlpha

Funcion 43 $f_{(x)} = x^5 - 1.5x^3 + 1.2x^2$

Ubica un mínimo en (0;0), máximo en (-1,15;1,86) como muestra la figura 220

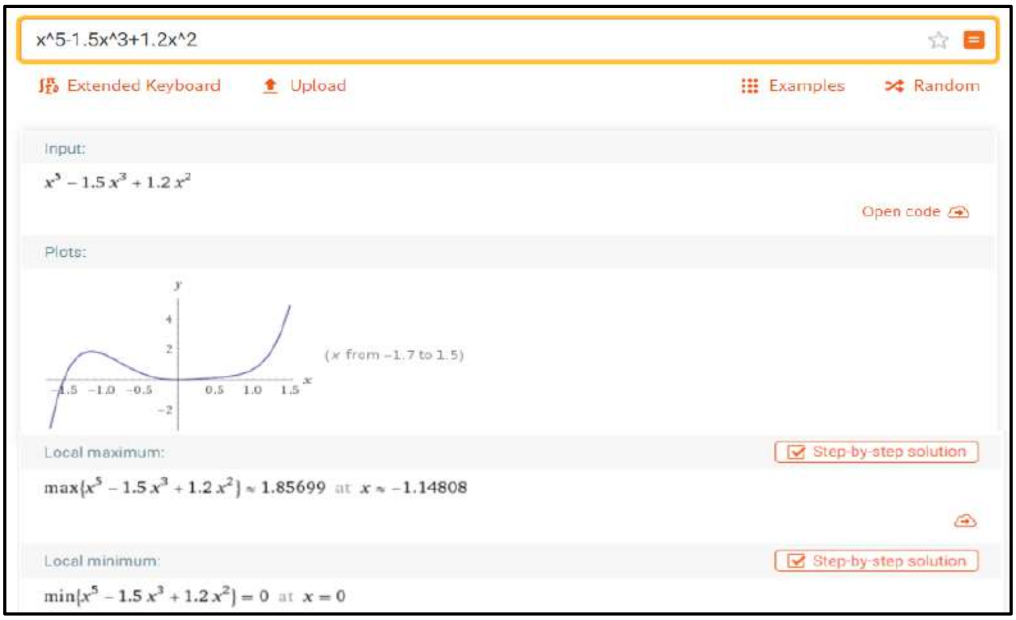


Figura N° 220 Función polinómica 29 Wolfram
Fuente: WolframAlpha

Funcion 44 $f_{(x)} = x^5 - 2x^3 + 2x$

No encuentra puntos extremos como muestra la figura 221

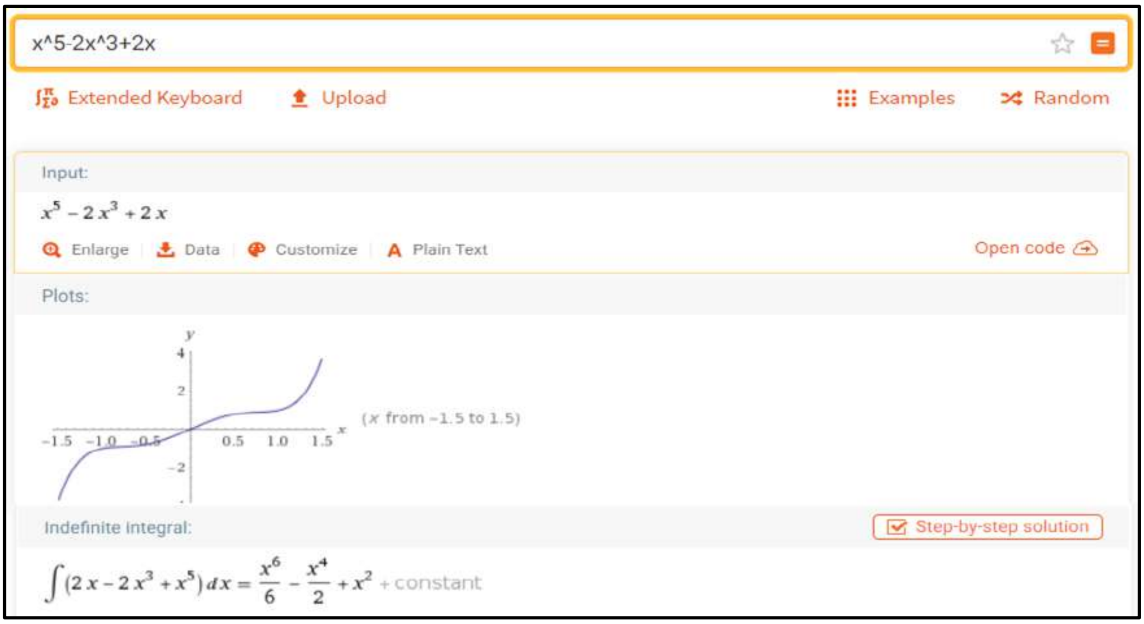


Figura N° 221 Función polinómica 30 Wolfram
Fuente: WolframAlpha

Funcion 45 $f_{(x)} = x^5 - 2x^3 + x$

Ubica dos mínimos en los puntos (-1;0) y (-0,447; -0,286) y dos máximos en (1;0) y (0,447; -0,286) como muestra la figura 222

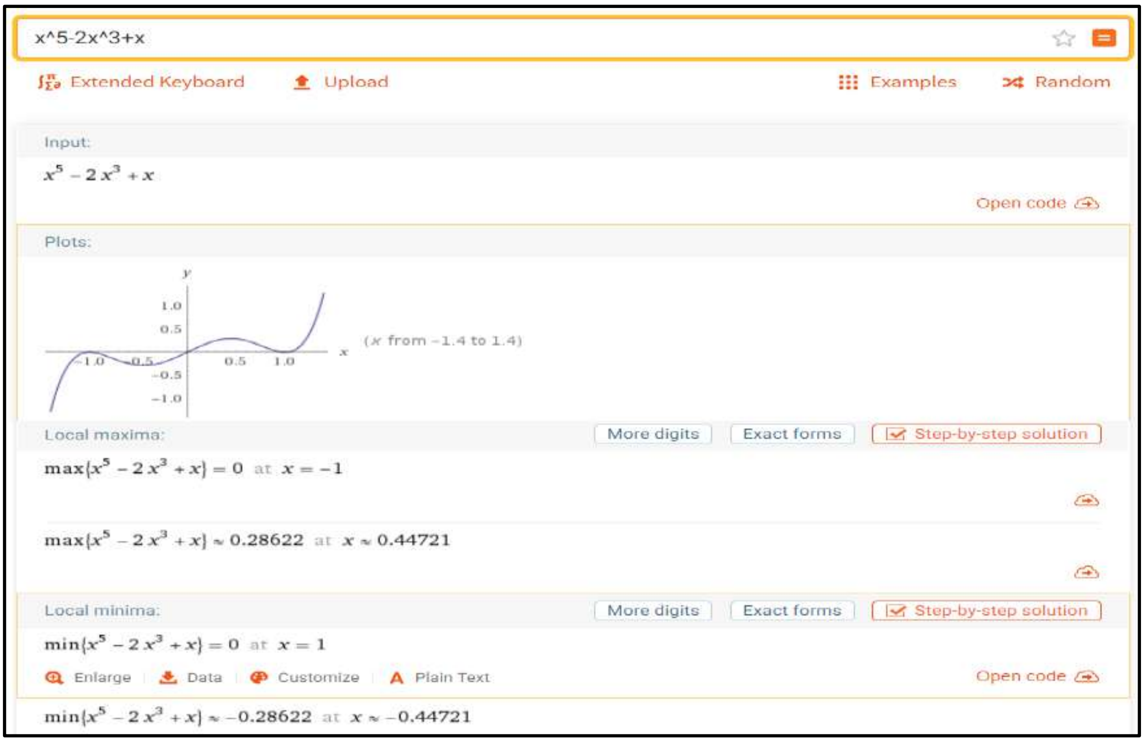


Figura N° 222 Función polinómica 31 Wolfram
Fuente: WolframAlpha

Funcion 46 $f_{(x)} = x^5 - x^3 + 1$

Ubica un máximo en (-0,7746;1,1859) y un mínimo en (0,7746;0,8141) como muestra la figura 223

Figura N° 223 Función polinómica 32 Wolfram
Fuente: WolframAlpha

Funcion 47 $f_{(x)} = x^5 + x^2 - 2x$

Ubica un máximo en (-0,93837;2,0297) y un mínimo en (0,6231;-0,764) como muestra la figura 224

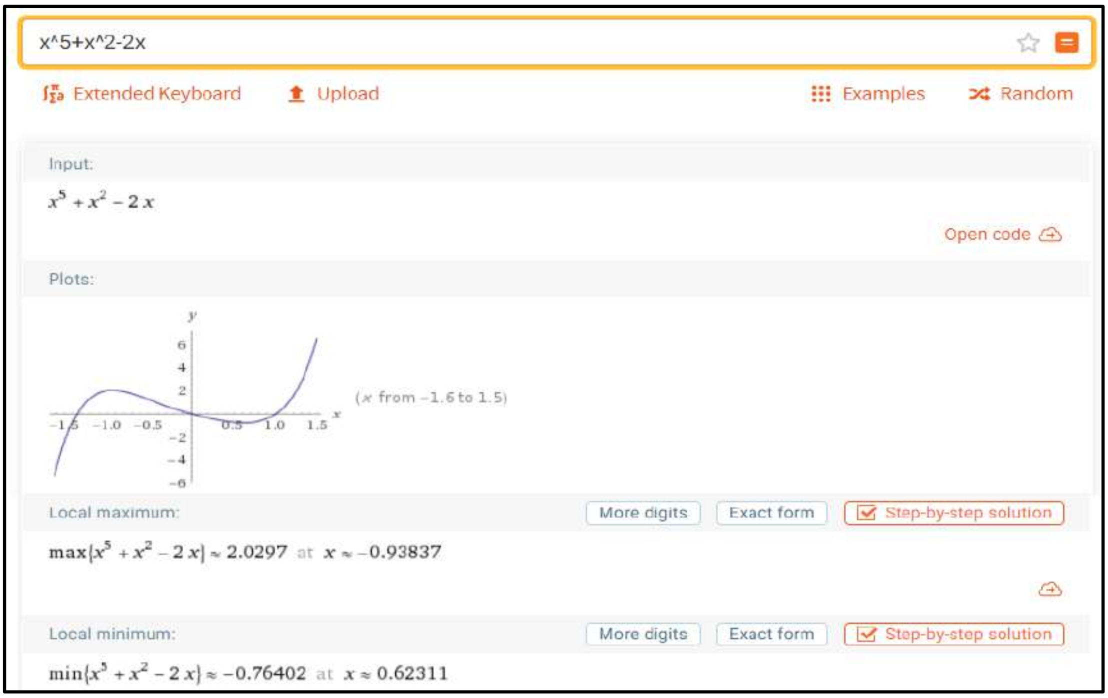


Figura N° 224 Función polinómica 33 Wolfram
Fuente: WolframAlpha

Funcion 48 $f_{(x)} = x^5 + x^2 - 2$

Ubica un mínimo en (0;-2) y un máximo en (-0,73681;-1,6743) como muestra la figura 225

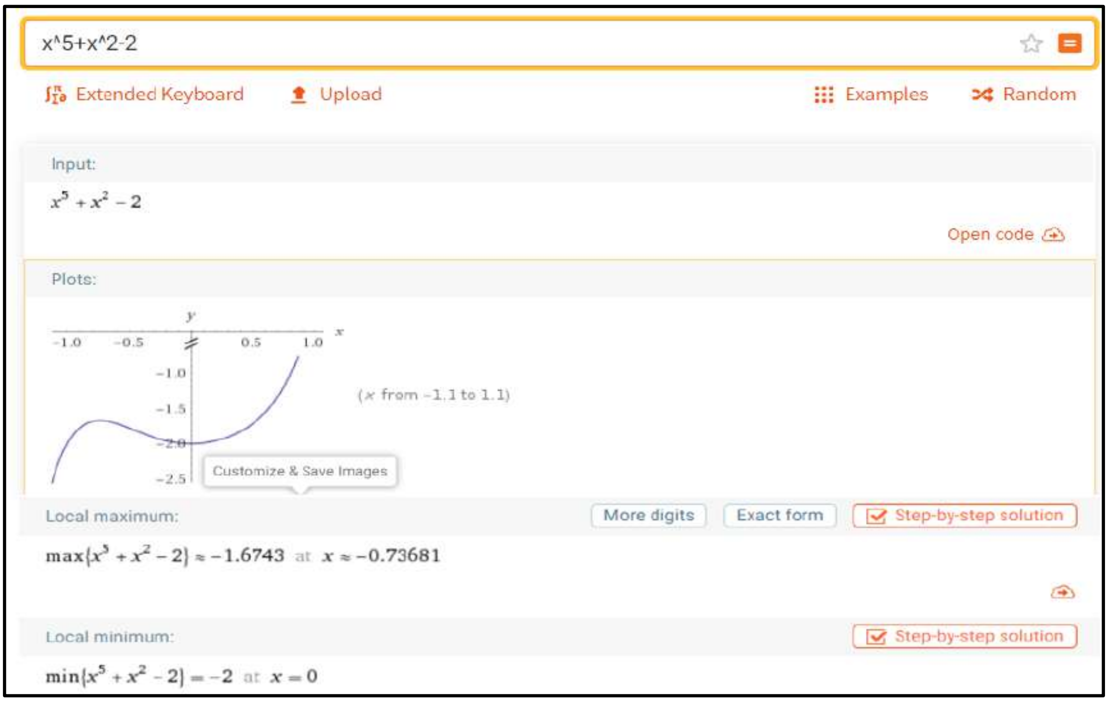


Figura N° 225 Función polinómica 34 Wolfram
Fuente: WolframAlpha

Funcion 49 $f_{(x)} = x^5 - 3x + 1$

Ubica un mínimo en (0,88011;-1,1123), un máximo en (-0,88011;3,1123) como muestra la figura 226

Figura N° 226 Función polinómica 35 Wolfram
Fuente: WolframAlpha

Funcion 50 $f_{(x)} = x^5 - x^4 + 2x^3 - 4x^2$

Ubica un mínimo en (1,0723;-2,0378), un máximo en (0;0) como muestra la figura 227

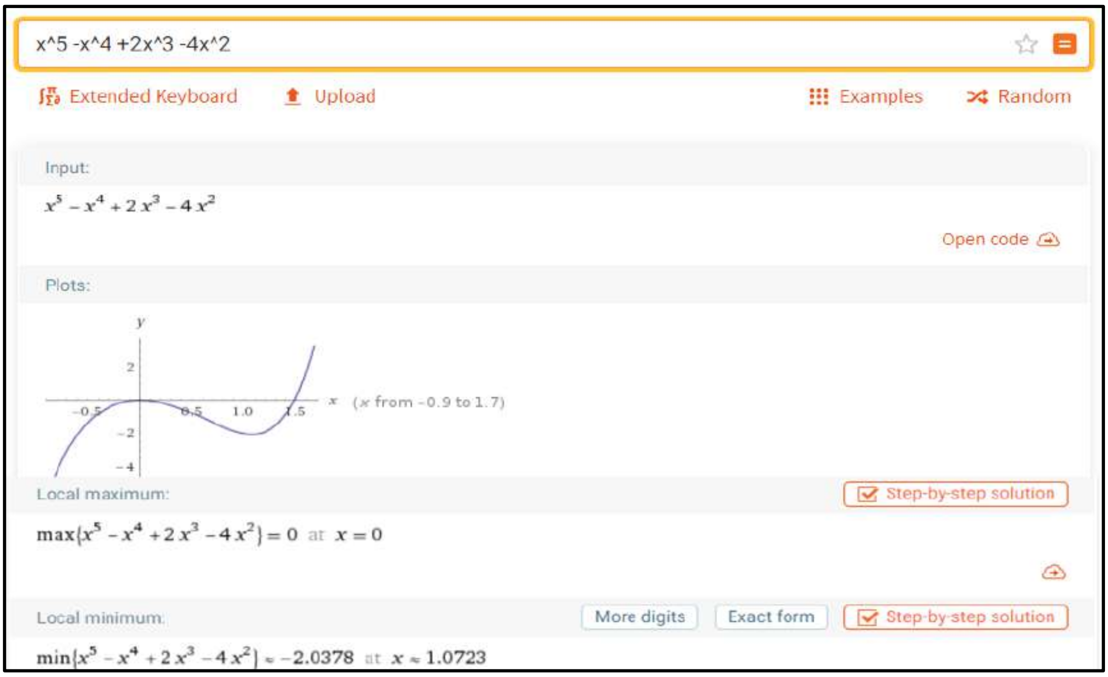


Figura N° 227 Función polinómica 36 Wolfram
Fuente: WolframAlpha

Funcion 51 $f_{(x)} = x^5 - x^4 + 0.5x^2 - 4x$

Ubica un mínimo en (1,16188;-3,67753), un máximo en (-0,8389;2,79673) como muestra la figura 228

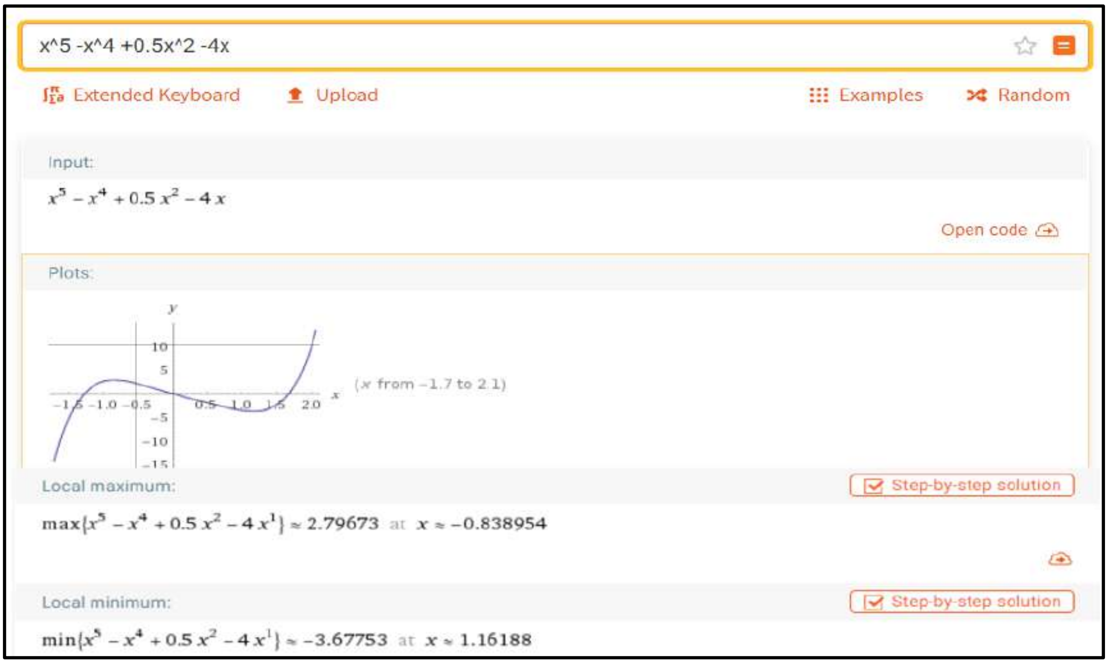


Figura N° 228 Función polinómica 37 Wolfram
Fuente: WolframAlpha

Funcion 52 $f_{(x)} = x^5 - x^4 - 2x^3 - x$

Ubica un máximo en (-0,8736; 1,1158) y un mínimo en (1,5992; -5,8598) como muestra la figura 229

Figura N° 229 Función polinómica 38 Wolfram
Fuente: WolframAlpha

Funcion 53 $f_{(x)} = 0.2x^5 - 0.25x^4 - x^2 - 2x$

Ubica un máximo en (-0,703478; 0,81639) y un mínimo en (1,87371; -5,72068) tal como muestra la figura 230. Cabe señalar que el tiempo **de respuesta de Wolfram a la fecha (14/10/ 2019) es mayor respecto a los demás software**

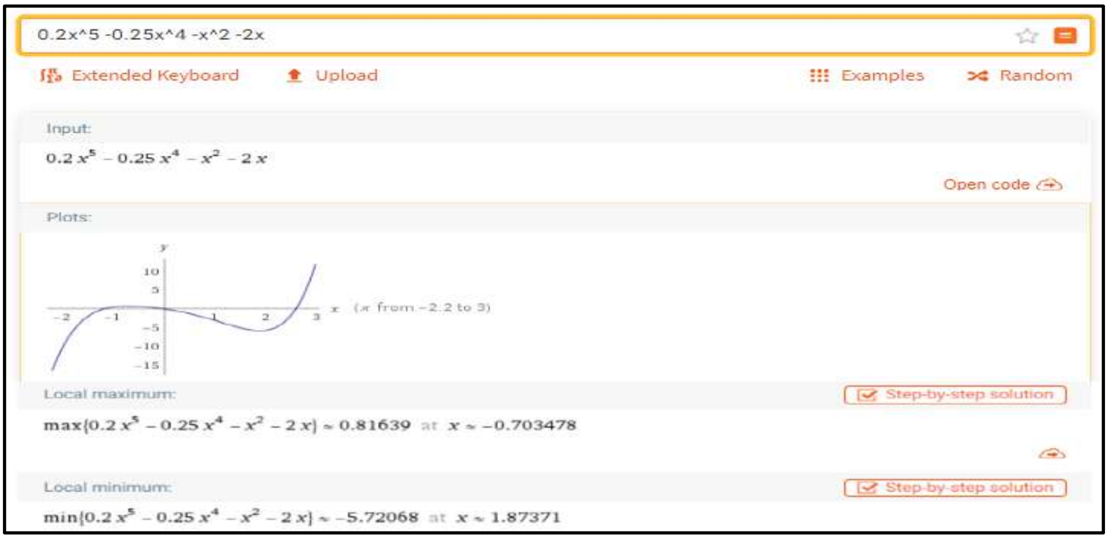


Figura N° 230 Función polinómica 39 Wolfram
Fuente: WolframAlpha

Funcion 54 $f_{(x)} = 0.4x^5 - 2x^3 - 2x^2 - x$

Ubica un máximo en x=-1,3 , un mpinimo en x= 2,02 como muestra la figura 231

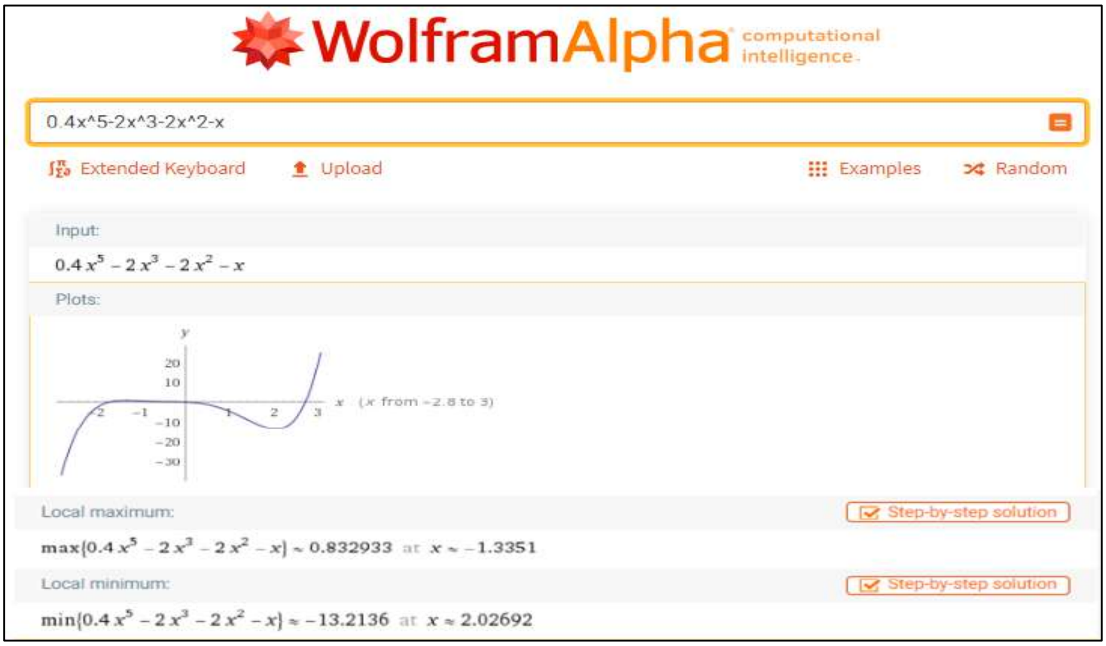


Figura N° 231 Función polinómica 40 Wolfram
Fuente: WolframAlpha

Funcion 55 $f_{(x)} = f_{(x)} = 0.2x^5 - 0.25x^4 - x^3 - 0.5x^2 - 1$

Ubica dos máximos en (0; -1) y (-1;-0,95) , dos mínimos en (-0,4142; -1,02452) y (2,4142; -10,0755) como muestra la figura 232

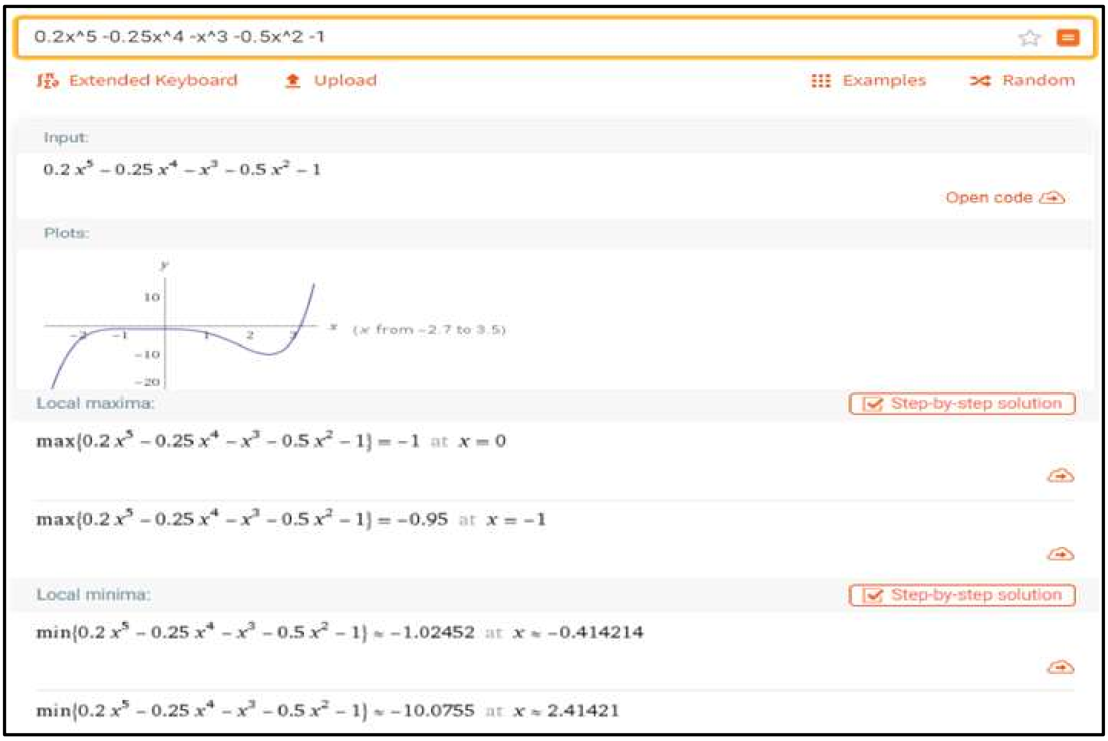


Figura N° 232 Función polinómica 41 Wolfram
Fuente: WolframAlpha

Funcion 56 $f_{(x)} = 0.2x^5 - x^3 - 0.5x^2 + 2x - 1$

Ubica un máximo en (0,7183; -0,15374), un mínimo en (1,7021;-1,1183) como ilustra la figura 233

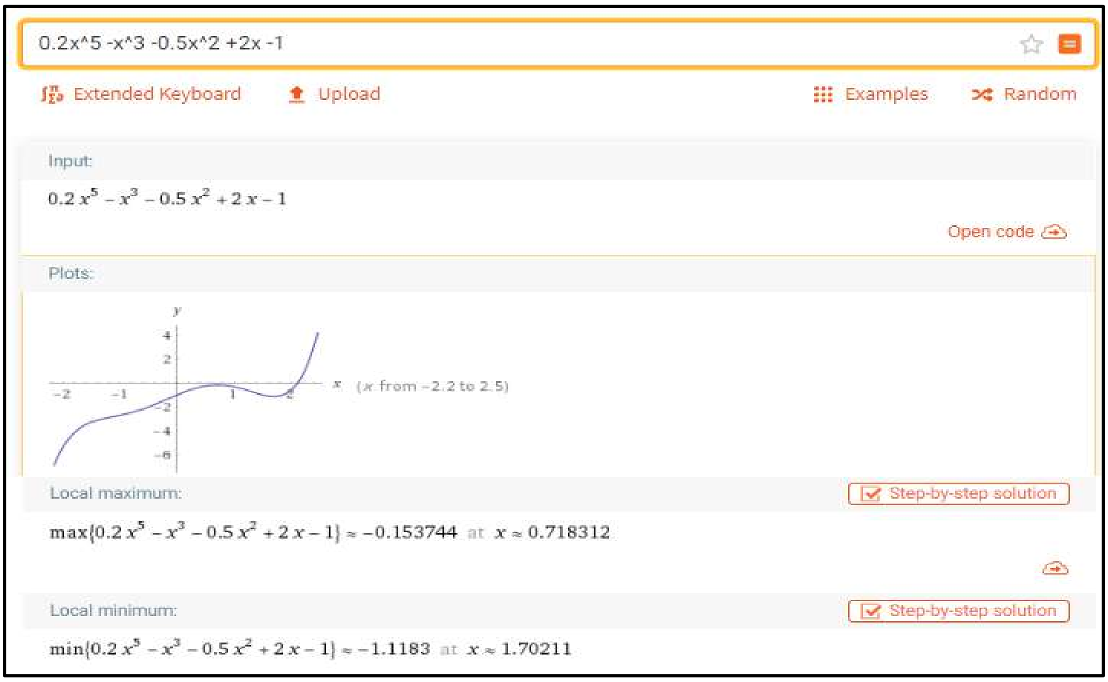


Figura N° 233 Función polinómica 42 Wolfram
Fuente: WolframAlpha

Funcion 57 $f_{(x)} = 0.2x^5 - 0.5x^4 - x^3 + 0.25x^2 + 1.5x - 1$

Ubica un mínimo en (2,91081; -13,28), un máximo en (0,687361; -0,256521) como indica la figura 234

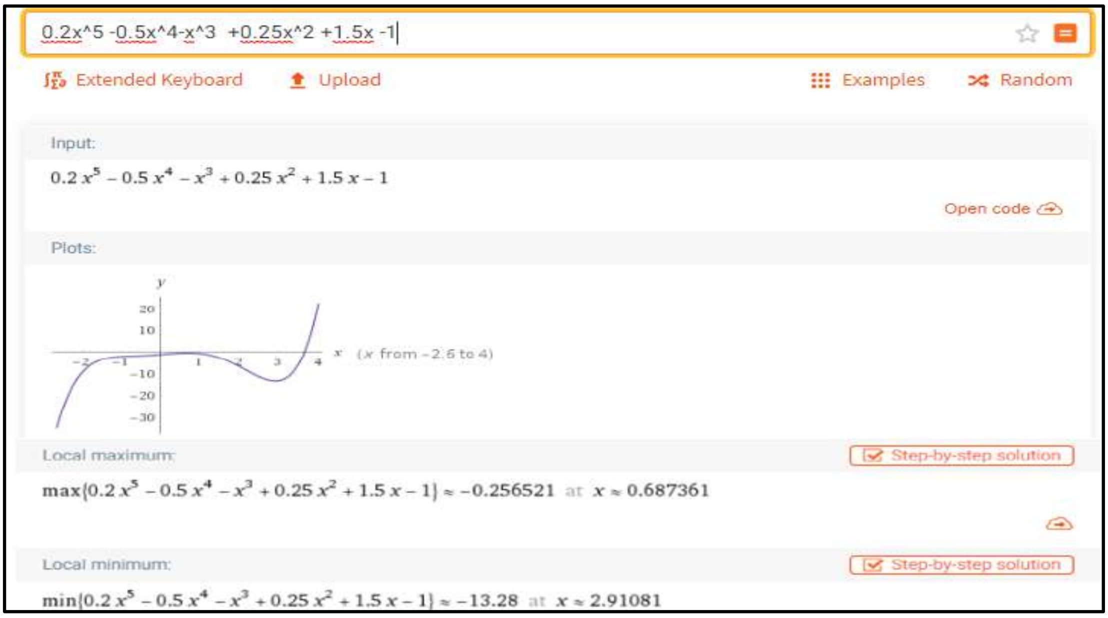


Figura N° 234 Función polinómica 43 Wolfram
Fuente: WolframAlpha

4.7.5.3 Ejecución con el software GeoGebra

Funcion 32 $f_{(x)} = 0,01x^5$

Ubica un punto de inflexión en el origen (0;0) como muestra la figura 235

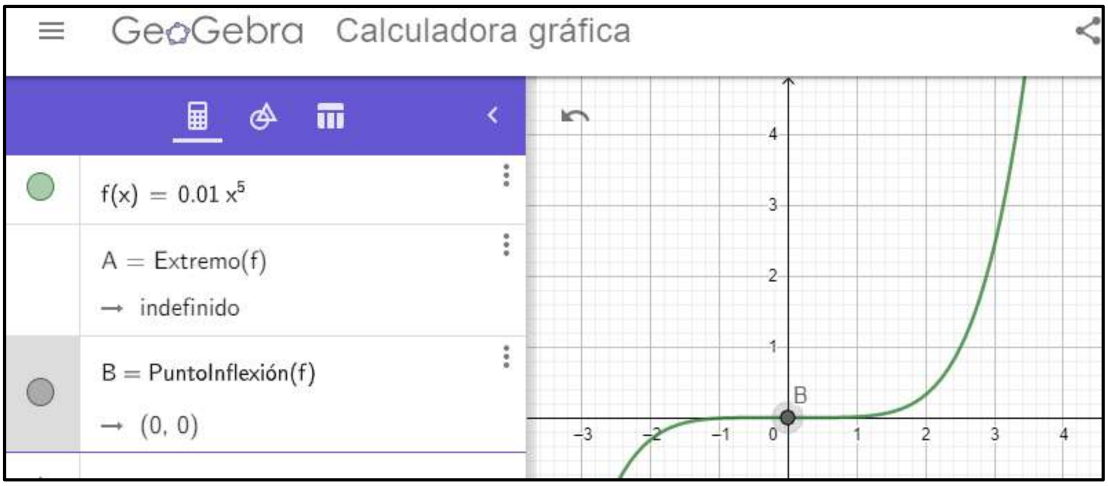


Figura N° 235 Función polinómica 18 GeoGebra
Fuente: GeoGebra

Funcion 33 $f_{(x)} = x^5 - x^4$

Ubica un mìnimo en (0,8; -0,8192) y un màximo en (-0,000000167;0) . Aquí se argumenta que **DERIN ofrece mejor aproximación que GeoGebra** por cuanto al resolver de forma clásica los puntos críticos luego de obtener la primera derivada e igualado a cero $f'(x) = 5x^4 - 4x^3 = 0$ se obtiene como puntos críticos x=0 y x=0,8 . En cuanto a puntos de inflexión ubica el punto (0,6;-0,051) como muestra la figura 236

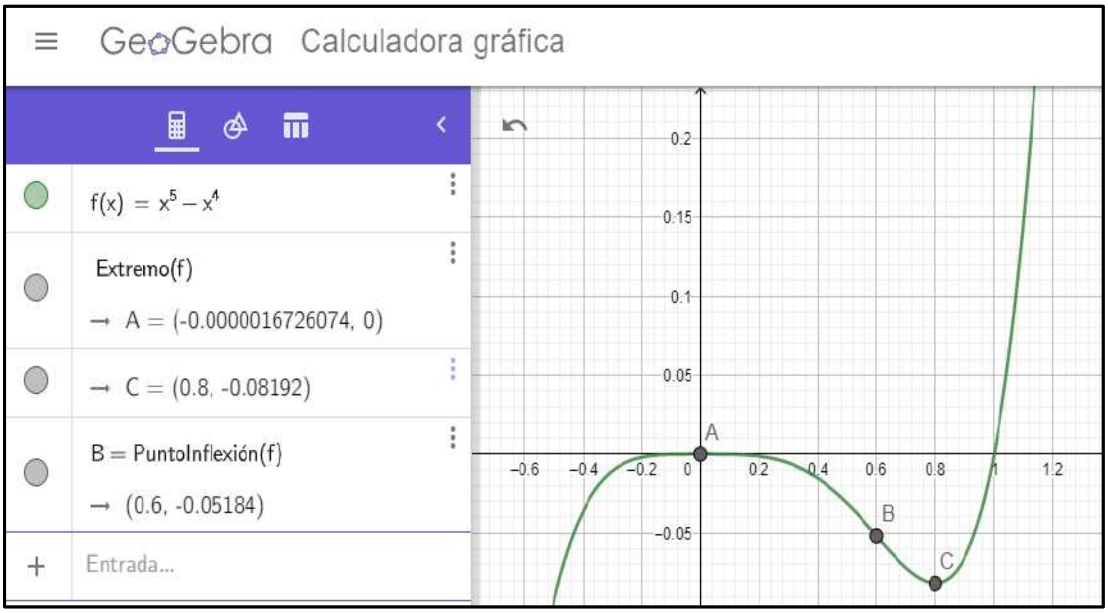


Figura N° 236 Función polinómica 19 GeoGebra
Fuente: GeoGebra

Funcion 34 $f_{(x)} = 3x^5 - 5x^3$

Ubica extremos en los puntos (1;-2) y (-1;2) así como puntos de inflexión en (-0,707;1,237) , (0,707;-1,237) y en (0;0) como indica la fgura 237

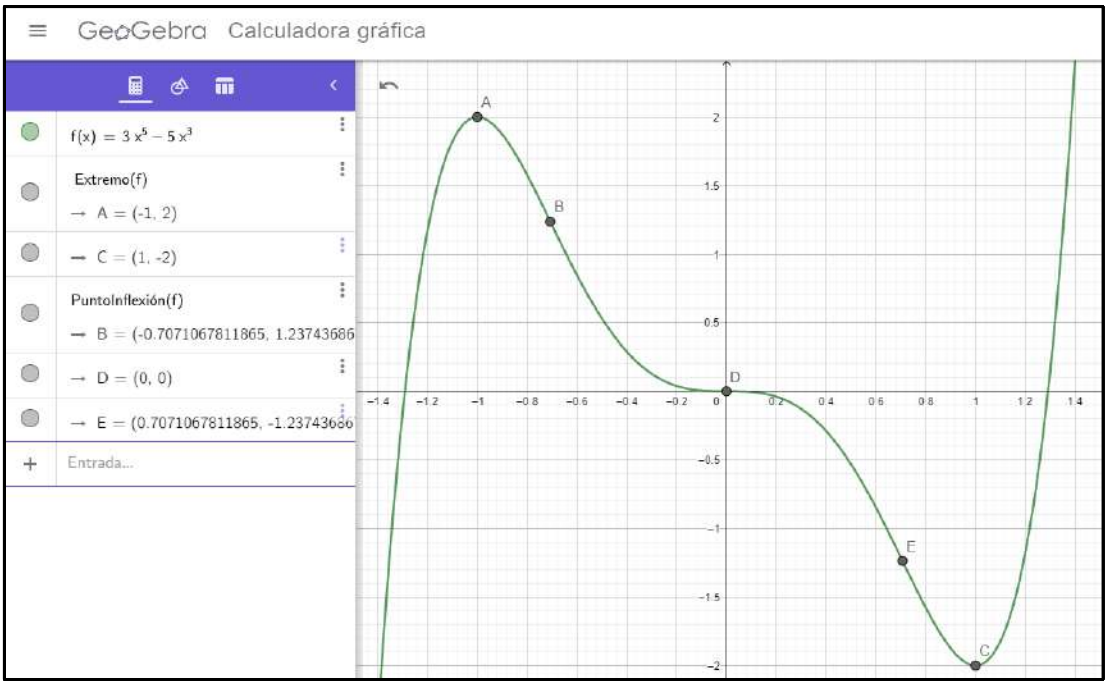


Figura N° 237 Función polinómica 20 GeoGebra
Fuente: GeoGebra

Funcion 35 $f_{(x)} = x^5 - 2x^2$

Ubica dos extremos en (0;0) y en (0,928;-1,03) así como una inflexión en (0,5848;-0,615) como ilustra la figura 238

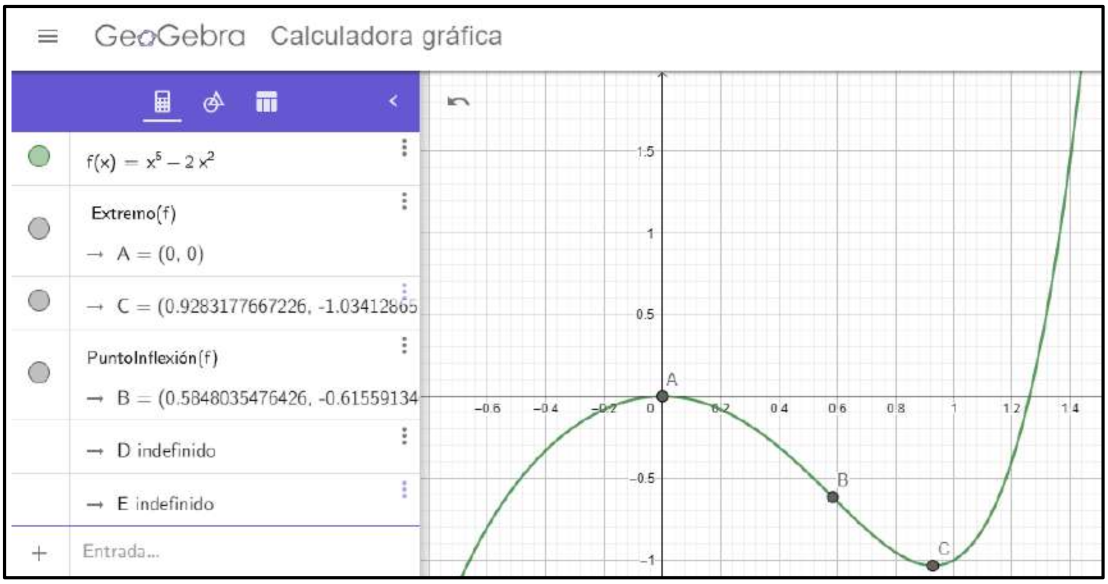


Figura N° 238 Función polinómica 21 GeoGebra
Fuente: GeoGebra

Funcion 36 $f_{(x)} = x^5 - 5x$

Ubica extremos en (1;-4) y en (-1;4) y una inflexión en (0;0) como muestra la figura 239

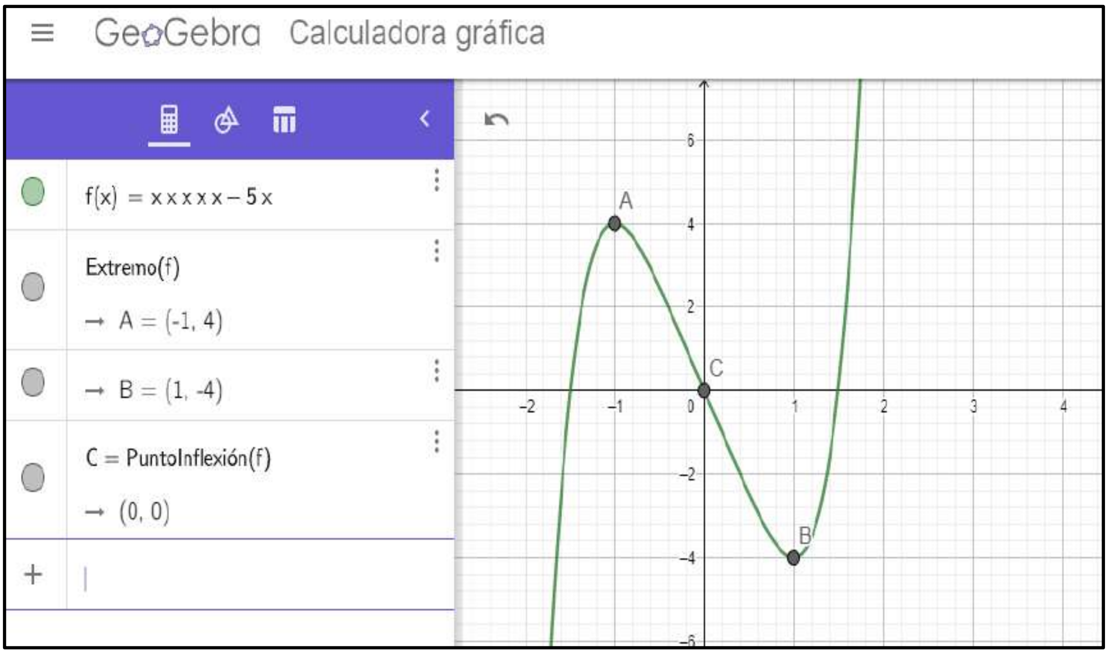


Figura N° 239 Función polinómica 22 GeoGebra
Fuente: GeoGebra

Funcion 37 $f_{(x)} = x^5 + 0,5x$

Ubica una inflexión en el origen como muestra la figura 240

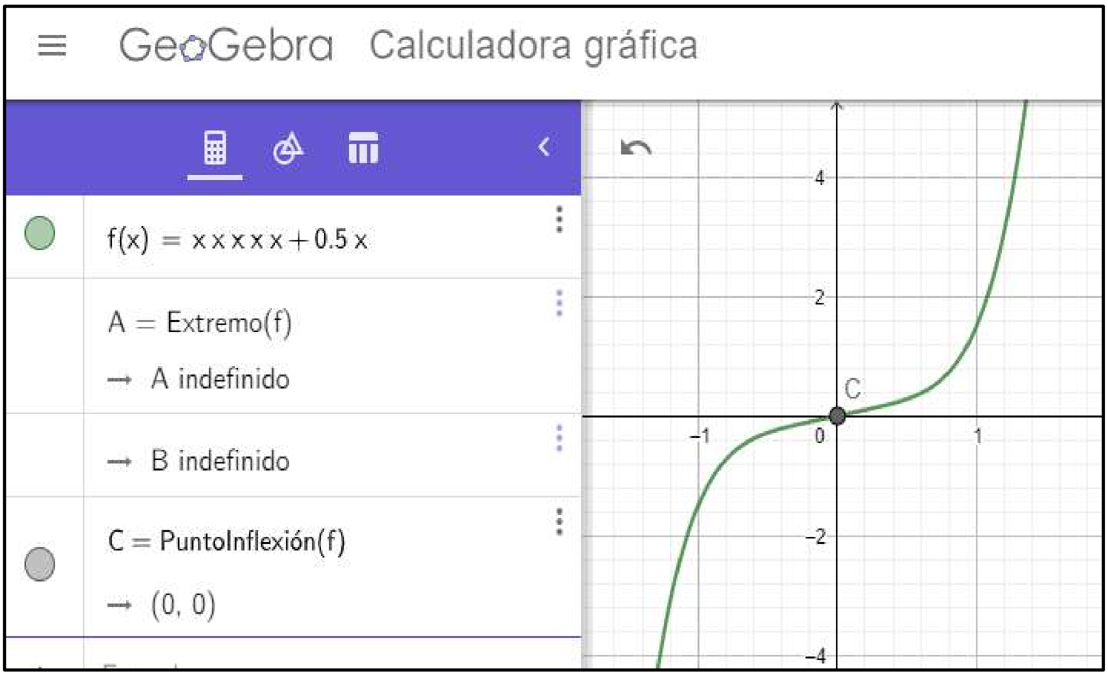


Figura N° 240 Función polinómica 23 GeoGebra
Fuente: GeoGebra

Funcion 38 $f_{(x)} = x^5 - 4x^4 + 4x^3$

Ubica extremos en (1,1999999998;1,11) y en (2,000000001;0) así como puntos de inflexión en (0;0) , (0,71;0,59) y en (1,69;0,46) como ilustra la figura 241

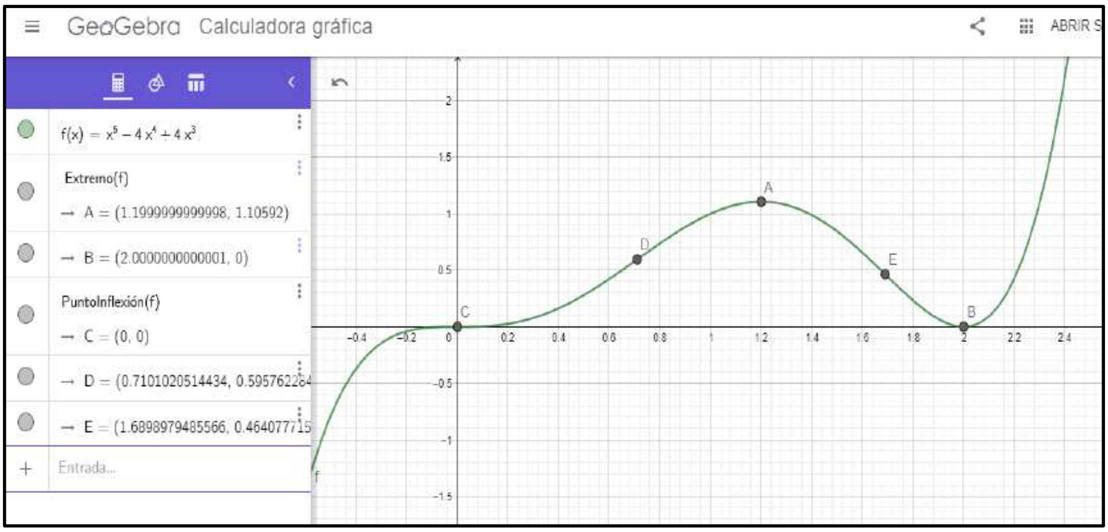


Figura N° 241 Función polinómica 24 GeoGebra
Fuente: GeoGebra

Funcion 39 $f_{(x)} = x^5 - 3x^4 + 3x^2$

Ubica extremos en (-0,692;0,618), (0;0), (0,892;1,052) y (2,137;-4,297) así como puntos de inflexión en (-0,372;0,35) , (0,476;0,55) y (-1,696;-2,157) como muestra la figura 242

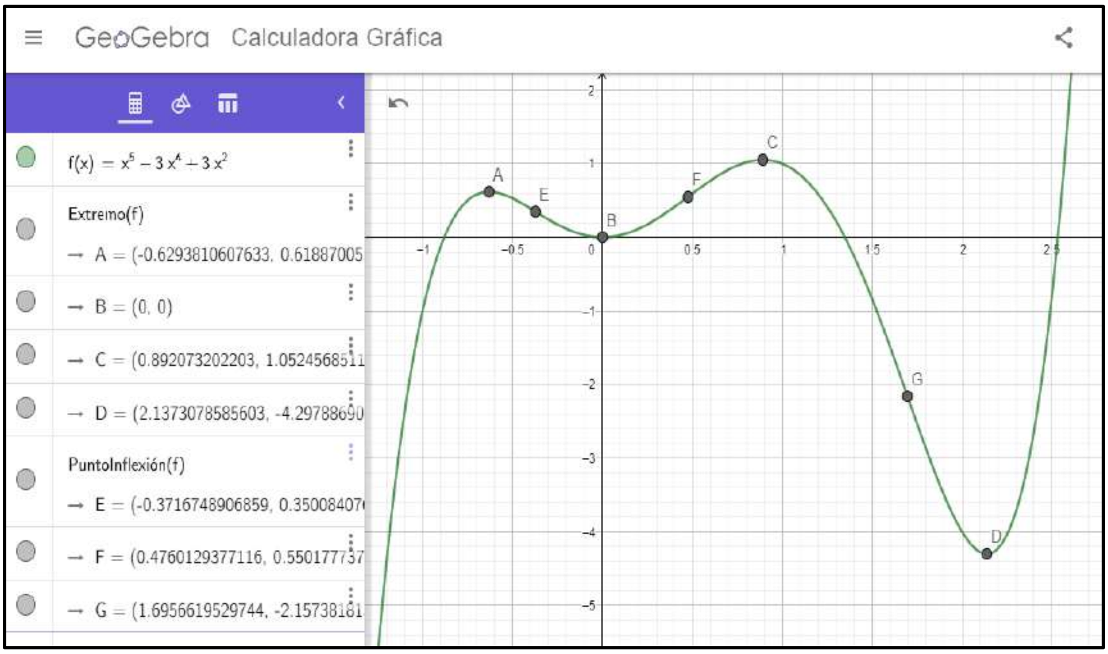


Figura N° 242 Función polinómica 25 GeoGebra
Fuente: GeoGebra

Funcion 40 $f_{(x)} = x^5 - 2x^4 - 2x$

Ubica extremos en (-0,569;0,868) y en (1,683;-5,909) y un punto de inflexión en (1,2;-4,058) como muestra la figura 243

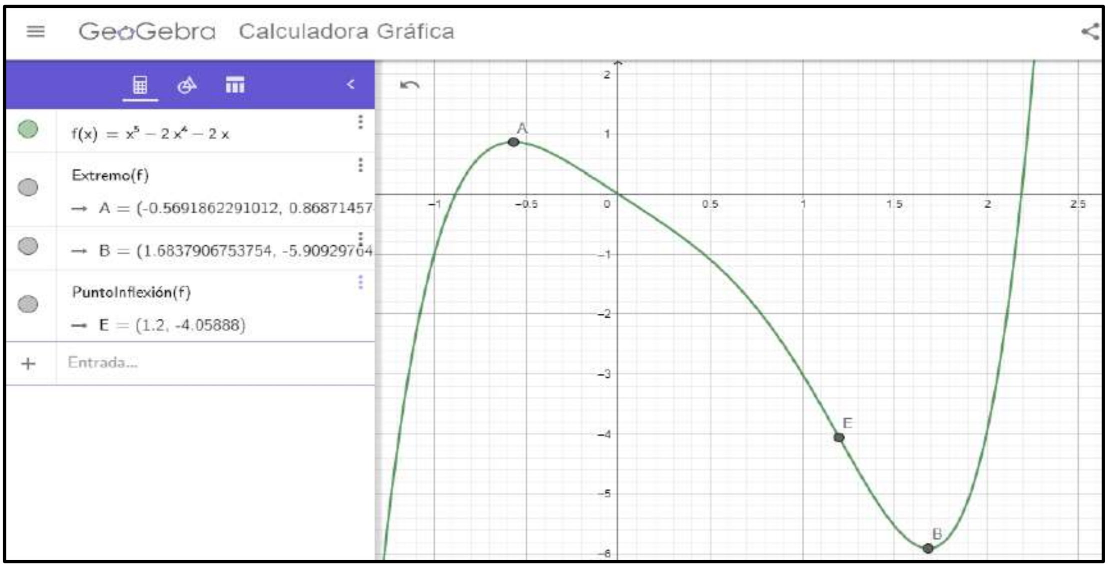


Figura N° 243 Función polinómica 26 GeoGebra
Fuente: GeoGebra

Funcion 41 $f_{(x)} = 0,2x^5 - 0,5x^4 - 1$

Ubica dos puntos extremo en (2;-2,6) y en (-0;000000167;-1) y una inflexión en (1,5;-2,0125) como muestra la figura 232. **Se deduce que DERIN tiene mejor aproximación** dado que la primera derivada igualada a cero determina la ecuación $x^4 - 2x^3 = 0$ donde alcanza como soluciones x = 0 y x = 2

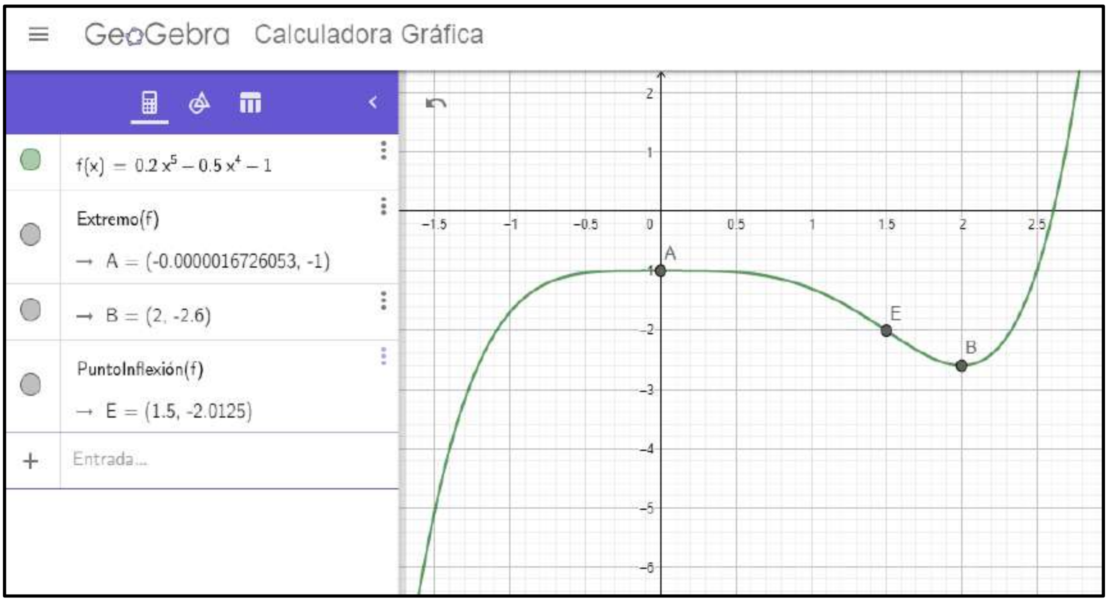


Figura N° 244 Función polinómica 27 GeoGebra
Fuente: GeoGebra

Función 42 $f_{(x)} = x^5 - 2x^3 - 3x^2$

Ubica extremos en (0;0) y en (1,428;-6,00345) y una inflexión en (0,956;-3,69) como muestra la figura 245

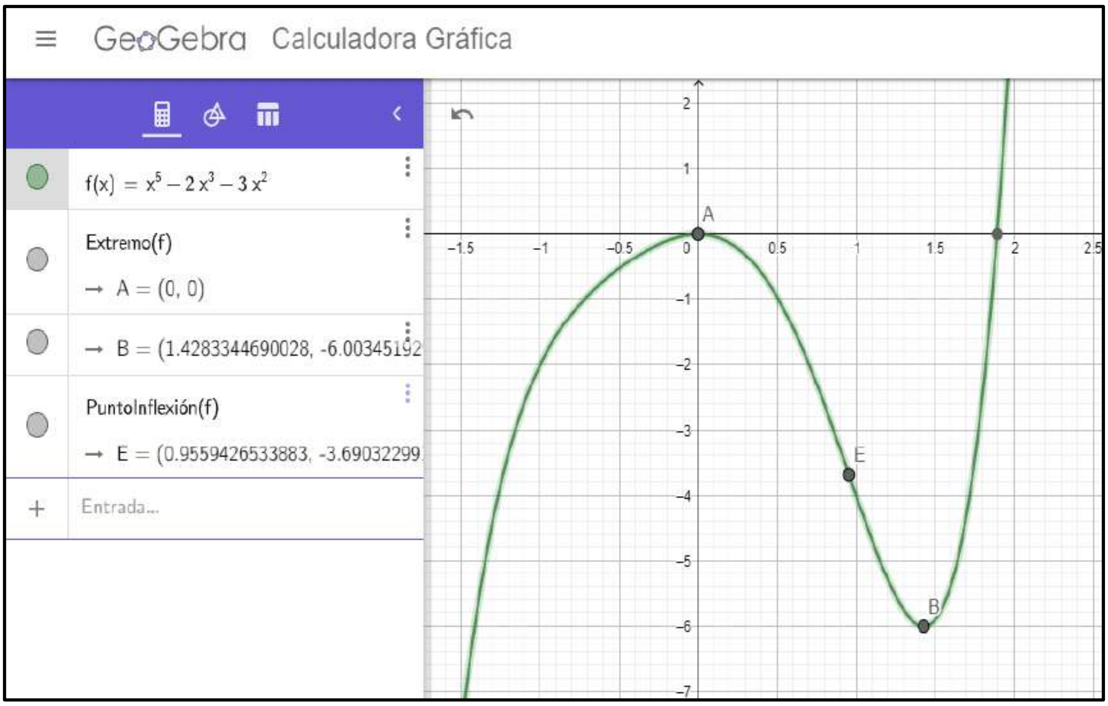


Figura N° 245 Función polinómica 28 GeoGebra
Fuente: GeoGebra

Funcion 43 $f_{(x)} = x^5 - 1.5x^3 + 1.2x^2$

Ubica dos extremos en (0;0) y en (-1,148;1856) y una inflexión en (-0,7774;1,146) como muestra la figura 246

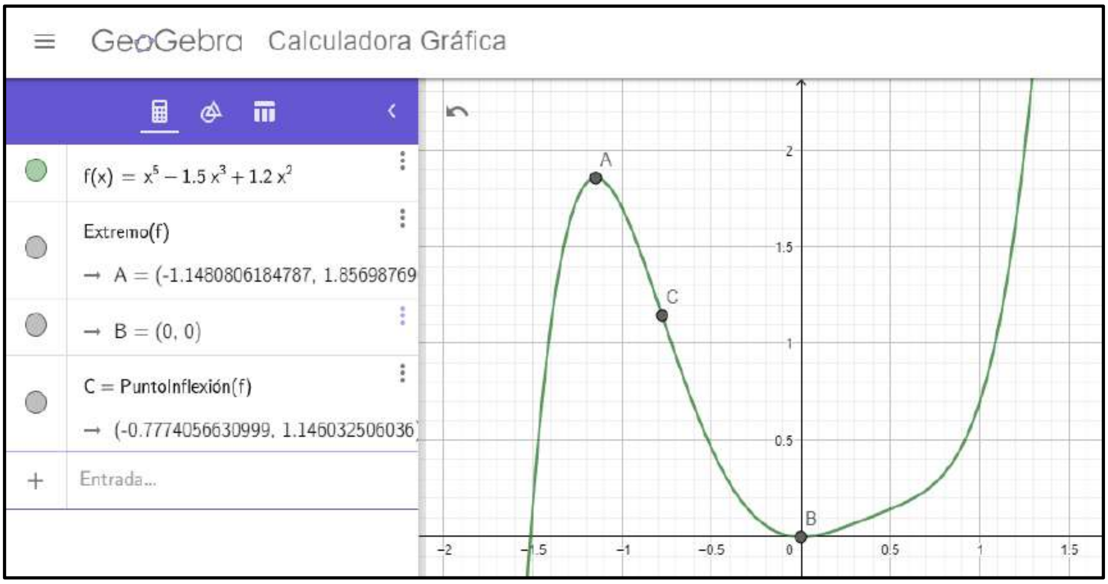


Figura N° 246 Función polinómica 29 GeoGebra
Fuente: GeoGebra

Funcion 44 $f_{(x)} = x^5 - 2x^3 + 2x$

Ubica inflexiones en (0;0), (-0,774;-0,898) y (0,774;0898) como muestra la figura 247

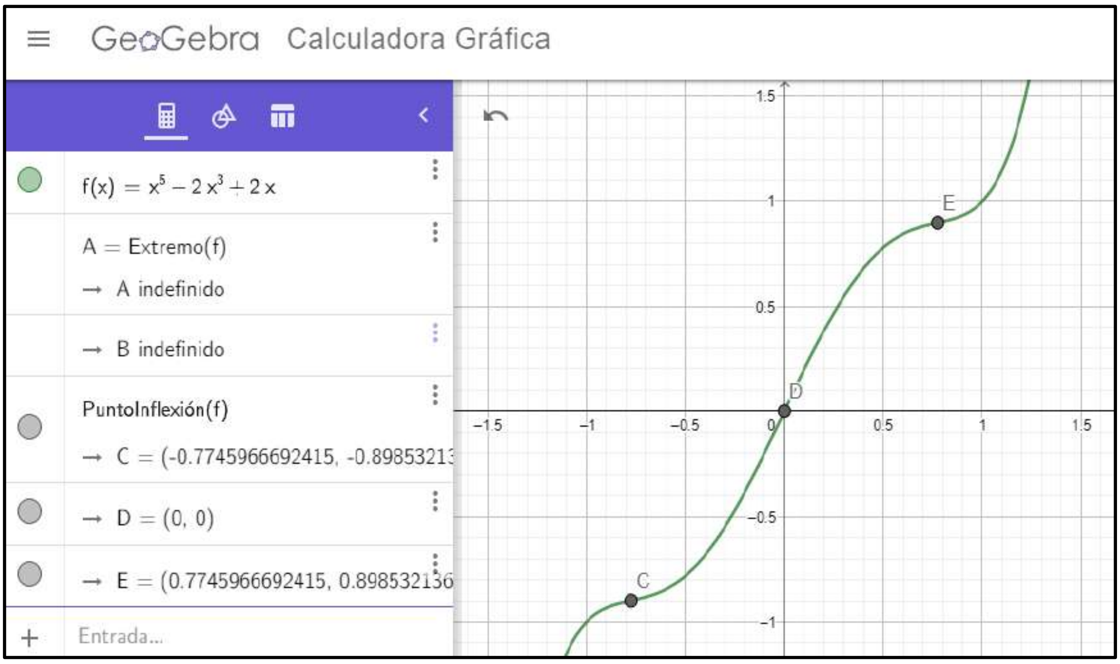


Figura N° 247 Función polinómica 30 GeoGebra
Fuente: GeoGebra

Funcion 45 $f_{(x)} = x^5 - 2x^3 + x$

Ubica extremos en (-1;0), (-0,447; -0,286), (0,447; -0,286) y (1;0) e inflexiones en (0;0) , (0,7746;0,1239) y (-0,7746;-0,1239) como muestra la figura 248

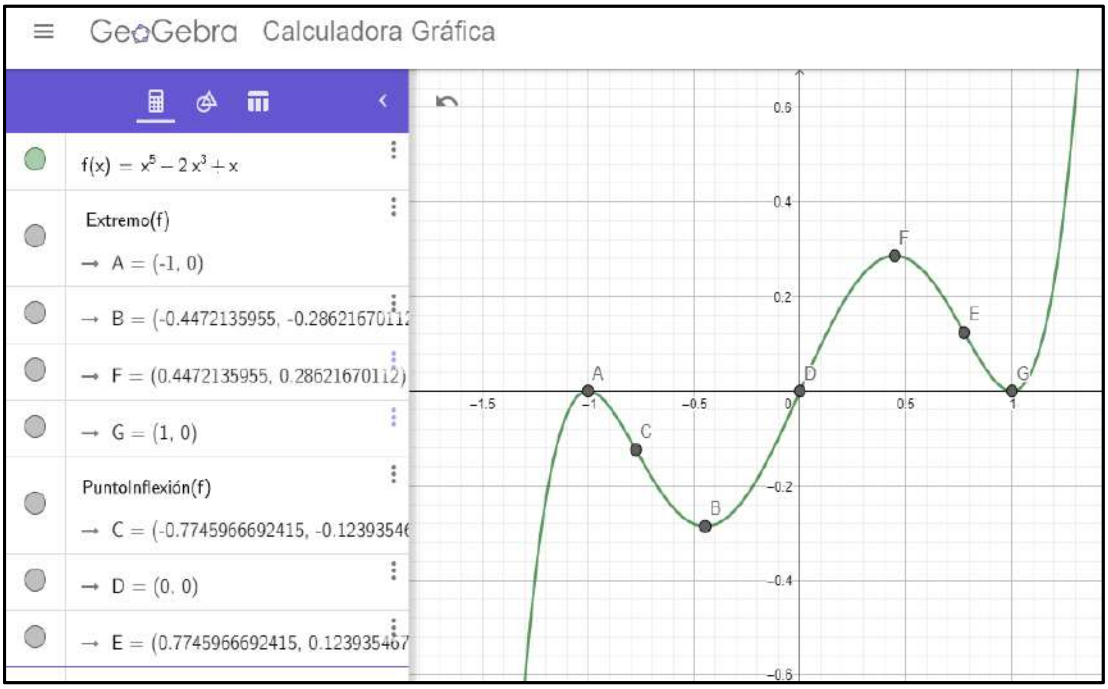


Figura N° 248 Función polinómica 31 GeoGebra
Fuente: GeoGebra

Funcion 46 $f_{(x)} = x^5 - x^3 + 1$

GeoGebra: Ubica dos extremos en (-0,7746;1,1859) y (0,7746; 0,8141) y tres inflexiones en (0;1) , (-0,5477:1,115) y (0,5477;0,8849) como muestra la figura 249

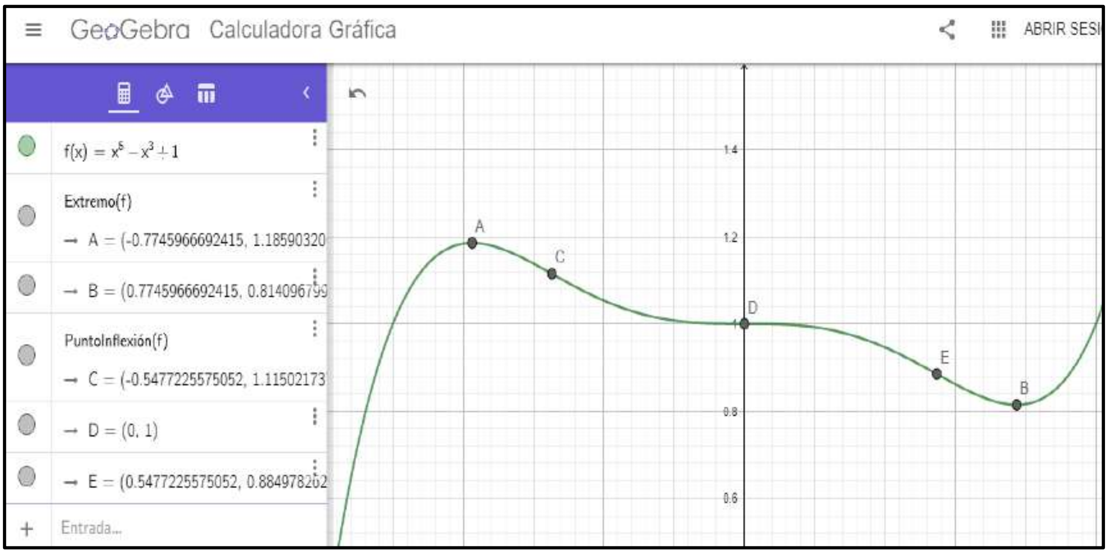


Figura N° 249 Función polinómica 32 GeoGebra
Fuente: GeoGebra

Funcion 47 $f_{(x)} = x^5 + x^2 - 2x$

Ubica extremos en (-0,9383;2,0297) y (0,6231;-0,764) y una inflexión en (-0,4641;1,1222) como muestra la figura 250

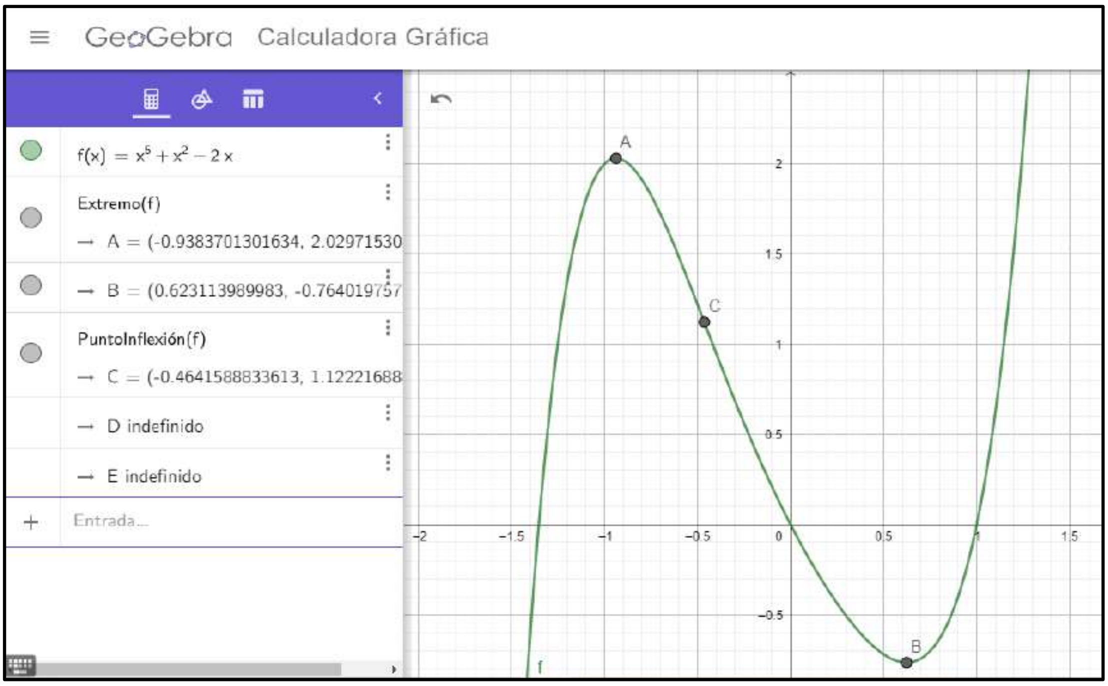


Figura N° 250 Función polinómica 33 GeoGebra
Fuente: GeoGebra

Funcion 48 $f_{(x)} = x^5 + x^2 - 2$

Ubica extremos en (-0,7368;-1,6743) y una inflexión en (-0,46415; -1,8061) como muestra la figura 251

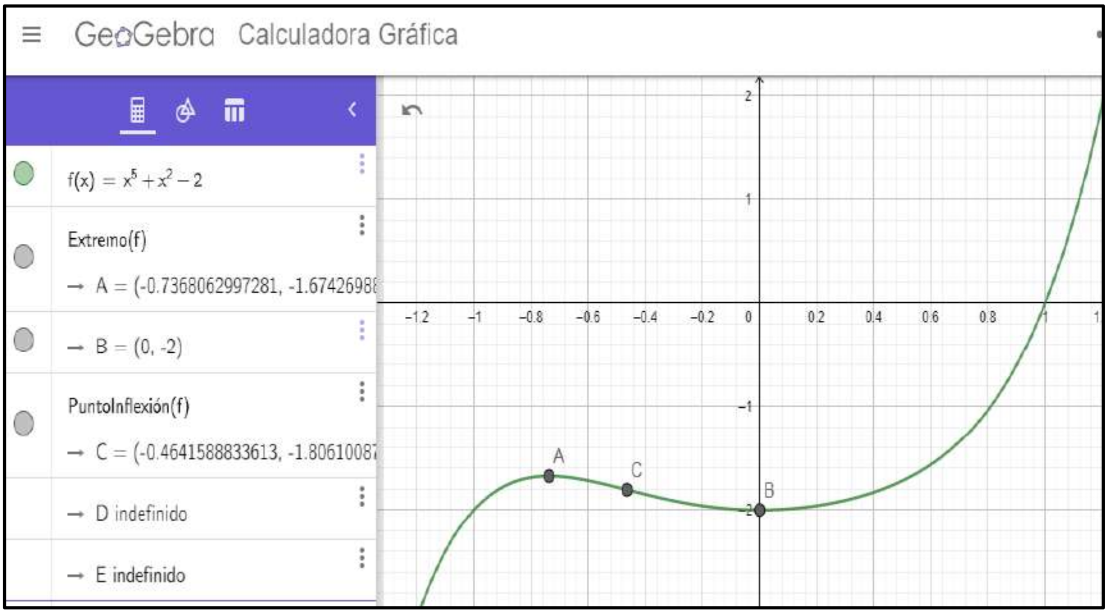


Figura N° 251 Función polinómica 34 GeoGebra
Fuente: GeoGebra

Funcion 49 $f_{(x)} = x^5 - 3x + 1$

Ubica dos extremos en (0,88011;-1,1123) y (-0,88011;3,1123) así como un punto de inflexión en (0;1) como muestra la figura 252

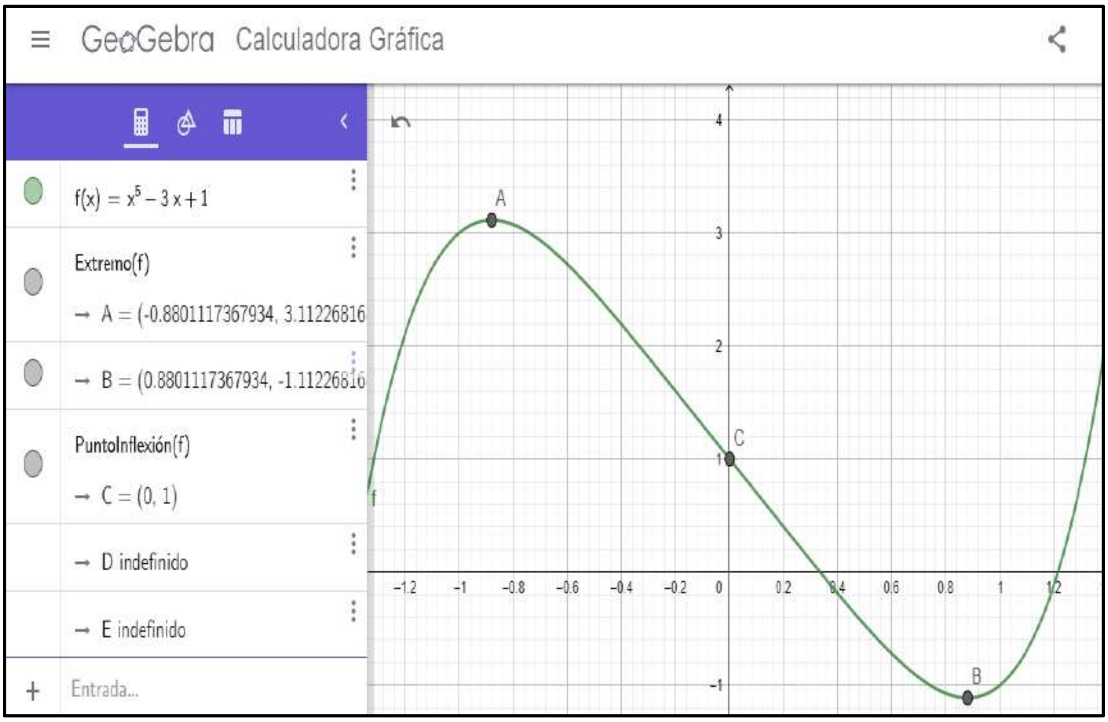


Figura N° 252 Función polinómica 35 GeoGebra
Fuente: GeoGebra

Funcion 50 $f_{(x)} = x^5 - x^4 + 2x^3 - 4x^2$

Ubica extremos en (0;0) y en (1,0723;-2,0378) así como una inflexión en (0,6396;-1,17346) como muestra la figura 253

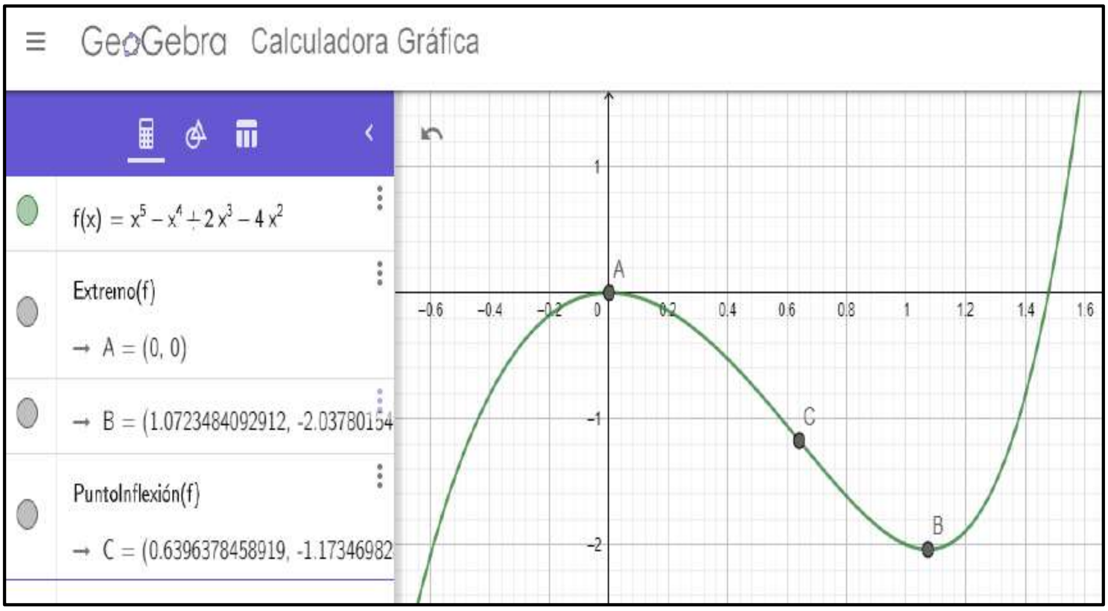


Figura N° 253 Función polinómica 36 GeoGebra
Fuente: GeoGebra

Funcion 51 $f_{(x)} = x^5 - x^4 + 0.5x^2 - 4x$

Ubica dos extremos en (.0,8389;2,7967) y (1,1618;-3,6775) así como una inflexión en (-0,24347; 0,99915) como muestra la figura 254

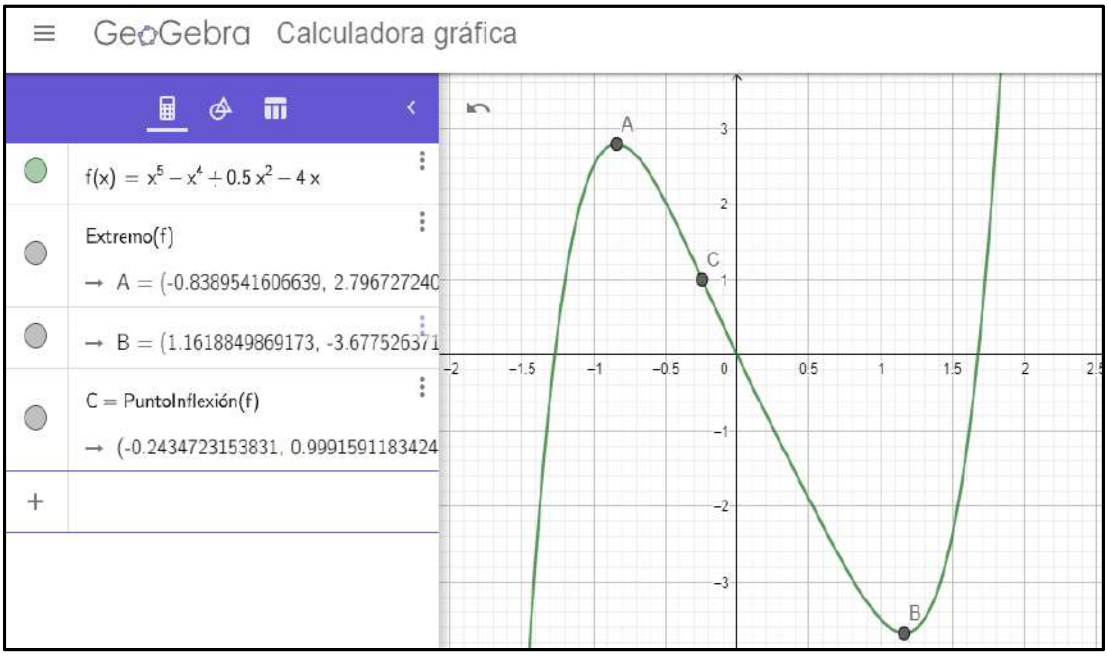


Figura N° 254 Función polinómica 37 GeoGebra
Fuente: GeoGebra

Funcion 52 $f_{(x)} = x^5 - x^4 - 2x^3 - x$

Ubica extremos en (-0,8736; 1,11576) y en (1,5992; -5,8598) así como inflexiones en (-0,53066; 0,7081) , (0;0) y (1,13066; -3,80799) como muestra la figura 255

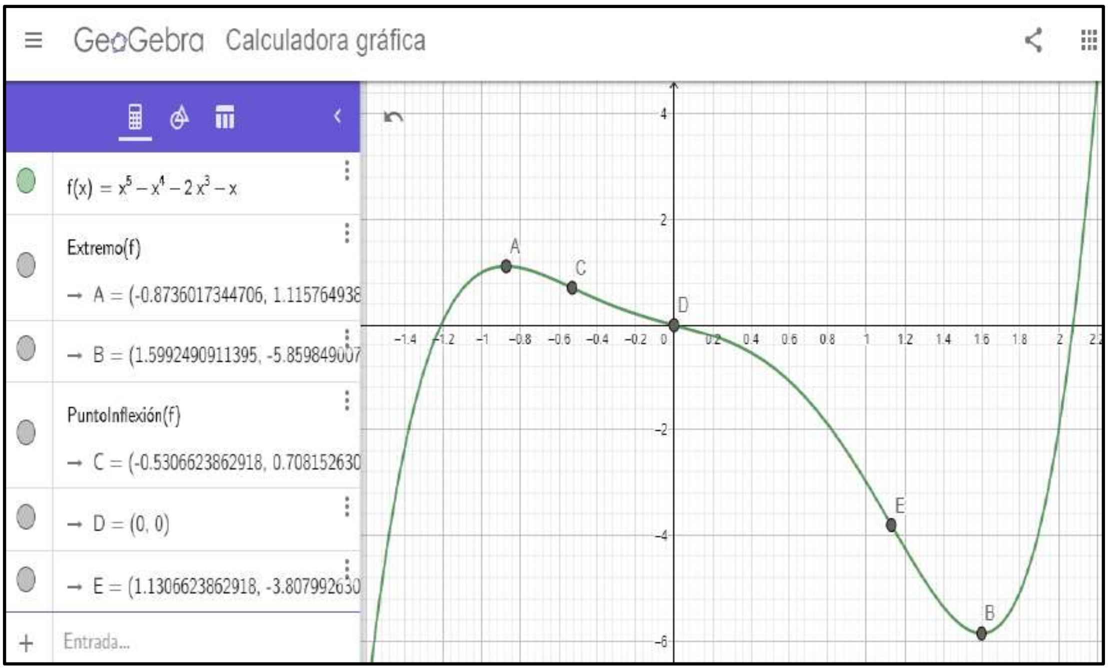


Figura N° 255 Función polinómica 38 GeoGebra
Fuente: GeoGebra

Funcion 53 $f_{(x)} = 0.2x^5 - 0.25x^4 - x^2 - 2x$

Ubica extremos en (-0,70347;0,81639) y (1,873708; -5,72068) así como una inflexión en (1,13686; -3,60397) como ilustra la figura 256

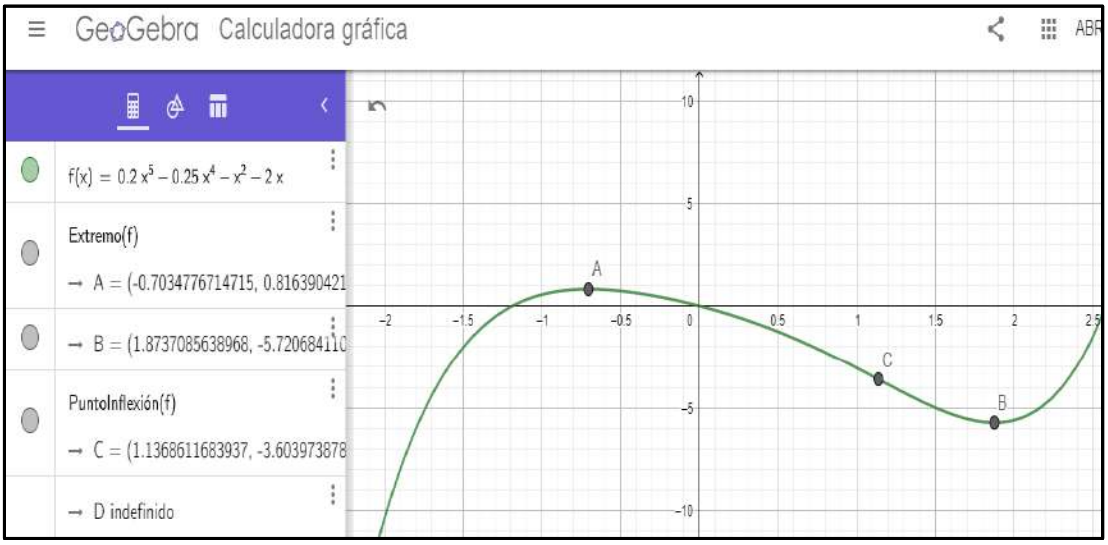


Figura N° 256 Función polinómica 39 GeoGebra
Fuente: GeoGebra

Funcion 54 $f_{(x)} = 0.4x^5 - 2x^3 - 2x^2 - x$

Ubica extremos en (-1,3351; 0,8329) y (2,02692; -13,2136) así como inflexiones en los puntos (-1; 0,6) , (-0,366; 0,1935) y (1,366; -8,2935) como ilustra la figura 257

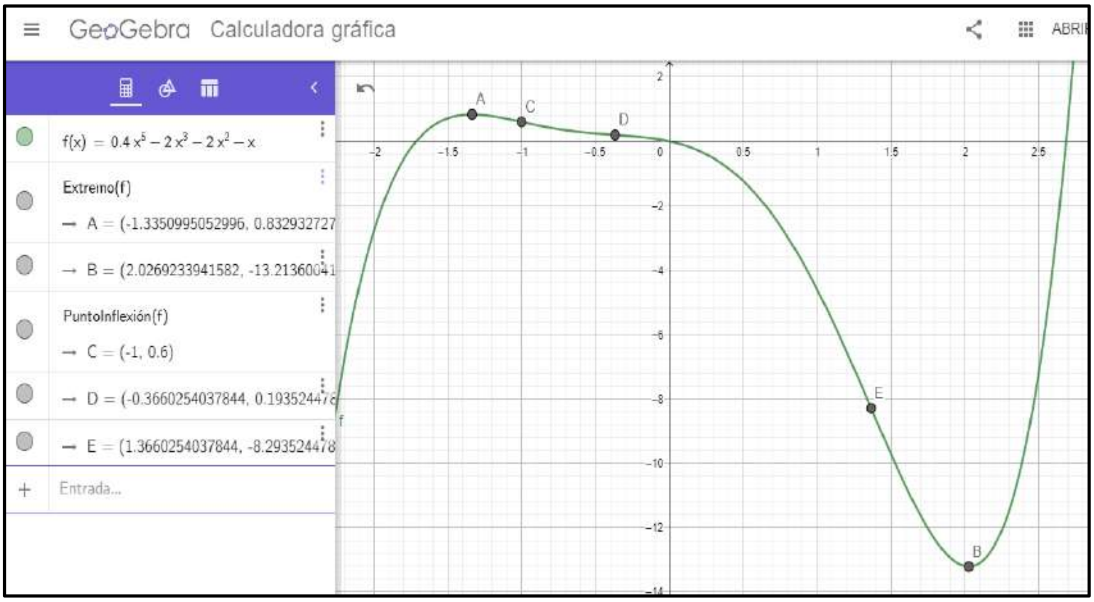


Figura N° 257 Función polinómica 40 GeoGebra
Fuente: GeoGebra

Funcion 55 $f_{(x)} = f_{(x)} = 0.2x^5 - 0.25x^4 - x^3 - 0.5x^2 - 1$

Ubica extremos en (-1; -0,95) , (-0,4142; -1,0245) , (0; -1) y (2,4142; -10,0754) así como puntos de inflexión en (-0,7726; -0,9814) , (-0,18904; -1,01147) y (1,71167; -6,6872) como muestra la figura 258

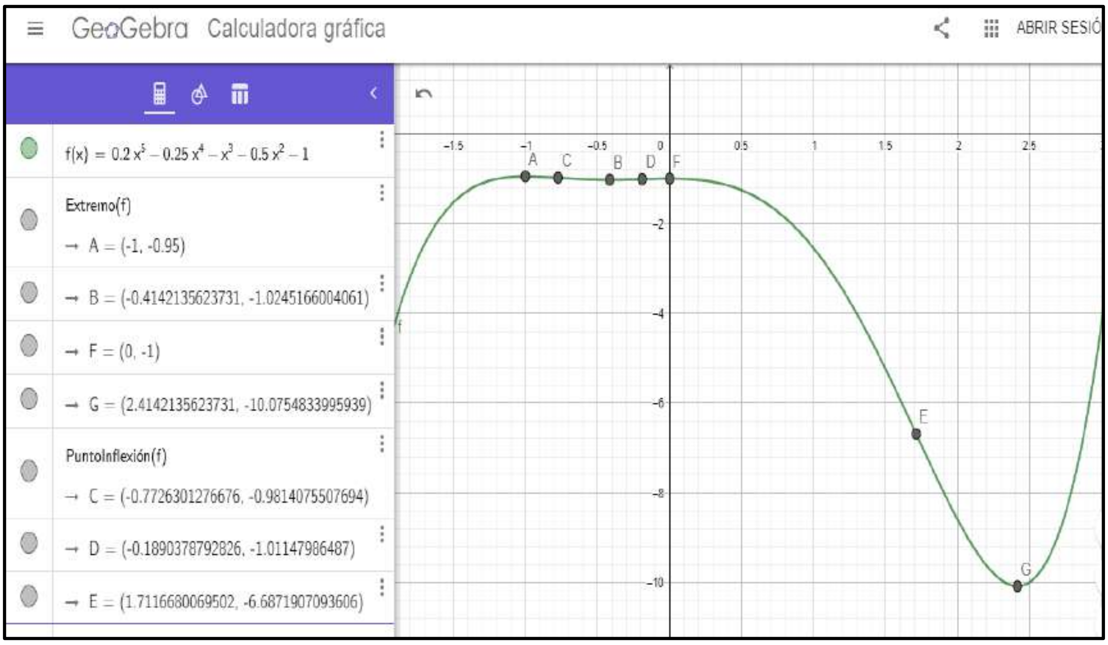


Figura N° 258 Función polinómica 41 GeoGebra
Fuente: GeoGebra

Funcion 56 $f_{(x)} = 0.2x^5 - x^3 - 0.5x^2 + 2x - 1$

Ubica extremos en (0,7183 ; -0,15374) y (1,7021;-1,1183) así como inflexiones en (-1,1309; -2,82487) , (-0,16993; --1,34944) y (1,30084; -0,70068) como ilustra la figura 259

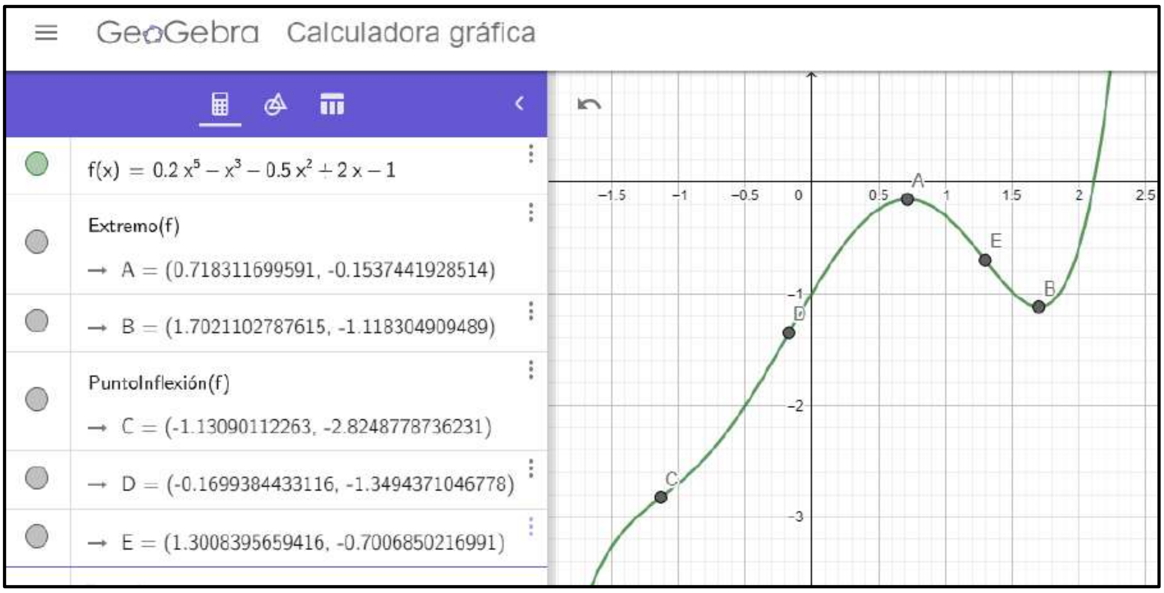


Figura N° 259 Función polinómica 42 GeoGebra
Fuente: GeoGebra

Funcion 57 $f_{(x)} = 0.2x^5 - 0.5x^4 - x^3 + 0.25x^2 + 1.5x - 1$

Ubica extremos en (0,68763;-0,25652) y (2,9108;-13,28) e inflexiones en (-0,743525;-1,76429) , (0,07762, -0,882548) y ((2,16591; -8,2094) como muestra la figura 260

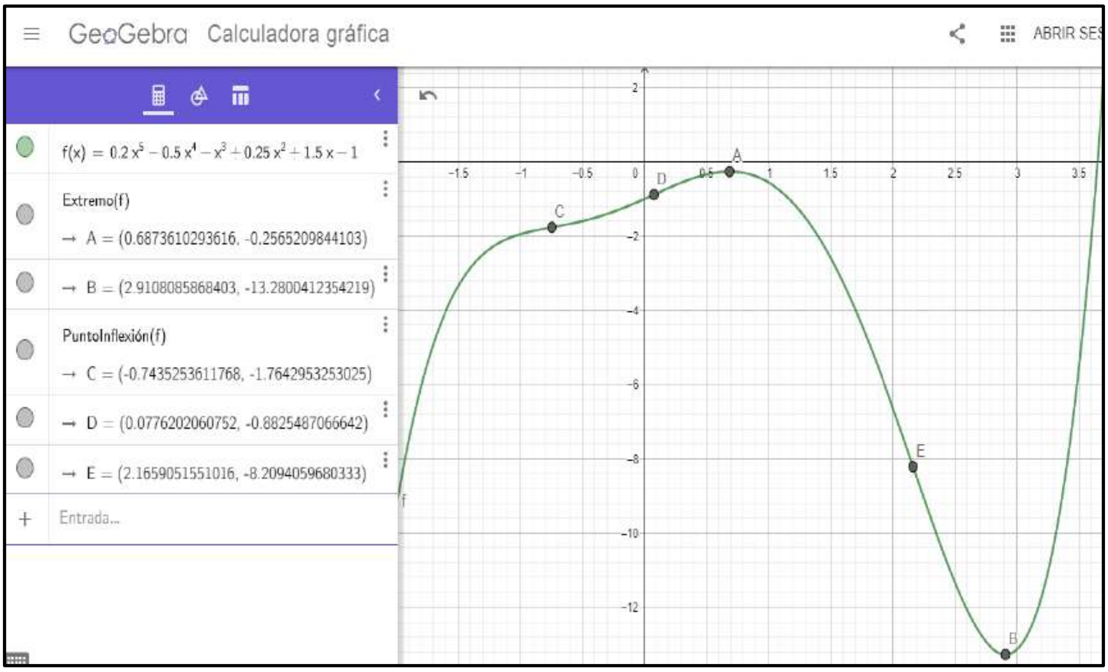


Figura N° 260 Función polinómica 43 GeoGebra
Fuente: GeoGebra

4.7.5.4 Ejecución con el software Symbolab

Funcion 32 $f_{(x)} = 0,01x^5$

Indica que (0,0) es un punto de silla como muestra la figura 261

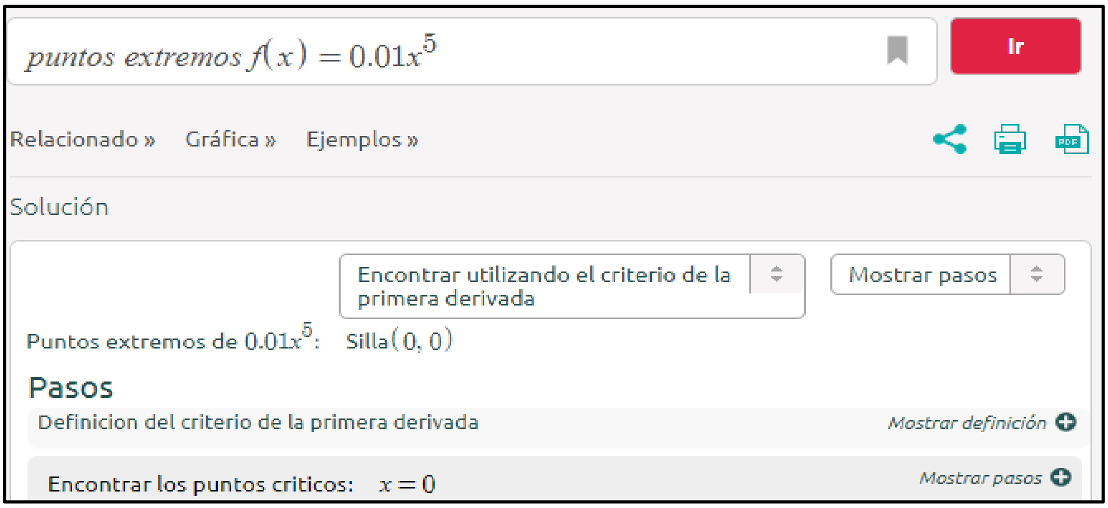


Figura N° 261 Función polinómica 18 Symbolab
Fuente: Symbolab

Funcion 33 $f_{(x)} = x^5 - x^4$

Ubica un máximo en (0;0), un mínimo en $\left(\frac{4}{5}; -\frac{256}{3125}\right)$ como muestra la figura 262

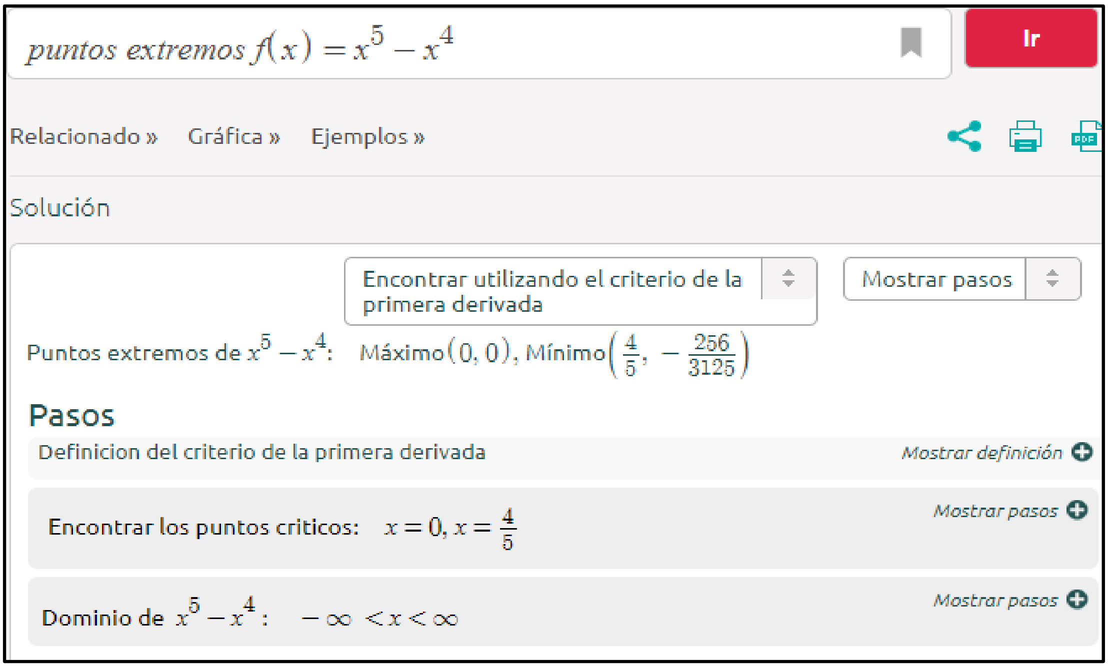


Figura N° 262 Función polinómica 19 Symbolab
Fuente: Symbolab

Funcion 34 $f_{(x)} = 3x^5 - 5x^3$

Ubica un mínimo en (1;-2) y un máximo en (-1;2) asi como un punto silla en (0;0) como muestra la figura 263

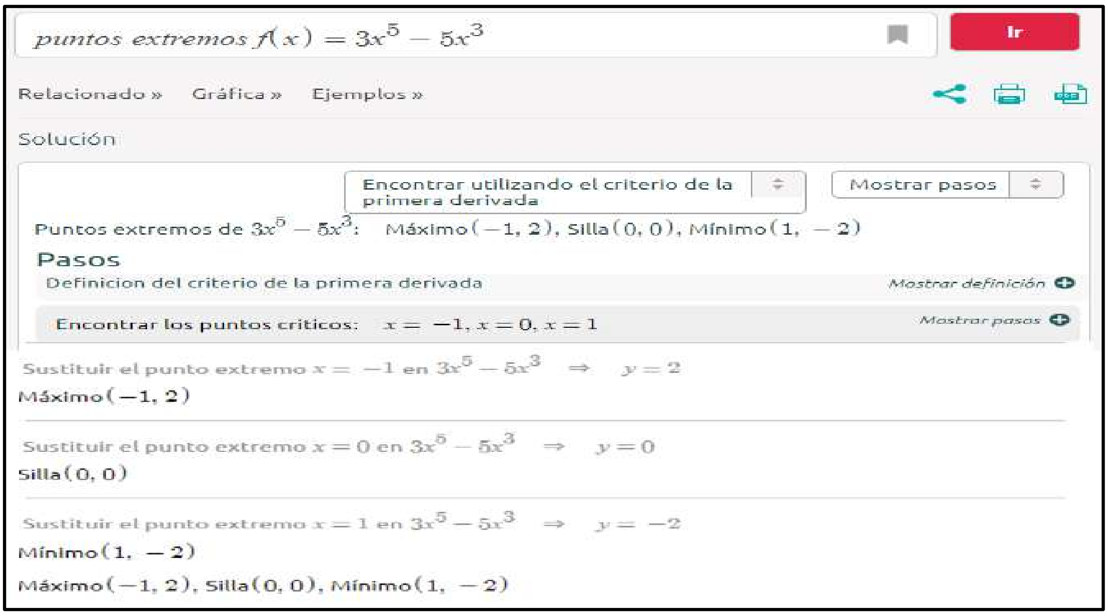


Figura N° 263 Función polinómica 20 Symbolab
Fuente: Symbolab

Funcion 35 $f_{(x)} = x^5 - 2x^2$

Ubica un máximo en (0;0), un mínimo en $\left(\sqrt[3]{\frac{4}{5}};\ \left(\frac{4}{5}\right)^{\frac{5}{3}} - 2\left(\frac{4}{5}\right)^{\frac{2}{3}}\right)$ donde $\sqrt[3]{\frac{4}{5}} = 0{,}928$

como ilustra la figura 264

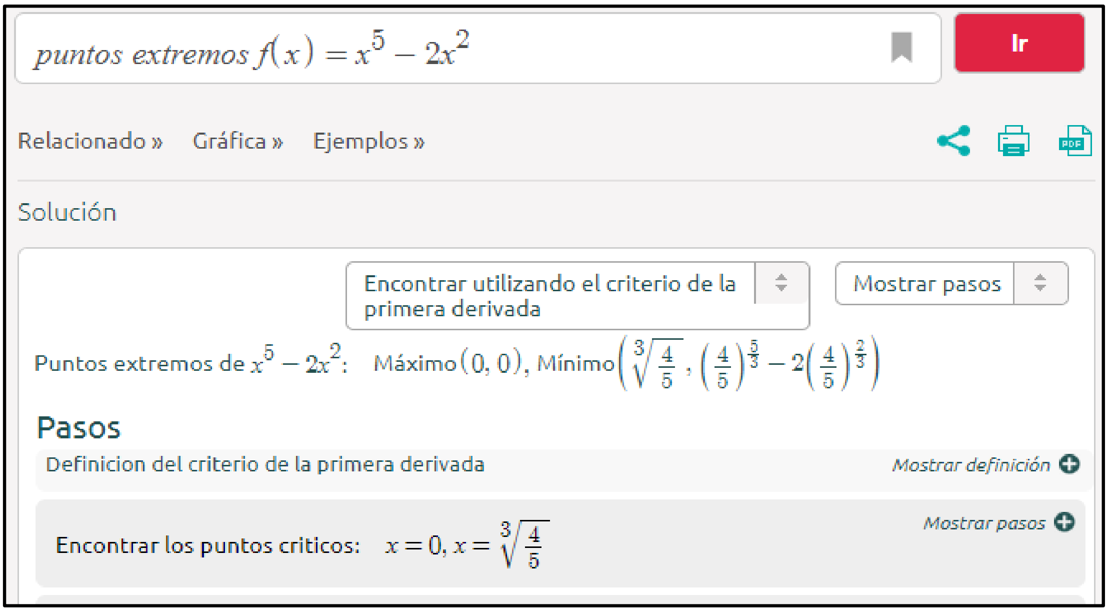


Figura N° 264 Función polinómica 21 Symbolab
Fuente: Symbolab

Funcion 36 $f_{(x)} = x^5 - 5x$

Ubica un mínimo en (1;-4) y un máximo en (-1;4), como ilustra la figura 265

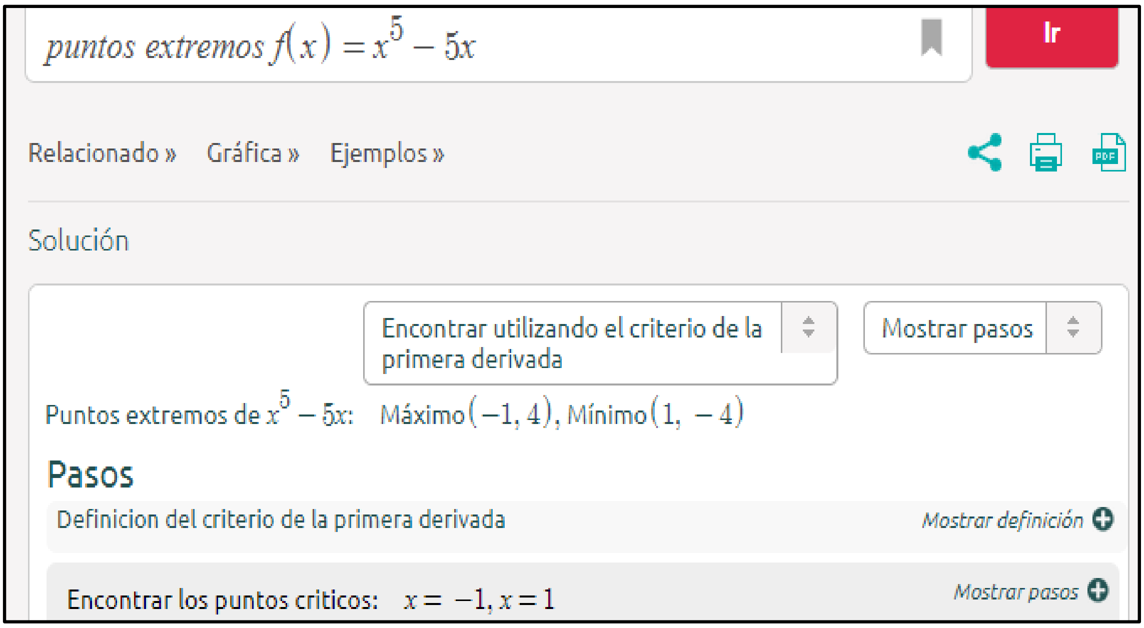


Figura N° 265 Función polinómica 22 Symbolab
Fuente: Symbolab

Funcion 37 $f_{(x)} = x^5 + 0,5x$

No encuentra puntos extremos ni **puntos de silla** como ilustra la figura 266

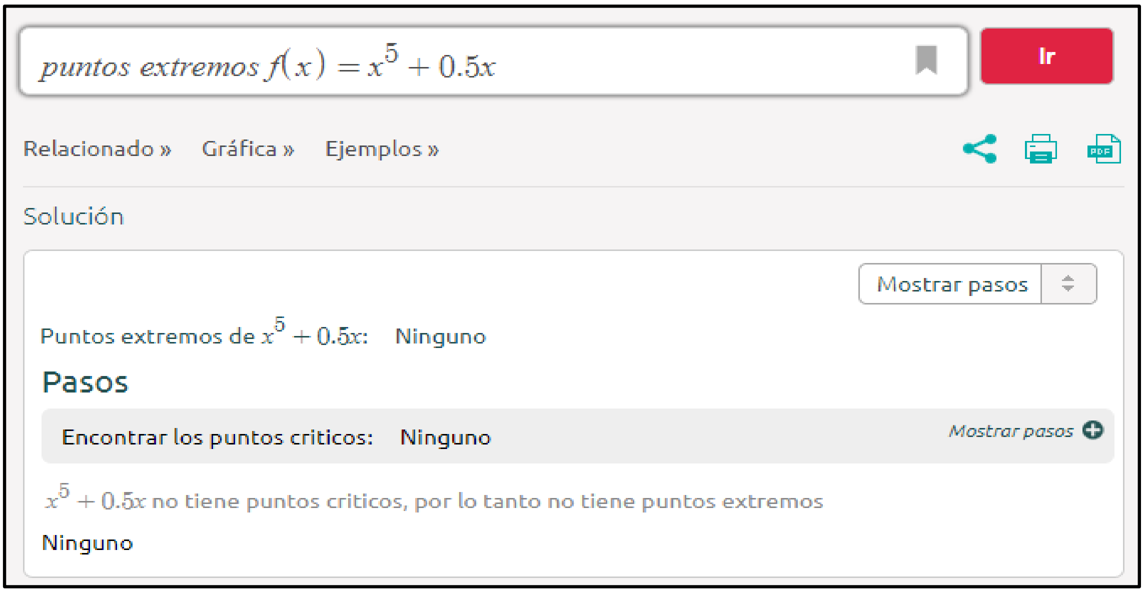


Figura N° 266 Función polinómica 23 Symbolab
Fuente: Symbolab

Funcion 38 $f_{(x)} = x^5 - 4x^4 + 4x^3$

Ubica un mínimo en (2;0) , un máximo en $\left(\frac{6}{5}; \frac{3456}{3125}\right)$ y un punto silla en (0;0) como ilustra la figura 267

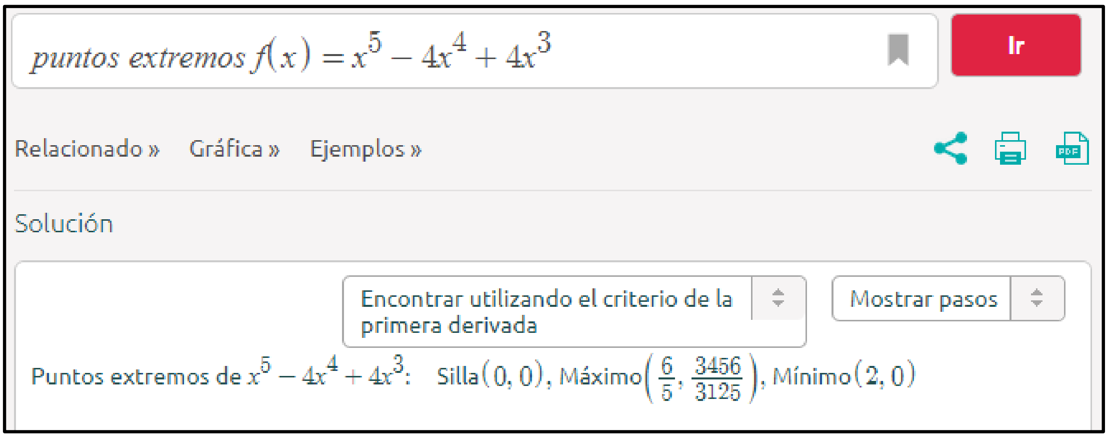


Figura N° 267 Función polinómica 24 Symbolab
Fuente: Symbolab

Funcion 39 $f_{(x)} = x^5 - 3x^4 + 3x^2$

Ubica un máximo cuando x = -0,629 y x = 0,892 asi como mínimos en x=0 y x = 2,137 como muestra la figura 268

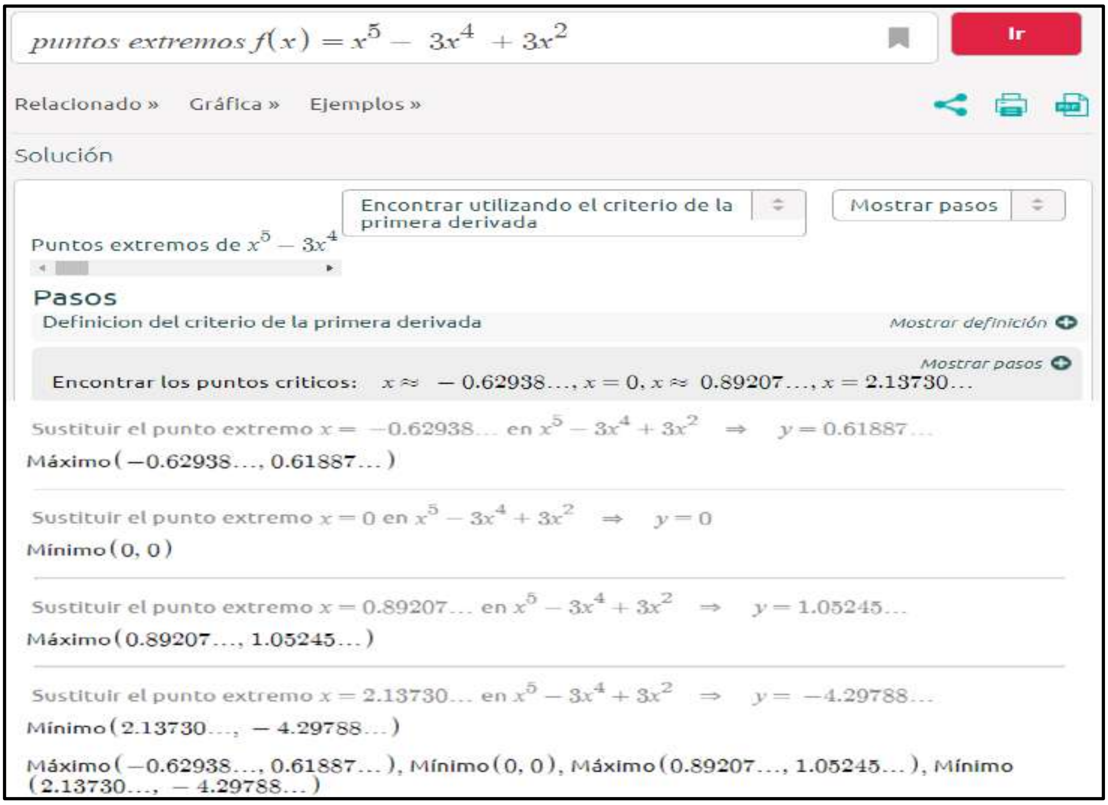


Figura N° 268 Función polinómica 25 Symbolab
Fuente: Symbolab

Funcion 40 $f_{(x)} = x^5 - 2x^4 - 2x$

Ubica un máximo en (-0,569;0,8689 y un mínimo en (1,683;-5,909) como muestra la figura 269

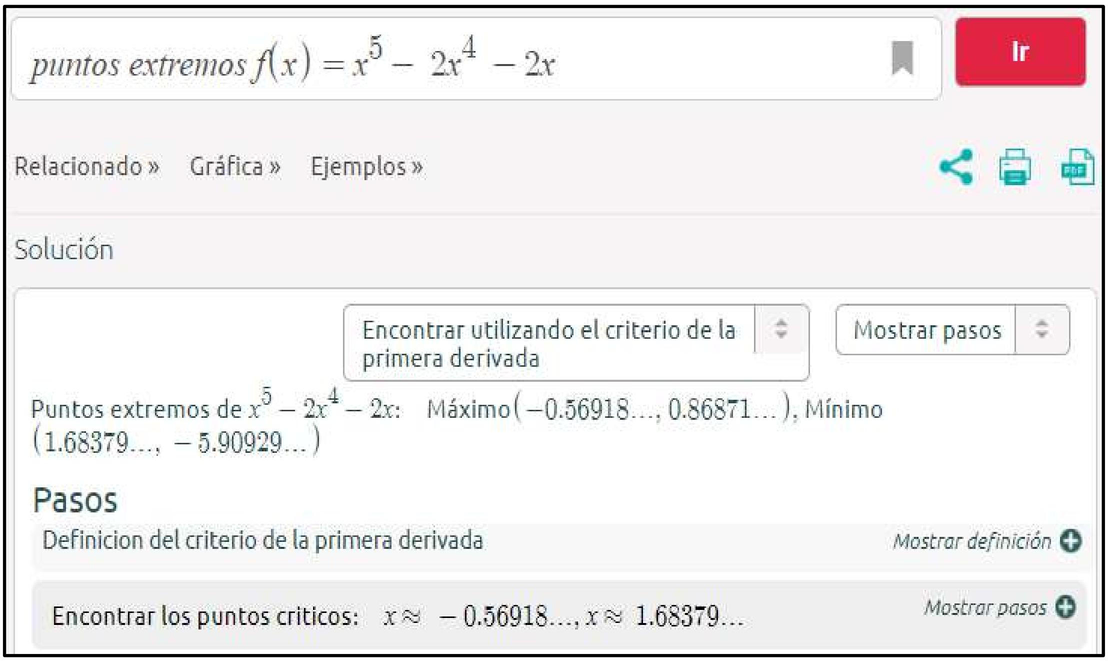


Figura N° 269 Función polinómica 26 Symbolab
Fuente: Symbolab

Funcion 41 $f_{(x)} = 0,2x^5 - 0,5x^4 - 1$

Ubica un máximo en (0;-1) ,un mínimo en (2;-2,6) como muestra la figura 2270

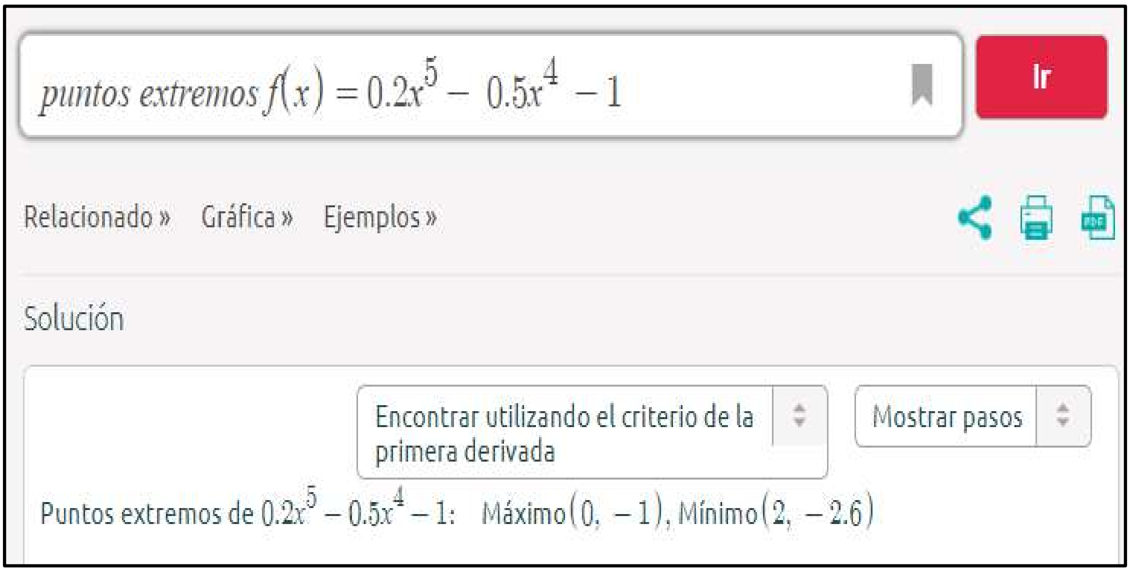


Figura N° 270 Función polinómica 27 Symbolab
Fuente: Symbolab

Función 42 $f_{(x)} = x^5 - 2x^3 - 3x^2$

Ubica un máximo en (0;0) ,un mínimo en (1,428;-6,00345) como muestra la figura 271

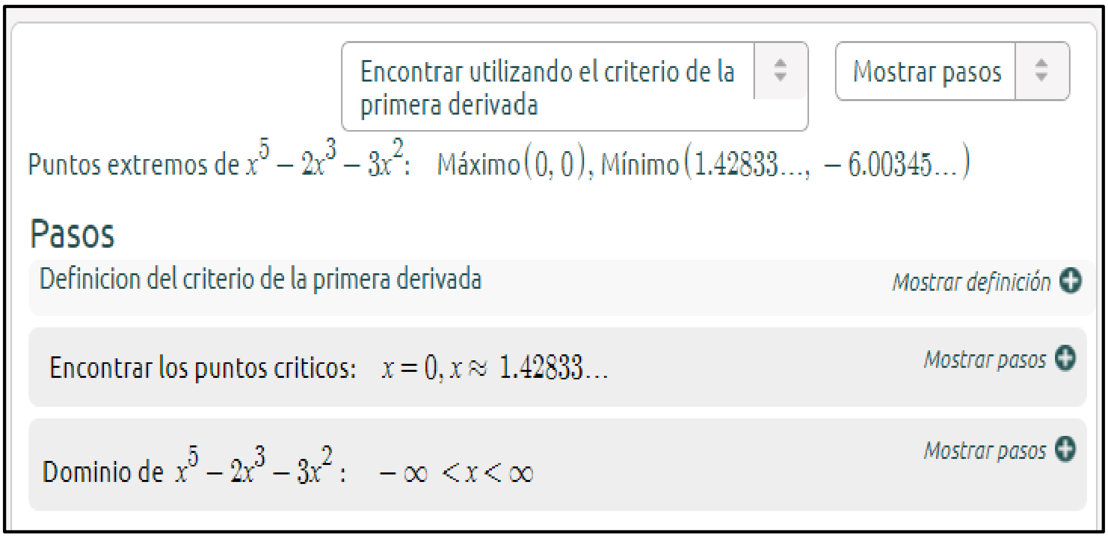


Figura N° 271 Función polinómica 28 Symbolab
Fuente: Symbolab

Funcion 43 $f_{(x)} = x^5 - 1.5x^3 + 1.2x^2$

Ubica un máximo en (-1,148;1,856) , un mínimo en (0;0) como muestra la figura 272

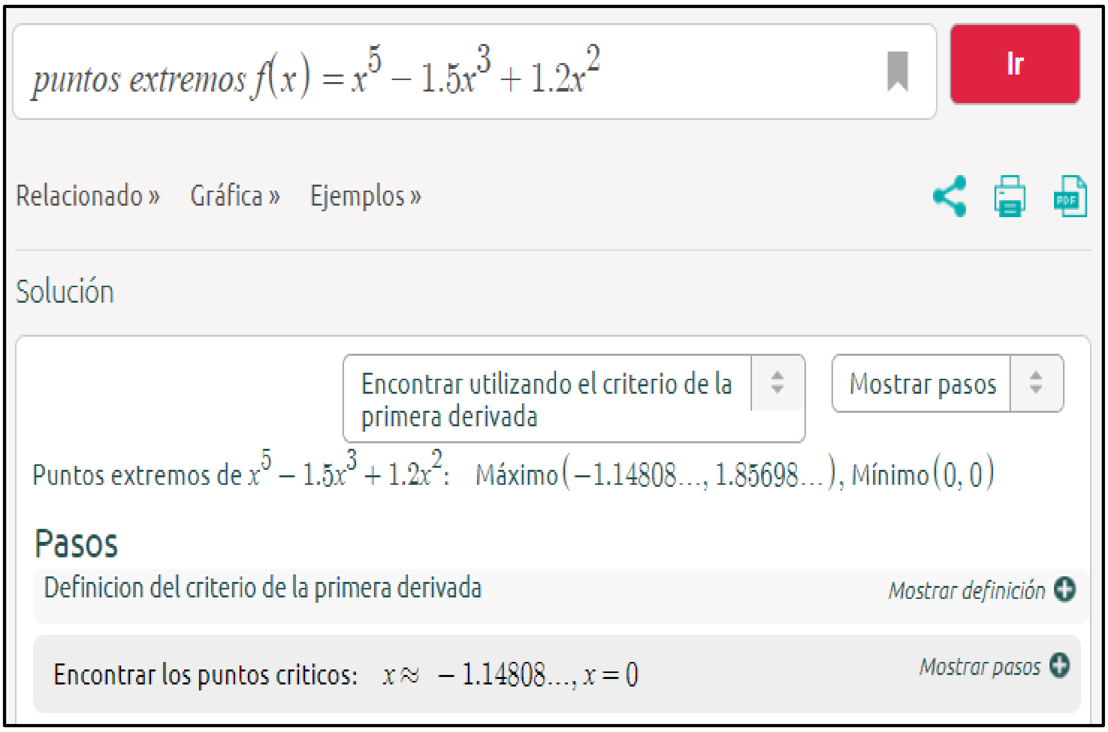


Figura N° 272 Función polinómica 29 Symbolab
Fuente: Symbolab

Funcion 44 $f_{(x)} = x^5 - 2x^3 + 2x$

No encuentra puntos extremos como muestra la figura 273

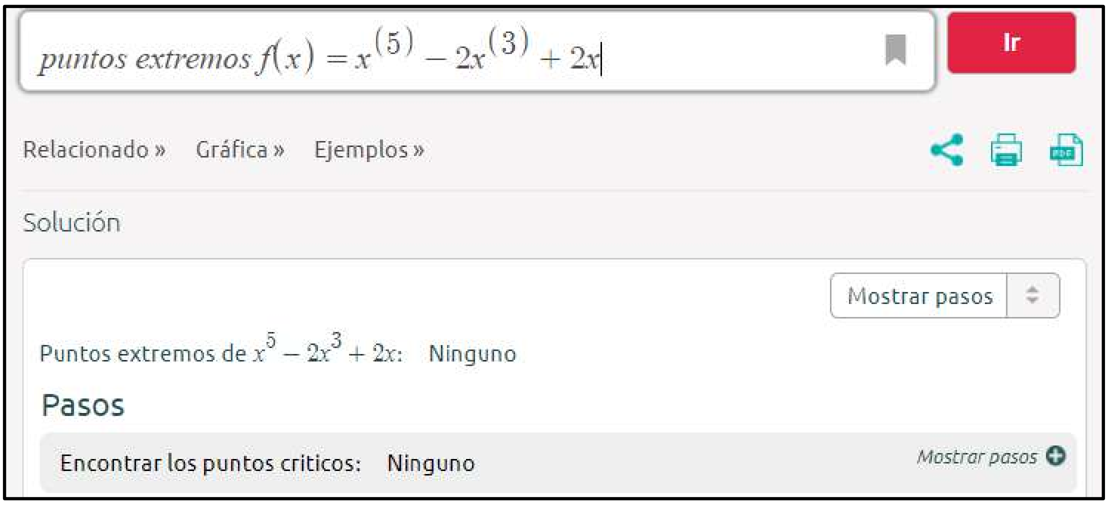


Figura N° 273 Función polinómica 30 Symbolab
Fuente: Symbolab

Funcion 45 $f_{(x)} = x^5 - 2x^3 + x$

Ubica dos máximos en $x = -1$, $x = \sqrt{\frac{1}{5}} = 0{,}447$ y dos mínimos en $x = 1$, $x = -\sqrt{\frac{1}{5}} =$

$-0{,}447$ como muestra la figura 274

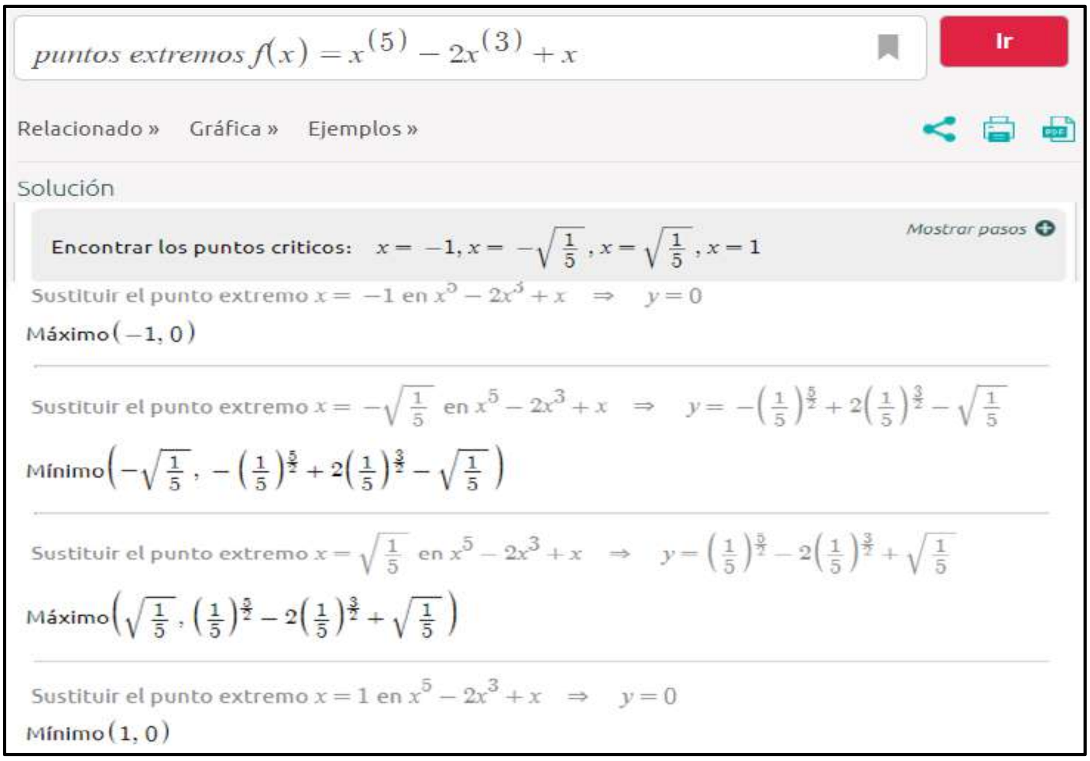


Figura N° 274 Función polinómica 31 Symbolab
Fuente: Symbolab

Funcion 46 $f_{(x)} = x^5 - x^3 + 1$

Ubica un máximo cuando $x = -\sqrt{\frac{3}{5}} = -0{,}7745$ y un mínimo cuando $x = \sqrt{\frac{3}{5}} = 0{,}7745$ y además señala un **punto silla en (0,1)** como indica la figura 275

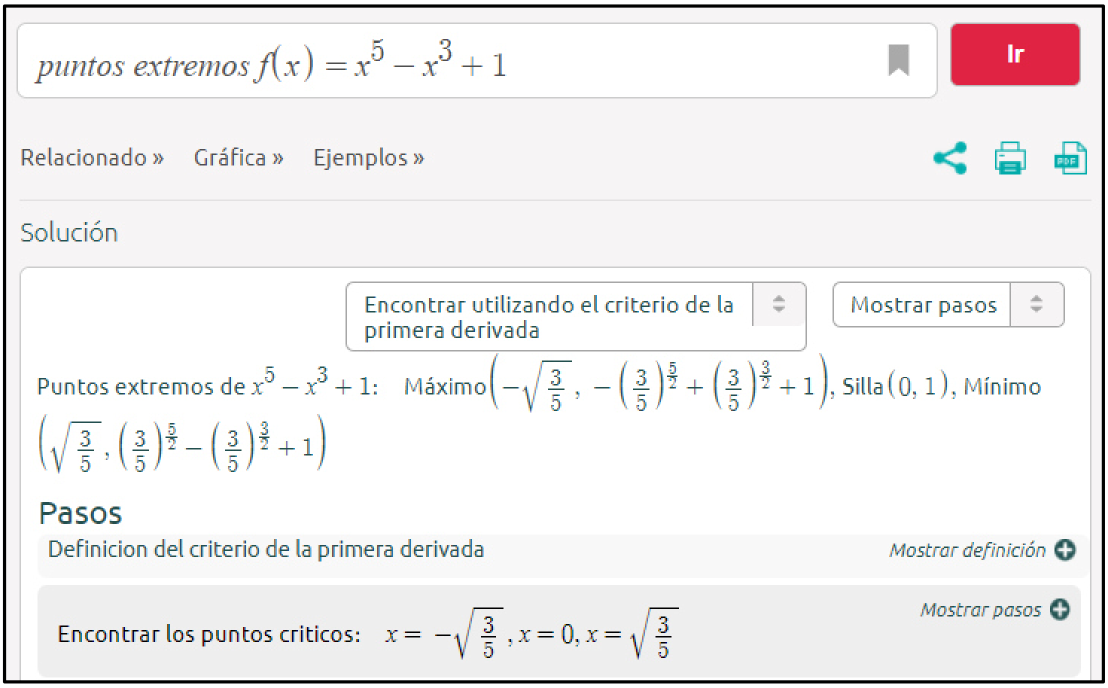


Figura N° 275 Función polinómica 32 Symbolab
Fuente: Symbolab

Funcion 47 $f_{(x)} = x^5 + x^2 - 2x$

Ubica un máximo en (-0,9383;2,0297) y un mínimo (0,62311;-0,76401) como se ilustra en la figura 276

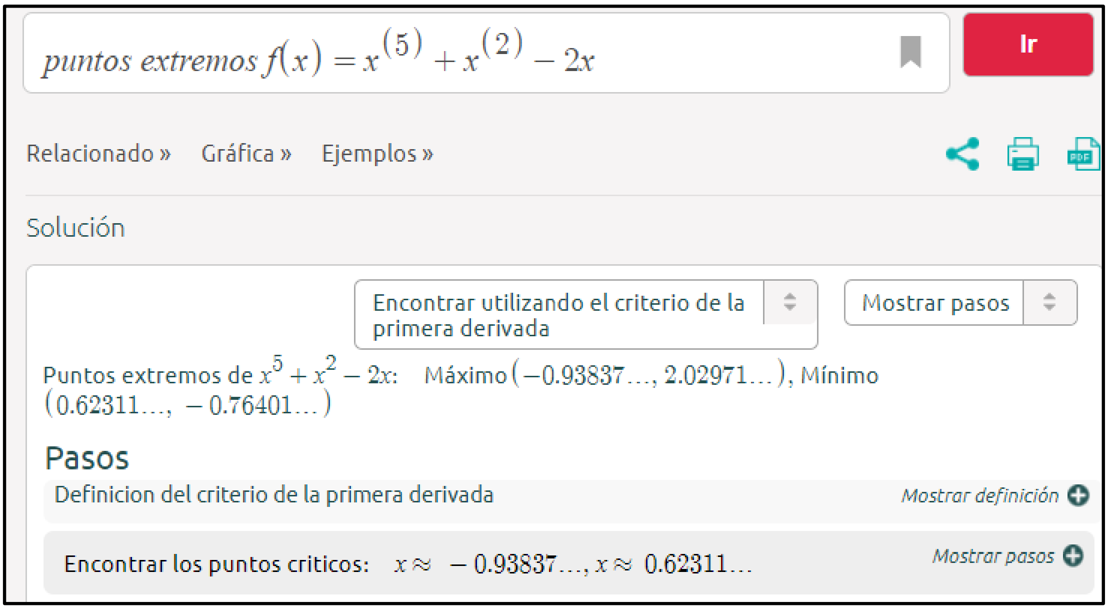


Figura N° 276 Función polinómica 33 Symbolab
Fuente: Symbolab

Funcion 48 $f_{(x)} = x^5 + x^2 - 2$

Ubica un mínimo en (0;-2) y un máximo en $x = -\sqrt[3]{\frac{2}{5}} = -0{,}7368$ como muestra la figura 277

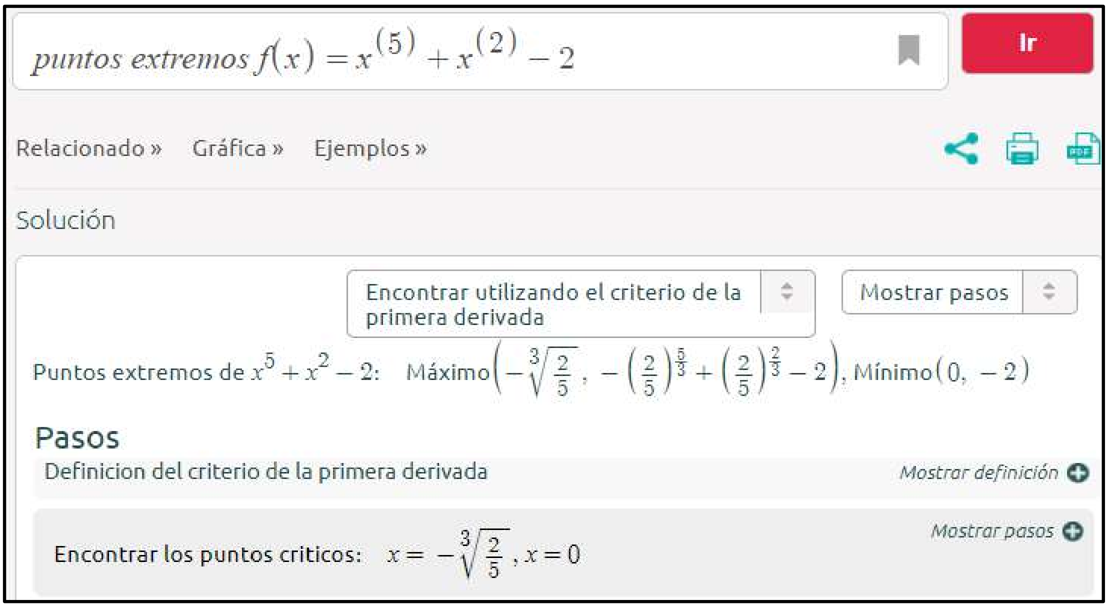


Figura N° 277 Función polinómica 34 Symbolab
Fuente: Symbolab

Funcion 49 $f_{(x)} = x^5 - 3x + 1$

Ubica un mínimo cuando $x = \sqrt[4]{\frac{3}{5}} = 0{,}88011$ y un máximo en $x = -\sqrt[4]{\frac{3}{5}} = -0{,}88011$ como muestra la figura 278

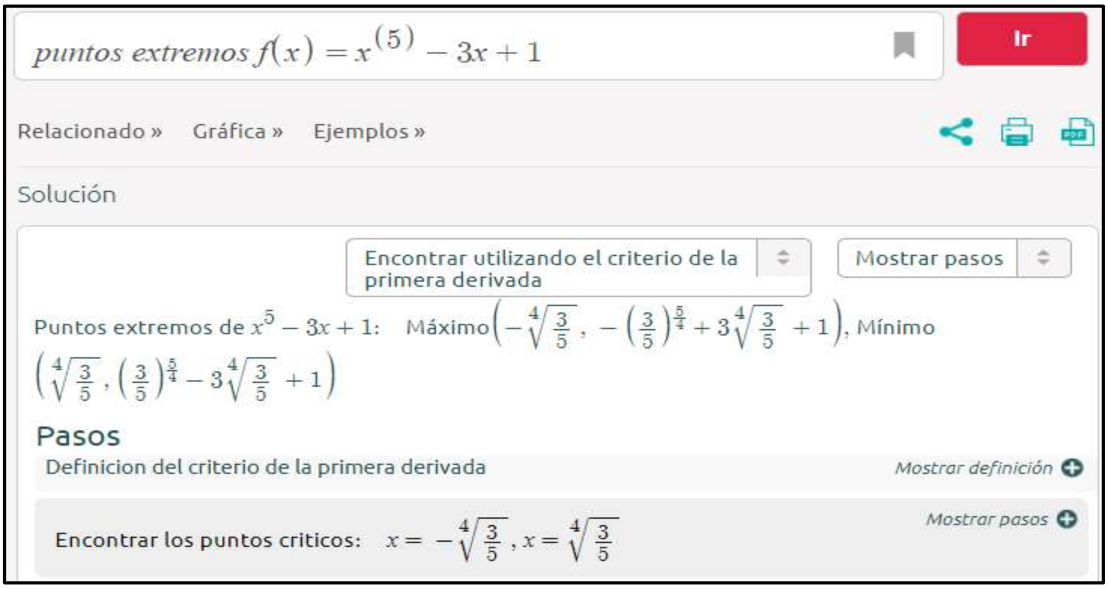


Figura N° 278 Función polinómica 35 Symbolab
Fuente: Symbolab

Funcion 50 $f_{(x)} = x^5 - x^4 + 2x^3 - 4x^2$

Ubica un máximo en (0;0) y un mínimo en (1,07234;-2,0378) como muestra la figura 279

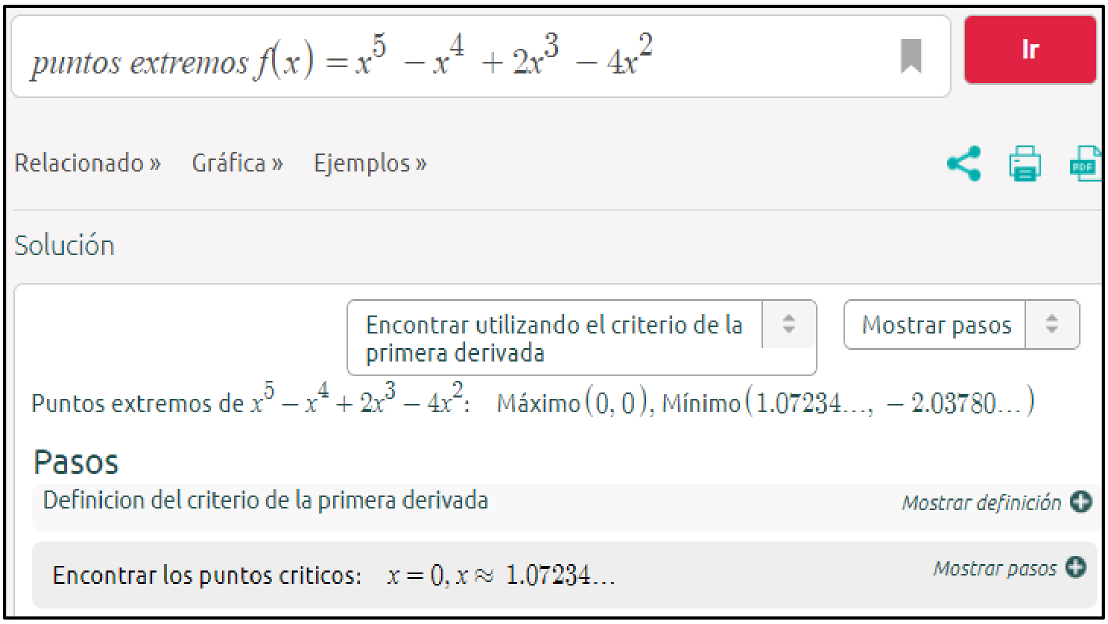


Figura N° 279 Función polinómica 36 Symbolab
Fuente: Symbolab

Funcion 51 $f_{(x)} = x^5 - x^4 + 0.5x^2 - 4x$

Symbolab : ubica un máximo en (-0,83895; 2,79672) y un mínimo en (1,16188; -3,67752) como muestra a figura 280

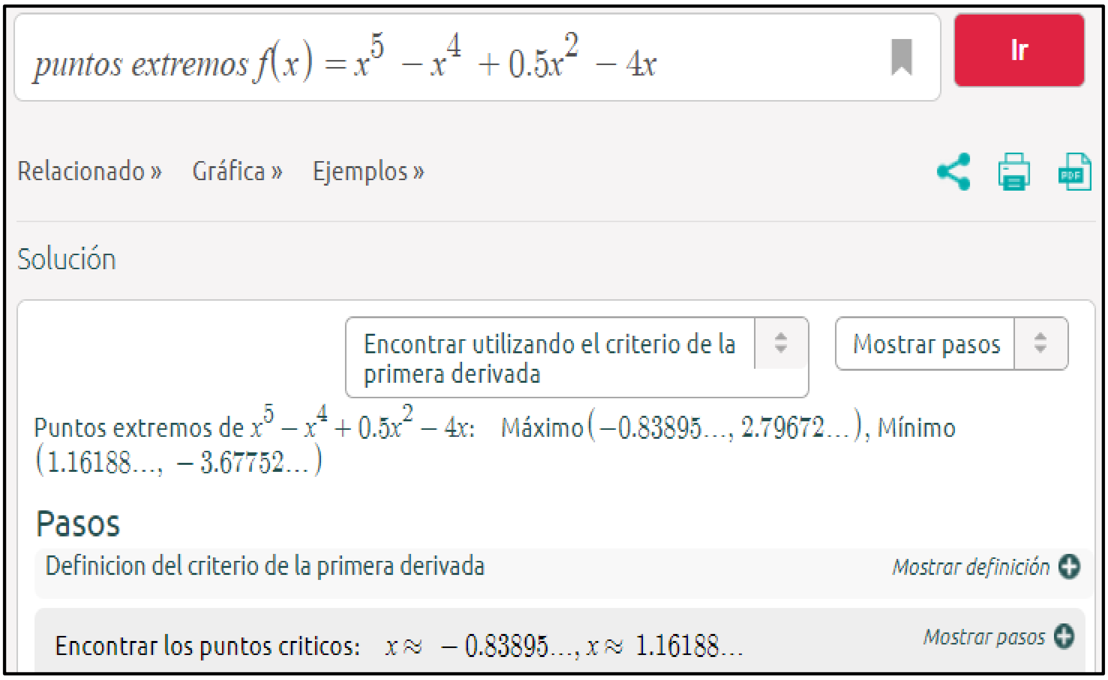


Figura N° 280 Función polinómica 37 Symbolab
Fuente: Symbolab

Funcion 52 $f_{(x)} = x^5 - x^4 - 2x^3 - x$

Ubica un máximo en (-0,8736; 1,11576), un mínimo en (1,59924; -5,8598) como muestra la figura 281

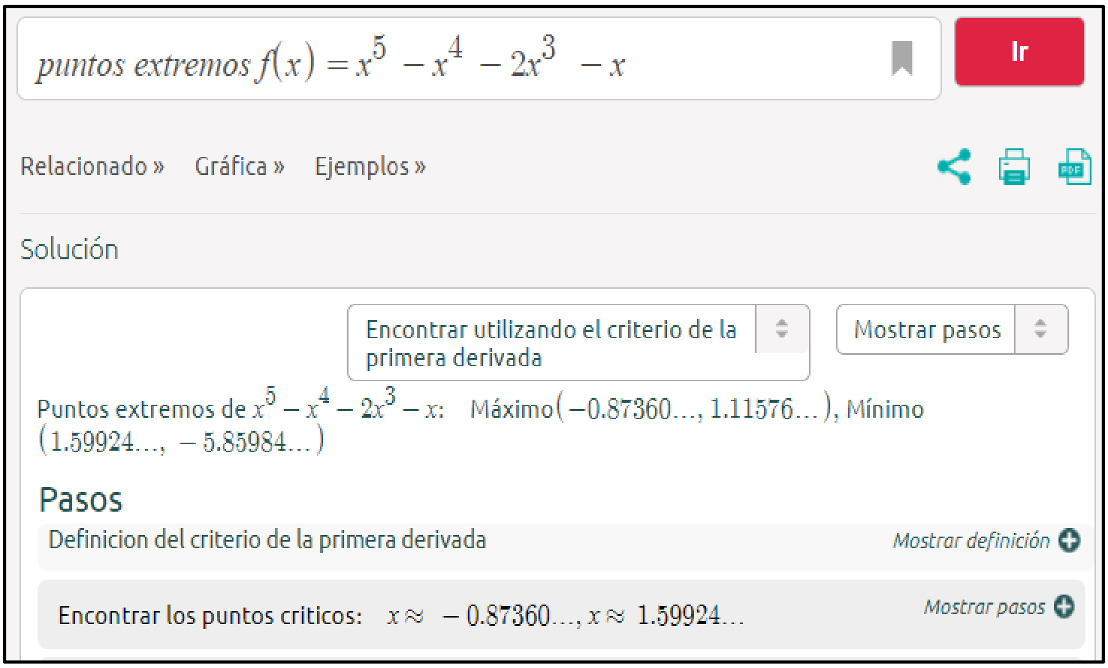


Figura N° 281 Función polinómica 38 Symbolab
Fuente: Symbolab

Funcion 53 $f_{(x)} = 0.2x^5 - 0.25x^4 - x^2 - 2x$

Ubica un mínimo en (1,8737; -5,72068) y un máximo en (-0,70347; 0,81639) como muestra la figura 282

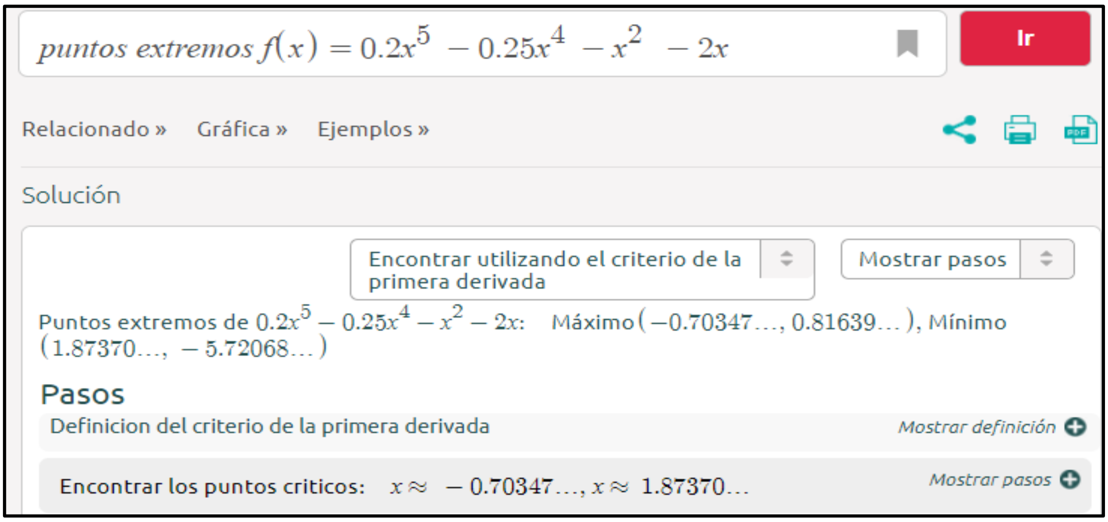


Figura N° 282 Función polinómica 39 Symbolab
Fuente: Symbolab

Funcion 54 $f_{(x)} = 0.4x^5 - 2x^3 - 2x^2 - x$

Ubica un máximo en (-1,33509; 0,83293) y un mínimo en (2,02692; -13,2136) como ilustra la figura 283

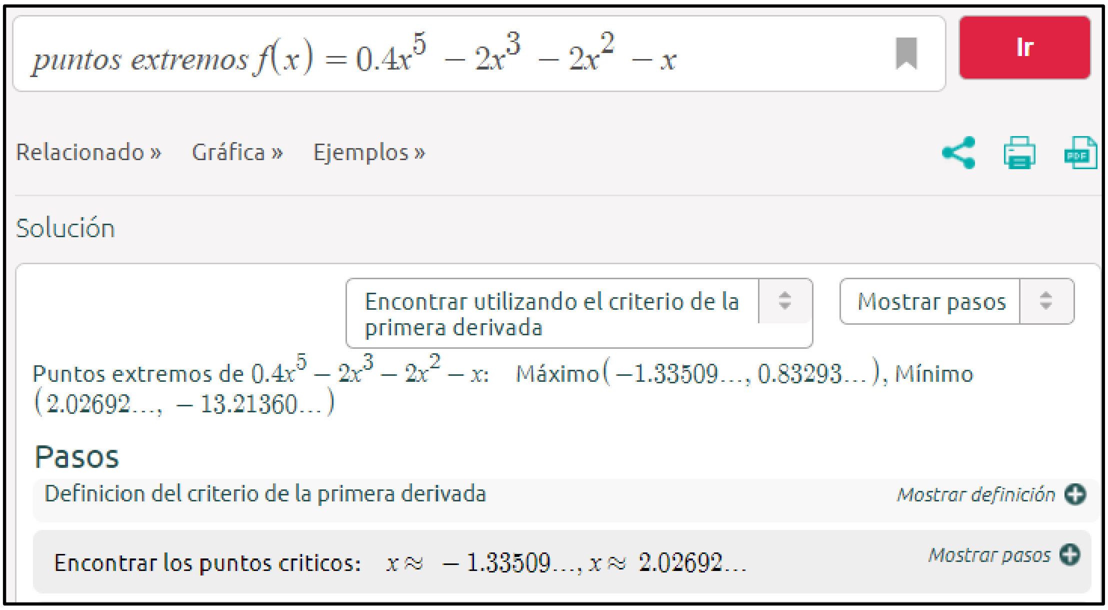


Figura N° 283 Función polinómica 40 Symbolab
Fuente: Symbolab

Funcion 55 $f_{(x)} = f_{(x)} = 0.2x^5 - 0.25x^4 - x^3 - 0.5x^2 - 1$

Ubica máximos en (-1; -0,95) y (0; -1) , mínimos en (-0,4142; -1,0245) , y (2,4142; -10,07548) como indica la figura 284

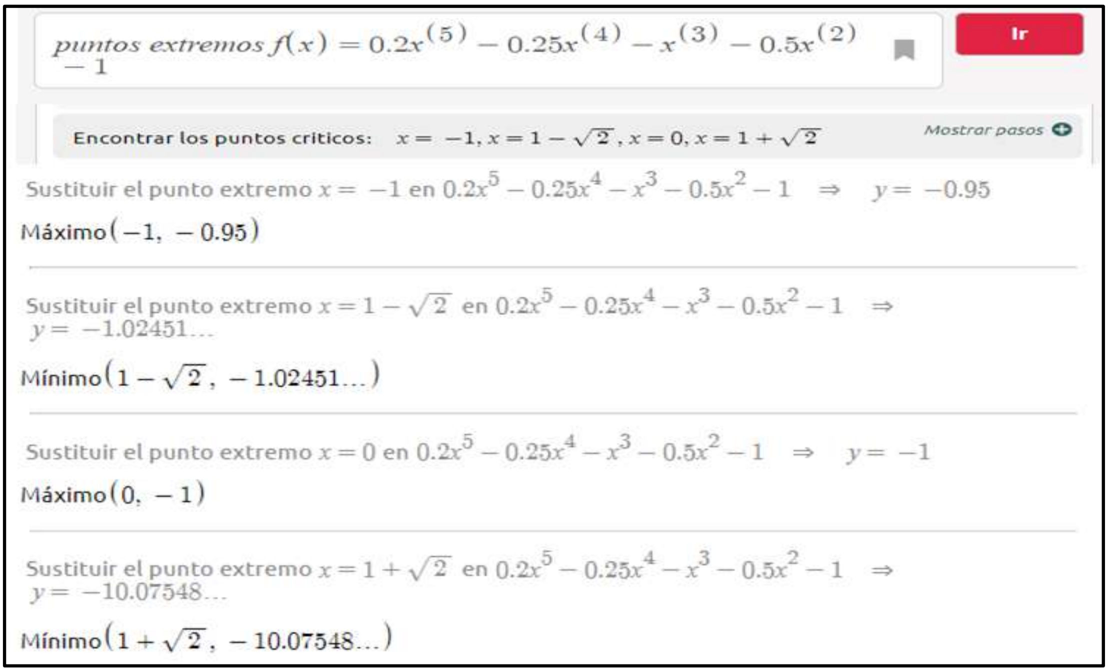


Figura N° 284 Función polinómica 41 Symbolab
Fuente: Symbolab

Funcion 56 $f_{(x)} = 0.2x^5 - x^3 - 0.5x^2 + 2x - 1$

Ubica un máximo en (0,71831; -0,15374) y un mínimo en (1,70211; - 1,1183) como ilustra la figura 285

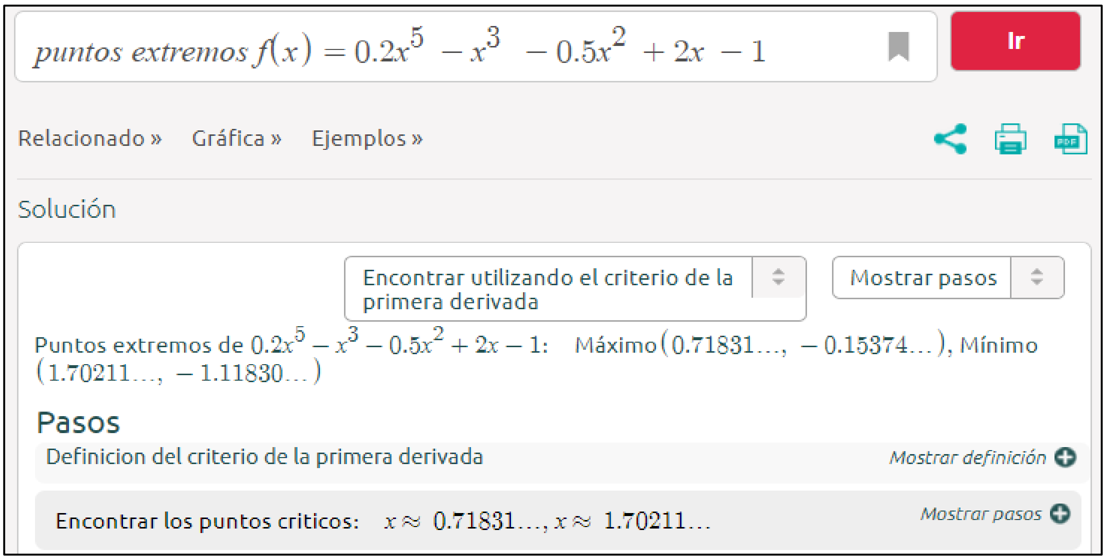


Figura N° 285 Función polinómica 42 Symbolab
Fuente: Symbolab

Funcion 57 $f_{(x)} = 0.2x^5 - 0.5x^4 - x^3 + 0.25x^2 + 1.5x - 1$

Ubica un máximo en (0,68736; -0,25652) y un mínimo en (2,9108;-13,28004) como indica la figura 286

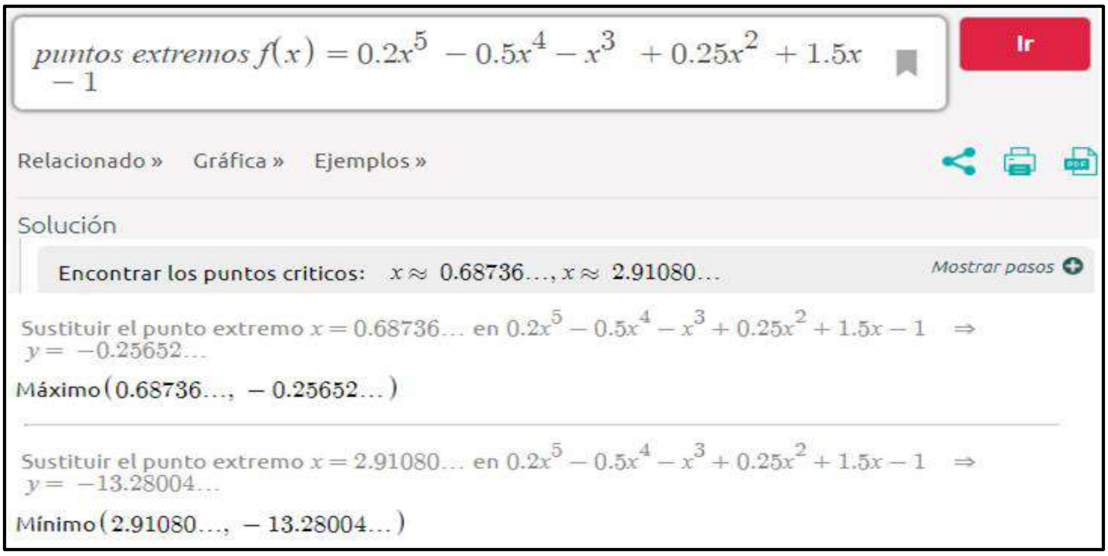


Figura N° 286 Función polinómica 43 Symbolab
Fuente: GeoGebra

4.7.6 Análisis de Resultados

La verificación del software DERIN se ha contrastado con otros software online: WolframAlpha, GeoGebra y Symbolab que en un analisis por grupo funcional se visualiza en el siguiente cuadro

Cuadro N 5 : Versus de software

Tipo de funcion	**Cuadrática**	**Cúbica**	**Grado4**	**Grado 5**
DERIN	Aceptable	Aceptable	Aceptable	Aceptable
GEogebra	Aceptable	Aceptable	Aceptable	En la función 33 se observó resultados donde el máximo ocurre en x= -0,000000167, sin embargo DERIN lo ubica en x=0 concordante con otros software. Igual situacion se presentó al evaluar la funcion 41
Wolfram Alpha	Aceptable	Aceptable	Al usar la funcion 19 no alcanza el valor minimo. En la funcion 20 no alcanza uno de los mínimos relativos y en las funciones 30 y 31 no ubica uno de los máximos	
Symbolab	Aceptable	En la funcion 6 se usó la primera derivada como criterio y estableció un punto silla (un punto que no es máximo ni mínimo) y al usar el criterio de la segunda derivada planteó que es poco concluyente	En las funciones 19 y 20 presentó los resultados en forma de radicales lo que hizo necesario usar una calcudora para calcular los valores,	

Fuente: Elaboración propia

El cuadro refleja que la potencialidad de DERIN radica en mejora de aproximacion de resultados y la forma en que lo presenta así como una forma única para obtener los resultados sin tener que elegir entre dos criterios para estar seguro de un resutlado

Para mostrar la bondad de DERIN se realiza un benchmarking como insumo futuro para mejoras de este.

Al respecto , sobre el benchmarking, *Rodríguez de Rivera dice:* “E*s un método para ayudar en la planificación y desarrollo de productos, servicios o sistemas que sistematiza la medición/evaluación de los niveles de las prestaciones técnicas o de calidad alcanzados en la firma propia en comparación con los resultados de los mejores competidores - en referencia a determinadas magnitudes que deben definirse como las más relevantes” (como es citado por Arianne de Cárdenas Cristia,* El Benchmarking como herramienta de evaluación)

Sobre el esquema de puntuación se ha considerado la calificación que han utilizado en la investigación “Análisis comparativo de software matemático para la formación de competencias de aprendizaje en cálculo diferencial” donde un nivel alto tiene una valoración de 5 puntos, un nivel medio tiene valoración de 3 puntos y un bajo nivel solo se valora con 1 punto.

En lo referente a los ítems seleccionados para calificación se tiene en cuenta los aspectos de la parte técnica, usabilidad y la interfaz

Cuadro 5: Comparación de Software

ANÁLISIS COMPARATIVO DE PRODUCTOS DE SOFTWARE				
Software / Indicadores	Geogebra	Wolfram Alpha	Symbolab	DERIN
Requerimientos técnicos				
Software orientado a web	5	5	5	5
Compatible con cualquier sistema operativo	5	5	5	5
Compatible con cualquier dispositivo de acceso a internet	5	5	5	5
Posee otras formas de presentacion	5	1	1	1
	20	**16**	**16**	**16**
Usabilidad				
Presenta resultados de extremos relativos y puntos de inflexión	3	1	3	5
Los resultados se presentan con facilidad, sin exceso de comandos	1	5	5	5
Se aplica a cualquier tipo de funcion	5	5	5	1
Los resultados son confiables	5	3	5	3
Las aproximaciones de los valores calculados son adecuadas	3	5	5	5
	17	**18**	**23**	**19**
Interfaz				
La presentación es amigable para el usuario	5	5	5	3
El ingreso de los datos es sencillo	3	3	3	5
	8	**8**	**8**	**8**
Total	**45**	**42**	**47**	**43**
Leyenda de funcionalidad				
5=Alto				
3=Medio				
1= Bajo				

Fuente: Elaboración propia

El diseño del cuadro de comparación requiere de ajustes y de la opinión de expertos y de usuarios para aumentar la credibilidad, sin embargo se puede deducir por las pruebas efectuadas que en lo particular determina los extremos relativos y los puntos de inflexión en forma simultánea a diferencia de los demás software.

En algunos casos mejora la aproximación de GeoGebra, calcula extremos reltivos que Wolfram no ubica y distingue el concepto de punto de inflexión frente al de punto silla que utiliza Symbolab.

CAPÍTULO V

CONCLUSIONES

1. El desarrollo del nuevo software DERIN utiliza el criterio de la derivada enésima describiendo un nuevo camino para obtener los máximos y mínimos relativos de una función polinómica lo que se ha reflejado en las pruebas con las funciones testigo y otros software usados para la verificación de resultados siendo aceptable su aplicacion.

2. El desarrollo del nuevo software DERIN utiliza el criterio de la derivada enésima describiendo un nuevo camino para obtener los puntos de inflexión relativos de una función polinómica lo que se ha reflejado en las pruebas con las funciones testigo y otros software usados para la verificación de resultados.

3. Utilizar el nuevo software DERIN como recurso de apoyo para la optimización de funciones, trazado de gráficas, así como la enseñanza y transferencia de conocimientos en los cursos de cálculo diferencial es viable en virtud a las bondades que el software presenta como son la mejor aproximacion de los resultados y la presentación de los mismos.

4. Las pruebas han permitido verificar que en WolframAlpha no determina en algunos casos el máximo o mínimo relativos de una función. Geoebrra tiene alguna dificultad con las aproximaciones y Symbolab presenta las soluciones en un formato que hace necesario usar una calculadora para aproximarlo en decimales. En consecuencia, DERIN tiene la versatilidad para afrontar estas situaciones, generando la oportunidad de contribuir en este proceso de mejora con nuevos algoritmos o de replantear los existentes en virtud de lo enfatizado por Dijkstra.

RECOMENDACIONES

1. El software DERIN no es ajeno a las dificultades que posee el software orientado al cálculo por lo que la revisión de los productos y sus algoritmos deben ser materia de una revisión constante. En particular DERIN está limitado por ahora a funciones polinómicas de grado menor a 6, aunque en algunas funciones de mayor grado alcanza los valores mostrando resultados que algunos no muestran como es el caso de la función $f(x) = x^8 - 8x^6$

2. Las raíces de una ecuación son un tema tangencial en el desarrollo del software DERIN, pero al mismo tiempo ha servido para involucrarme con un método como Graeffe (raíces de un polinomio), el cual tiene una gran virtud pues no requiere valor inicial para la búsqueda de raíces a diferencia de otros métodos, pero tiene como dificultad la desviación de la convergencia de la raíz. Revisando la literatura sobre el tema encontré estudios de Juan Babini, Rey Pastor que se plantean de transformar el método y mejorarlo.

TRABAJOS FUTUROS

1. Resulta importante insistir en las investigaciones futuras de modo que podamos integrar el software DERIN con la potencialidad de WolframAlpha o algún software afín con el objetivo de compartir nuestros avances e integrarlos con los mencionados permitiendo la distribución de productos libres y cuya difusión contribuya al a la mejora de procesos de enseñanza al convertirse en un recurso para. Este fin.

2. Es propósito de la investigación innovar procesos de enseñanza donde el recurso tecnológico tenga mayor incidencia. DERIN desea ser parte de esta alternativa por tal motivo se ha creado orientado a web para que sea accesible desde cualquier dispositivo conectado a la red sin necesidad de actualizaciones, lo que requiere es un proceso de mejora donde podríamos incluir la idea que se convierta en un software inteligente es decir que aprenda a medida que reconoce nuevas funciones.

REFERENCIAS BIBLIOGRÁFICAS

[1] ACM. (Association for Computing Machinary -Dgital Library).

[2] ASÍS LÓPEZ, Efracio. 2010 métodos *Numéricos con Matlab*. 1ra Edición. Fondo Editorial Universidad de Ciencias y Humanidades. ISBN: 978-612-45707-2-8

[3] BAPTISTA LUCIO, P.; FERNANDEZ-COLLADO, C. y HERNANDEZ SAMPIERI, R. 2006 Metodología de la Investigación. Editorial Mc Graw Hill

[4] BABINI, José Sobre la Transformación del Método de Graeffe en Homenaje a Julio Rey Pastor Instituto de Matemática:1945-1946 Vol 1 Rosario - Argentina Consultado en http://dialnet.unirioja.es/servlet/libro? codigo=403555 el 20 de Noviembre de 2013

[5] CARAPAZ CARCANQUI, José (2014) Utilización del software MATLAB como herramienta didáctica en el aprendizaje de la matemática de los estudiantes de quinto semestre de as carreras de física y mamética de la FECYT. Tesis de Grado. Consultado en http://repositorio.utn.edu.ec/bitstream/123456789/4155/1/05%20FECYT%202086%20TESIS.pdf el 15 de noviembre de 2017

[6] CORTEZ ZAVALA, J. (2002) Software para la enseñanza de la derivada. Consultado en http://mattec.matedu.cinvestav.mx/el_calculo/data/docs/Dc2l3taQW10.pdf el 15 de noviembre de 2013.

[7] CUEVAS VALLEJO, C. A. & PLUVINAGE, F. Cálculo y Tecnología, El Cálculo y su Enseñanza © 2009 Cinvestav del Instituto Politécnico Nacional, México D.F. Consultado en http://mattec.matedu.cinvestav.mx/el_calculo/data/docs/Dc2l3taQW10.pdf el 15 de noviembre de 2013.

[8] CHAPRA, Steven & CANALE , Raymond 2012. Métodos Numéricos para Ingenieros 6ta Edición. Editorial Mc Graw Hill

[9] DÁVILA ARAIZA, Maria Teresa 2010. *La Derivada a Partir de Problemas de Optimización en Ambientes Dinámicos Creados con GeoGebra.* Tesis de Maestría Universidad de Sonora – México En http://www.repositorioinstitucional.uson.mx/bitstream/handle/unison/885/davilaaraizamariateresam.pdf?sequence=1&isAllowed=y consultado el 10 octubre 2013

[10] ESPINOZA RAMOS, Eduardo 2002. Análisis Matemático I. Editorial Servicios Gráficos.

[11] GRANVILLE, William Anthony. 2009 Cálculo Diferencial e Integral novena edición. Editorial Limusa – México

[12] GALVEZ OBISPO, Y.; JULCA CARPIO, R. y TINEO FLORES, R. (2017) Uso del software Wofram y su efecto en el nivel de aprendizaje de los sistemas de ecuaciones lineales en estudiantes del tercer gado de secundaria del colegio experimental de Aplicación La Cantuta. En http://repositorio.une.edu.pe/bitstream/handle/UNE/1331/2.pdf?sequence=1&isAllowed=y . Consultado el 05 de julio 2018

[13] HAEUSSLER, Ernest 2000 *Matemáticas para Administración, Economía, CC.SS. y la Vida*, 10ma. Edición. Prentice Hall México

[14] Institute of Electrical and Electronics Engineers IEEE 2017 versión 1.0

[15] KONG, Maynard 2001 Cálculo Cuarta Edición. PUCP Fondo Editorial

[16] LARSON, Ron y EDWARD, Bruce. 2010 Cálculo I, novena edición Editorial Mc Graw Hill.

[17] LÁZARO CARRIÓN, Moisés. 2002 Cálculo Diferencial I, Editorial Moshera, Lima - Perú

[18] LEITHOLD , Louis , 1999. El Cálculo . 7ma Edición –Mexico

[19] MITACC, Máximo y TORO, Luis. (2009) Cálculo Volumen I 3ra Edicion

[20] MORENO MORENO, María del Mar, 2005 . El Papel de la Didáctica en la enseñanza del Cálculo: evolución, estado actual y retos futuros. En http://funes.uniandes.edu.co/1325/1/Gonzalez2005El_SEIEM_81.pdf Consultado el 25 de noviembre del 2013

[21] MOSQUERA RIOS, M. y VIVAS IDROBO, S. 2017. Analisis comparativo de software matemático para la formación de competencias de aprendizaje en cálculo diferencial. Consultado en https://dialnet.unirioja.es/servlet/articulo?codigo=6157572 el 08 de agosto del 2018

[22] NAKAMURA, Schoichiro, 1992. Métodos Numéricos Aplicados con Software, Edit. Prentice Hall - México

[23] PISKUNOV , N. 1983 . Cálculo Diferencial e Integral Tomo I 3ra Edicion Editorial MIR – Moscú

[24] PRESSMAN, R (2011) Ingenieria del Software . Cuarta edición. Editorial Mc Graw Hill

[25] PURCELL, E., VARBEG , D. y RIGDON, S. (2007). Cálculo Diferencial e Integral. Novena edición. Editorial Pearson

[26] QUINTANA SANCHEZ, Diana. (2010) Tratamiento didáctico de la derivada-Aplcaicion del program Derive. Tesis de Maestria. Consultado en https://pirhua.udep.edu.pe/bitstream/handle/11042/1411/MAE_EDUC_072.pdf?sequence=1&isAllowed=y el 04 -08 -2015

[27] RIVEROS VASQUEZ , Daniel 2015 “Aplicación de la investigación de operaciones al problema de la distribución a una empresa logística” . Tesis de grado. Consultado en http://cybertesis.unmsm.edu.pe/bitstream/handle/cybertesis/4365/Riveros_vd.pdf?sequence=1&isAllowed=y el 05 de Febrero 2019

[28] RUIZ LIZAMA, Edgar (2006) IntegraLAB un software para la integración de funciones y solución de ecuaciones diferenciales por métodos numéricos Consutado el 04-08-2011 en

http://tesis.pucp.edu.pe/repositorio/bitstream/handle/20.500.12404/132/RUIZ_EDGAR_INTEGRALAB_SOFTWARE_INTEGRACION.pdf?sequence=1&isAllowed=y

[29] STEWART, James. Calculo de una variable Trascendentes tempranas 7ma Edición Cengage Learning.

[30] TOLA PASQUEL, José Análisis I 1970 Biblioteca de la PUCP

[31] VENERO, Armando 2007. Análisis Matemático I, Ediciones Gemar, Lima -Perú

[32] ZILL, Denis y WRIGTH, Warren. 2011 Cálculo Diferencial, Cuarta edición editorial Mc Graw Hill

ANEXO 1: GLOSARIO DE TÉRMINOS

Android	: Sistema operativo basado en el núcleo de Linux, diseñado para aplicaciones móviles
Ant	: Software para procesos de automatización de compilación, desarrollado en lenguaje Java y requiere la plataforma Java,
API	: Interfaz de Programación de Aplicaciones, abreviada como API (del inglés: Application Programming Interface)
Derivada Enésima	: Proceso de derivación iterativa
DERIVE	: Software para matemática.
GeoGebra	: Software aplicativo en matemática
MATLAB	: Lenguaje de alto nivel elaborado por Mathworks
GUI	: Interfaz Gráfica de Usuario
GUIDE	: Entorno de desarrollo de GUI
GNU	: La Licencia Pública General de GNU o más conocida por su nombre en inglés GNU General Public License
HTML	: HyperText Markup Language , hace referencia al lenguaje de marcado para la elaboración de páginas Web
IDE	: Entorno de desarrollo integrado
JAVA	: Lenguaje de Programación
LINUX	: Sistema operativo, es un conjunto de programas que le permiten interactuar con su ordenador y ejecutar otros programas.
PI1E	: Punto de inflexión de primera especie.
PI2E	: Punto de Inflexión de segunda especie.
PHP	: Hypertext Pre-processor.
VHDL	: VHDL es un lenguaje definido por el IEEE (Institute of Electrical and Electronics Engineers) (ANSI/IEEE 1076-1993) usado por ingenieros y científicos
WolframAlpha	: Plataforma matemática online.

Anexo 2

Código Fuente Método Graeffe

Compartimos el código del método numérico para determinar las raíces de un polinomio, destacando que es código libre y se utiliza siendo extraído desde http://www.sc.ehu.es/sbweb/fisica_/numerico/raices/graeffe.html al mismo que se ha incorporado código adicional para mejora en el cálculo de raíces.

```
package matematicaderivacion;
/**
 *
 * Extraido
 *
 */
public class Graeffe {
   public int n;
   public double[] raicesReales;
   public Complejo[] raicesComplejas=new Complejo[4];
   public static int numReales;
   public static int numComplejas;
   public double[][] a;
   private int pot2=1;
   private int m;
   private final int MAX_ITER=5;
   private static final double CERO=0.0001;
   private double[] moduloComplejas=new double[2];

   public Graeffe(double[] coef) {
      n=coef.length-1;
      raicesReales=new double[n];
      a= new double[MAX_ITER][n+1];
//la primera fila de la tabla guarda los coeficientes del polinomio
      for(int j=0; j<n+1; j++){
         a[0][j]=coef[j];
      }
      for(int m=1; m<MAX_ITER; m++){
         for(int j=0; j<n+1; j++){
            a[m][j]=0.0;
         }
      }
      numReales=0;
      numComplejas=0;
   }
   private void tabla(){
      int k,l, signo;
//divide el polinomio por el primer coeficiente, las raíces no cambian
      for(int i=1; i<n+1; i++){
         a[0][i]/=a[0][0];
      }
      a[0][0]=1.0;
      m=1;
exterior:
      do{
//cuadrados
         for(int i=0; i<n+1; i++){
            a[m][i]=a[m-1][i]*a[m-1][i];
            if(Double.isInfinite(a[m][i])){
               break exterior;
            }
         }
//dobles productos
         for(int i=1; i<n; i++){
            for(int s=1; s<n/2+1; s++){
               k=i-s;     l=i+s;
               if((k<0)||(l>n))   break;   //términos simétricos
               signo=(s%2==0)? +1: -1;
```

```
                a[m][i]+=signo*2*a[m-1][k]*a[m-1][l];
                if(Double.isInfinite(a[m][i])){
                    break exterior;
                }
            }
        }
        m++;
    }while(m<MAX_ITER);
//--------------------------------------------------
    m--;
//potencia de m de 2
    pot2=1;
    for(int i=1; i<=m; i++){
        pot2*=2;
    }
  }
//valor de un polinomio para una variable real
  public double valorPolinomio(double x){
    double y=0.0;
//sucesivas potencias de x, se puede utilizar también la funcion Math.pow
    double[] pot_x=new double[n+1];
    pot_x[0]=1.0;
    for(int i=1; i<n+1; i++){
        pot_x[i]=pot_x[i-1]*x;
    }
//valores de los sucesivos términos
    for(int i=0; i<n+1; i++){
        y+=a[0][i]*pot_x[n-i];
    }
    return y;
  }
  public Complejo valorPolinomio(Complejo x){
    Complejo y=new Complejo();
    for(int i=0; i<n+1; i++){
        y=Complejo.suma(y, Complejo.producto(a[0][i],
          Complejo.potencia(x, (n-i))));
    }
    return y;
  }

  private void raizRealSimple(int j){
//valor absoluto de la raíz
   // System.out.println("Raiz simple");
    double logaritmo=(Math.log(Math.abs(a[m][j]))-Math.log(Math.abs(a[m][j-1])))/pot2;
    double raiz=Math.exp(logaritmo);
//determinación del signo, y1 o y2 tienen que ser casi cero
raicesReales[numReales]=(Math.abs(valorPolinomio(raiz))<Math.abs(valorPolinomio(-raiz)))?  raiz  :  -
raiz;
    numReales++;
  }
 private void raizRealDoble(int j){
//valor absoluto de la raíz
    double logaritmo=(Math.log(a[m][j+1])-Math.log(a[m][j-1]))/(2*pot2);
    double raiz=Math.exp(logaritmo);
    boolean bPositiva=false, bNegativa=false;
    if (Math.abs(valorPolinomio(raiz))<CERO){
      raicesReales[numReales]=raiz;
      numReales++;
      bPositiva=true;
    }
    if (Math.abs(valorPolinomio(-raiz))<CERO){
      raicesReales[numReales]=-raiz;
      numReales++;
      bNegativa=true;
    }
    if(bPositiva && !bNegativa){
     raicesReales[numReales]=raiz;
```

```
        numReales++;
      }
      if(!bPositiva && bNegativa){
        raicesReales[numReales]=-raiz;
        numReales++;
      }
    }
    private void unaRaizCompleja(){
      double suma=0.0;
      for(int i=0; i<numReales; i++){
        suma+=raicesReales[i];
      }
      double u, v;
      u=-(a[0][1]+suma)/2;
      v=Math.sqrt(moduloComplejas[0]*moduloComplejas[0]-u*u);
      raicesComplejas[0]=new Complejo(u, v);
      raicesComplejas[1]=new Complejo(u, -v);
    }
    private void dosRaicesComplejas(){
      double suma=0.0;
      double producto=1.0;
      double inversa=0.0;
      for(int i=0; i<numReales; i++){
        suma+=raicesReales[i];
        producto*=raicesReales[i];
        inversa+=1/raicesReales[i];
      }
      double r1=moduloComplejas[0];
      double r2=moduloComplejas[1];
      double y=-(a[0][1]+suma)/2;
      int signo=((n-1)%2==0)? +1: -1;
      double z=signo*a[0][n-1]/(2*producto)-r1*r1*r2*r2*inversa/2;
      double u1=(y*r1*r1-z)/(r1*r1-r2*r2);
      double u2=(-y*r2*r2+z)/(r1*r1-r2*r2);
      double v1=Math.sqrt(r1*r1-u1*u1);
      double v2=Math.sqrt(r2*r2-u2*u2);
      raicesComplejas[0]=new Complejo(u1, v1);
      raicesComplejas[1]=new Complejo(u1, -v1);
      raicesComplejas[2]=new Complejo(u2, v2);
      raicesComplejas[3]=new Complejo(u2, -v2);
    }
    private boolean cambiaSigno(int j){
      double logaritmo;
      for(int k=2; k<=m; k++){
        if(a[k][j]>0)   continue;
        numComplejas++;
//máximo dos raíces complejas, 4 contando sus respectivas conjugadas
        if(numComplejas<3){
          logaritmo=(Math.log(a[m][j+1])-Math.log(a[m][j-1]))/(2*pot2);
          moduloComplejas[numComplejas-1]=Math.exp(logaritmo);
          return true;
        }
      }
      return false;
    }

    public void hallarRaices(){
      tabla();
//el pimer coeficiente a[m][0] es siempre 1
      for(int i=1; i<n+1; i++){      //i es la raíz
        if(cambiaSigno(i)){
//raíz compleja y su correspondiente conjugada
          i++;
          continue;
        }
//raíces simple y dobles
        double logaritmo=Math.log(a[m][i])-2*Math.log(a[m-1][i]);
```

```
            if(Math.abs(logaritmo)<CERO){
                raizRealSimple(i);
            }else{
                raizRealDoble(i);
                i++;
                continue;
            }
        }
        if(numComplejas==1){
            unaRaizCompleja();
        }
        if(numComplejas==2){
            dosRaicesComplejas();
        }
     }

    public double [] mostrarRaices(int r){
        hallarRaices();
//        long
        long v;
        double raiz[] = new double[numReales+r];
//raíces reales
//        System.out.println("Raíces reales");
        for(int i=0; i<numReales; i++){
            raiz[i] = raicesReales[i];
        }
        return raiz;
//        System.out.println("");
//raíces complejas
//        System.out.println("Raíces complejas");
//        for(int i=0; i<numComplejas; i++){
//            System.out.println(raicesComplejas[2*i]+" ---> "+valorPolinomio(raicesComplejas[2*i]));
//            System.out.println(raicesComplejas[2*i+1]+" --->"+valorPolinomio(raicesComplejas[2*i]));
//        }
//        System.out.println("");
    }
}
```

Anexo 3
Código Fuente derivada enésima

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package matematicaderivacion;

import java.util.ArrayList;
import java.util.List;
import java.util.StringTokenizer;
/**
 *
 * @author Isaac Paliza <anndysaac@hotmail.com>
 */
public class DerivacionPolinomica {
    public static String[] cargarFx(String ImputFx) {
        String Fx = "";
        StringTokenizer auxFx = new StringTokenizer(ImputFx);
        while (auxFx.hasMoreElements()) {
            Fx += auxFx.nextElement();
        }
        int index = 0;
        int indexSeparador[] = new int[100];
        byte[] separador = Fx.trim().getBytes();
        for (int i = 0; i < separador.length; i++) {
            if (separador[i] == '+' || separador[i] == '-') {
                index++;
                indexSeparador[index] = i;
            }
        }
        String auxAuxFx[] = new String[index - 1];
        int b = 0;
        for (int i = 1, j = 2, t = 0; t <= index - 2; i = i + 1, j = j + 1, t++) {
            auxAuxFx[b] = Fx.trim().substring(indexSeparador[i], indexSeparador[j]).trim();
            b++;
        }
        return auxAuxFx;
    }
    public static String[] CompletarFx(String Fx[]) {
        String monomio = "";
        int grado = gradoFuncional(Fx) + 1;
        String auxFx[] = new String[grado];
        for (int i = 1; i <= Fx.length; i++) {
            monomio = Fx[i - 1];
            int indexSimbolo = 0;
            int indexVariable = 0;
            int indexTipoCoeficiente = 0;
            byte[] separacionp = monomio.trim().getBytes();
            for (int k = 0; k < separacionp.length; k++) {
                if (separacionp[k] == 'x') {
                    indexVariable = k;
                }
                if (separacionp[k] == '^') {
                    indexSimbolo = k;
                }
                if (separacionp[k] == '.') {
                    indexTipoCoeficiente = k;
                }
            }
            int exp = Integer.parseInt(monomio.substring(indexSimbolo + 1, monomio.length()));
            if (indexTipoCoeficiente == 0) {
                int cof = Integer.parseInt(monomio.substring(1, indexVariable));
                if (exp < 0) {
```

```
                //no hace nada
            }
            if (exp >= 0) {
//              auxFx[exp] = monomio.substring(0, indexVariable) + cof + "x^" + exp;
                auxFx[exp] = monomio.substring(0, indexVariable - Integer.toString(cof).length()) + cof +
"x^" + exp;
            }
        }
        if (indexTipoCoeficiente != 0) {
//              int exp = Integer.parseInt(monomio.substring(indexSimbolo + 1, monomio.length()));
            double cof = Double.parseDouble(monomio.substring(1, indexVariable));

            if (exp < 0) {
                //no hace nada
            }
            if (exp >= 0) {
//              auxFx[exp] = monomio.substring(0, indexVariable) + cof + "x^" + exp;
                auxFx[exp] = monomio.substring(0, indexVariable - Double.toString(cof).length()) + cof +
"x^" + exp;
            }
        }

    }
    for (int i = 0; i < auxFx.length; i++) {
        if (auxFx[i] == null) {
            auxFx[i] = "+0x^" + i;
        }
    }
    return OrdenDecrecienteFx(auxFx);
}
public static String[] OrdenDecrecienteFx(String Fx[]) {
    int gradoFx = gradoFuncional(Fx);
    String auxFx[] = new String[gradoFx + 1];
    for (int i = gradoFx, j = 0; i >= 0; j++, i--) {
        auxFx[j] = Fx[i];
    }
    return auxFx;
}

public static int gradoFuncional(String Fx[]) {
    String monomio = "";
    int gradoFx[] = new int[Fx.length];
    for (int i = 0; i < Fx.length; i++) {
        monomio = Fx[i];
        if (monomio == null) {
            //no hace nada
        } else {
            int indexSimbolo = 0;
            byte[] separacionp = monomio.trim().getBytes();
            for (int k = 0; k < separacionp.length; k++) {
                if (separacionp[k] == '^') {
                    indexSimbolo = k;
                }
            }
            int exp = Integer.parseInt(monomio.substring(indexSimbolo + 1, monomio.length()));
            gradoFx[i] = exp;
        }
    }
    int gradoMayor = gradoFx[0];
    if (Fx.length == 1 && gradoMayor == 0) {
        return gradoFx[0];
    }
    for (int k = 0; k < Fx.length; k++) {
        if (gradoFx[k] > gradoMayor) {
            gradoMayor = gradoFx[k];
        }
    }
```

```
        return gradoMayor;
    }

    public static String[] derivarFx(String Fx[]) {
        int grado = gradoFuncional(Fx);
        if (grado == 0) {
            grado = 1;
        }
        String auxFx[] = new String[grado];
        for (int i = 1; i <= Fx.length; i++) {
            String monomio = Fx[i - 1];
            int ev = 0;
            int pv = 0;
            int v = 0;
            byte[] separacionp = monomio.trim().getBytes();
            for (int k = 0; k < separacionp.length; k++) {
                if (separacionp[k] == '^') {
                    ev = k;
                }
                if (separacionp[k] == 'x') {
                    pv = k;
                }
                if (separacionp[k] == '.') {
                    v = k;
                }
            }
            int exp = Integer.parseInt(monomio.substring(ev + 1, monomio.length()));
            if (v == 0) {
                int cof = Integer.parseInt(monomio.substring(1, pv));

                int cofNuevo = exp * cof;
                int expNuevo = exp - 1;
                if (expNuevo < 0) {
                    //no hace nada
                }
                if (expNuevo >= 0) {
                    auxFx[expNuevo] = monomio.substring(0, pv - Integer.toString(cof).length()) + cofNuevo +
"x^" + expNuevo;
                }
            }
            if (v != 0) {
                Double cof = Double.parseDouble(monomio.substring(1, pv));
                Double cofNuevo = exp * cof;
                int expNuevo = exp - 1;
                if (expNuevo < 0) {
                    //no hace nada
                }
                if (expNuevo >= 0) {
                    auxFx[expNuevo] = monomio.substring(0, pv - Double.toString(cof).length()) + cofNuevo +
"x^" + expNuevo;
                }
            }
        }
        for (int i = 0, t = 0; t < auxFx.length; i++, t++) {
            if (auxFx[i] == null) {
                auxFx[i] = "+0x^" + i;
            }
        }
        return OrdenDecrecienteFx(auxFx);
    }

    public static double ReducirDouble(String result) {
        byte[] separador = result.trim().getBytes();
        String n = "";
        for (int i = 0; i < separador.length; i++) {
            if (separador[i] == '.') {
                n = result.substring(0, i + 2);
```

```
            }
        }
        return Double.parseDouble(n);
    }

    public static double ReemplazoCoordFxGrafico(double Fx[], double raiz, int ii) {
        double result = 0;
        for (int i = 0, j = Fx.length - 1; i < Fx.length; i++, j--) {
            result += Fx[i] * (Math.pow(raiz, j));
        }
        return result;
    }

    public static double ReemplazoCoordFx_Grafico(double Fx[], double raiz, int ii) {
        double result = 0;
        for (int i = 0, j = Fx.length - 1; i < Fx.length; i++, j--) {
            result += Fx[i] * (Math.pow(raiz, j));
        }
        return result;
    }
    public static double ReemplazoCoordFx(double Fx[], double raiz, int ii) {
        double result = 0;
        for (int i = 0, j = Fx.length - 1; i < Fx.length; i++, j--) {
            if(raiz == 0 && i==(Fx.length-1)){
                result = Fx[i] * (Math.pow(raiz, j));
            }
            if(raiz != 0){
                result += Fx[i] * (Math.pow(raiz, j));
            }
        }
        if(ii == 1){
            return (double)Math.round(result);
        }
        return (double)Math.round(result*1000)/1000;
    }
    public static double ReemplazoCoordFx____(double Fx[], double raiz, int ii) {
        double result = 0;
        for (int i = 0, j = Fx.length - 1; i < Fx.length; i++, j--) {
            if (raiz != 0) {
                result += Fx[i] * (Math.pow(raiz, j));
            }
            if (raiz == 0 && i == (Fx.length - 1)) {
                result = Fx[i] * (Math.pow(raiz, j));
            }

        }
        if (ii == 1) {
            return (double) Math.round(result);
        }
        return (double) Math.round(result * 1000) / 1000;
    }
    public static double[] ReducionCoeficientes(String Fx[]) {
        String monomio = "";

//        int grado = gradoFuncional(Fx) + 1;
        int grado = Fx.length;
        double coefFx[] = new double[grado];
        for (int i = 0; i < grado; i++) {
            monomio = Fx[i];
            if (monomio == null) {
                coefFx[i] = 0;
            } else {

                int index = 0;
                int indexR = 0;
                byte[] separacion = monomio.trim().getBytes();
                for (int j = 0; j < separacion.length; j++) {
```

```
                if (separacion[j] == 'x') {
                    index = j;
                }
                if (separacion[j] == '.') {
                    indexR = j;
                }
            }
            if (indexR == 0) {
                coefFx[i] = Integer.parseInt(monomio.substring(0, index));
            }
            if (indexR != 0) {
                coefFx[i] = Double.parseDouble(monomio.substring(0, index));
            }
        }
    }
    if (coefFx.length == 1) {
        return coefFx;
    }
    int iiC = uC(coefFx);
    if (iiC > 0) {
        if (coefFx[grado - iiC] == 0) {
            double newCoef[] = new double[grado - iiC];
            for (int i = 0, j = 0; i < grado - iiC; j++, i++) {
                newCoef[i] = coefFx[j];
            }
            return newCoef;
        }
    }
    return coefFx;
}

public static double[] Coeficientes(String Fx[]) {
    String monomio = "";

//        int grado = gradoFuncional(Fx) + 1;
    int grado = Fx.length;
    double coefFx[] = new double[grado];
    for (int i = 0; i < grado; i++) {
        monomio = Fx[i];
        if (monomio == null) {
            coefFx[i] = 0;
        } else {

            int index = 0;
//                int indexR = 0;
            byte[] separacion = monomio.trim().getBytes();
            for (int j = 0; j < separacion.length; j++) {
                if (separacion[j] == 'x') {
                    index = j;
                }
//                    if (separacion[j] == '.') {
//                        indexR = j;
//                    }
            }
//                if (indexR == 0) {
//                    coefFx[i] = Integer.parseInt(monomio.substring(0, index));
//                }
//                if (indexR != 0) {
            coefFx[i] = Double.parseDouble(monomio.substring(0, index));
//                }
        }
    }
//        if (coefFx.length == 1) {
//            return coefFx;
//        }
//        int iiC = uC(coefFx);
//        if (iiC > 0) {
```

```
//            if (coefFx[grado - iiC] == 0) {
//                double newCoef[] = new double[grado - iiC];
//                for (int i = 0, j = 0; i < grado - iiC; j++, i++) {
//                    newCoef[i] = coefFx[j];
//                }
//                return newCoef;
//            }
//        }
        return coefFx;
    }

    public static int uC(double coef[]) {
        int pC = 0;
        int tamaño = coef.length;
        for (int i = 1, t = 0; t < tamaño; i++, t++) {
            double iC = coef[tamaño - i];
            if (iC == 0) {
                pC++;
            } else if (iC != 0) {
                return pC;
            }
        }
        return pC;
    }

    public static List DerivadaEnesima(String F2x[], double raizF1x[]) {
        List a = new ArrayList();
        for (int i = 0; i < raizF1x.length; i++) {
//            String vecFx[] = new String[F2x.length + 1];
            List vecFx = new ArrayList();
            a.add(Derivada(vecFx, F2x, raizF1x[i], 0));
        }
        return a;
    }
    public static List DerivadaEnesima2(String F2x[], double raizF1x[]) {
        List a = new ArrayList();
        for (int i = 0; i < raizF1x.length; i++) {
//            String vecFx[] = new String[F2x.length + 1];
            List vecFx = new ArrayList();
            a.add(Derivada2(vecFx, F2x, raizF1x[i], 0));
        }
        return a;
    }
    static List p1 = new ArrayList();
    static List p2 = new ArrayList();
    public static List Derivada(List vecFx, String F2x[], double raizF1x, int i) {
        if (i == 0) {
            double reempl = ReemplazoCoordFx(Coeficientes(F2x), raizF1x, 1);
            String auxnFx = "";
            for (String nFx1 : F2x) {
                auxnFx += nFx1;
            }
            String auxAuxnFx = "";
            String simbolo = auxnFx.substring(0, 1);
            if (simbolo.equals("+")) {
                auxAuxnFx = auxnFx.substring(1, auxnFx.length());
            }
            if (simbolo.equals("-")) {
                //Cambio de signos al Fx
            }
            vecFx.add("Iteracion         "+(i+2)+"         Fd"+(i+2)+"("+Math.rint(raizF1x*10000)/10000+")         :
"+auxAuxnFx+" = "+Math.rint(reempl*10000)/10000);
            if (reempl != 0) {
                p1.add(reempl);
                return vecFx;
            }
//            if (reempl == 0) {
```

```
//              return vecFx;
//          }
            i++;
            return Derivada(vecFx, F2x, raizF1x, i);
        } else if (i != 0) {
            String nFx[] = derivarFx(F2x);
            double reempl = ReemplazoCoordFx(Coeficientes(nFx), raizF1x, 1);
            if (reempl != 0) {
                String auxnFxx = "";
                for (String nFx1 : nFx) {
                    auxnFxx += nFx1;
                }
                String auxAuxnFx = "";
                String simbolo = auxnFxx.substring(0, 1);
                if (simbolo.equals("+")) {
                    auxAuxnFx = auxnFxx.substring(1, auxnFxx.length());
                }
                if (simbolo.equals("-")) {
                    //Cambio de signos al Fx
                }
                p1.add(reempl);
                vecFx.add("Iteracion       "+(i+2)+"       Fd"+(i+2)+"("+Math.rint(raizF1x*10000)/10000+")       :
"+auxAuxnFx+" = "+Math.rint(reempl*10000)/10000);
                return vecFx;
            }
            String auxnFxx = "";
            for (String nFx1 : nFx) {
                auxnFxx += nFx1;
            }
            String auxAuxnFx = "";
            String simbolo = auxnFxx.substring(0, 1);
            if (simbolo.equals("+")) {
                auxAuxnFx = auxnFxx.substring(1, auxnFxx.length());
            }
            if (simbolo.equals("-")) {
                //Cambio de signos al Fx
            }
            vecFx.add("Iteracion      "+(i+2)+      "      Fd"+(i+2)+"("+Math.rint(raizF1x*10000)/10000+")       :
"+auxAuxnFx+" = "+Math.rint(reempl*10000)/10000);
            i++;
            return Derivada(vecFx, nFx, raizF1x, i);
        }
        return vecFx;
    }
       public static List Derivada2(List vecFx, String F2x[], double raizF1x, int i) {
        if (i == 0) {
            double reempl = ReemplazoCoordFx(Coeficientes(F2x), raizF1x, 1);

            String auxnFx = "";
            for (String nFx1 : F2x) {
                auxnFx += nFx1;
            }
            String auxAuxnFx = "";
            String simbolo = auxnFx.substring(0, 1);
            if (simbolo.equals("+")) {
                auxAuxnFx = auxnFx.substring(1, auxnFx.length());
            }
            if (simbolo.equals("-")) {
                //Cambio de signos al Fx
            }
            vecFx.add(" Iteracion " + (i + 2) + "  Fd" + (i + 2) + "(" + Math.rint(raizF1x * 100) / 100 + "): " +
auxAuxnFx + " = " + reempl);
            if (reempl != 0) {
                p2.add(reempl);
                return vecFx;
            }
//          if (reempl == 0) {
```

```
//              return vecFx;
//           }
          i++;
          return Derivada(vecFx, F2x, raizF1x, i);
      } else if (i != 0) {
          String nFx[] = derivarFx(F2x);
          double reempl = ReemplazoCoordFx(Coeficientes(nFx), raizF1x, 1);
          if (reempl != 0) {
              String auxnFxx = "";
              for (String nFx1 : nFx) {
                  auxnFxx += nFx1;
              }
              String auxAuxnFx = "";
              String simbolo = auxnFxx.substring(0, 1);
              if (simbolo.equals("+")) {
                  auxAuxnFx = auxnFxx.substring(1, auxnFxx.length());
              }
              if (simbolo.equals("-")) {
                  //Cambio de signos al Fx
              }
              p2.add(reempl);
              vecFx.add(" Iteracion " + (i + 2) + "  Fd" + (i + 2) + "(" + Math.rint(raizF1x * 100) / 100 + "): " +
auxAuxnFx + " = " + reempl);
              return vecFx;
          }
          String auxnFxx = "";
          for (String nFx1 : nFx) {
              auxnFxx += nFx1;
          }
          String auxAuxnFx = "";
          String simbolo = auxnFxx.substring(0, 1);
          if (simbolo.equals("+")) {
              auxAuxnFx = auxnFxx.substring(1, auxnFxx.length());
          }
          if (simbolo.equals("-")) {
              //Cambio de signos al Fx
          }
          vecFx.add("Iteracion " + (i + 2) + "  Fd" + (i + 2) + "(" + Math.rint(raizF1x * 100) / 100 + "): " +
auxAuxnFx + " = " + reempl);
          i++;
          return Derivada(vecFx, nFx, raizF1x, i);
      }
      return vecFx;
  }
  public static double[] evaluar(double coef[], double FPrimos, int tCoef) {
      double resultado[] = new double[coef.length];
      for (int i = 0; i < coef.length; i++) {
          if (i == 0) {
              resultado[i] = coef[i];
          }
          if (i != 0) {
              double r = resultado[i - 1] * FPrimos + coef[i];
              resultado[i] = r;
          }
      }
      return resultado;
  }
  public static double[] raizPolinomica(double raiz[], double FPrimos[], double coef[], int tCoef, int j) {
//       if (tCoef == 14) {
//           return raiz;
//       }
      for (int i = 0; i < FPrimos.length; i++) {
          double co[] = evaluar(coef, FPrimos[i], tCoef);
          if (co[co.length - 1] == 0) {
              raiz[j] = FPrimos[i];
              raizPolinomica(raiz, FPrimos, co, tCoef + 1, j + 1);
          }
```

```
    }
    return raiz;
}
static String[] auxFx;

public static String EncadenarRaiz(double raiz[]) {
    String raices = "";
    for (int i = 0; i < raiz.length; i++) {
        raices += " (" + Math.rint(raiz[i]*10000)/10000 + "; 0)\n";
    }
    return raices;
}
public static String EncadenarRaizSIMPLE(double raiz) {
    return " (" + Math.rint(raiz*10000)/10000 + "; 0)\n";
}// Math.rint(raizFd1[y]*10000)/10000
static double raizFd1[];
static double raizFd2[];

public static List CargarDatos(String imputTextFx) {
    List a = new ArrayList();
    String dato[] = new String[7];
    auxFx = CompletarFx(cargarFx(imputTextFx));
    String Fx = "";
    for (int i = 0; i < auxFx.length; i++) {
        Fx += auxFx[i] + " ";
    }
    String datoCorteF1;
    dato[0] = "F(x) = " + Fx + "\n";
    datoCorteF1 = "PUNTO DE CORTE CON EL EJE Y\n";
    datoCorteF1 += "(0; " + ReemplazoCoordFx(Coeficientes(auxFx), 0, 1) + ")\n";
    dato[1] = datoCorteF1;

    String datoCorteF2 = "";
    double[] aux2Fx = ReducionCoeficientes(auxFx);
    int flag = 1;
    double raiz = 0;
    double R_2G[] = new double[2];
    if (aux2Fx.length == 3 || aux2Fx.length == 4 || aux2Fx.length == 5) {
        flag = 0;
    }
    if (aux2Fx.length == 5) {
        if (Math.abs(aux2Fx[0]) == 0) { // x: 4
            flag = 1;
        }
        if (Math.abs(aux2Fx[1]) != 0) { // x: 3
            flag = 1;
        }
        if (Math.abs(aux2Fx[2]) != 0) { // x: 2
            flag = 1;
        }
        if (Math.abs(aux2Fx[3]) != 0) { // x: 1
            flag = 1;
        }
        if (Math.abs(aux2Fx[4]) == 0) { // x: 0
            flag = 1;
        }
        if (flag == 0) {
            if(((aux2Fx[4]/aux2Fx[0])*-1) < 0){
                flag = 1;
            }
            if(((aux2Fx[4]/aux2Fx[0])*-1) >= 0){
                raiz = Math.pow((aux2Fx[4]/aux2Fx[0])*-1, 1.0/4.0);
            }
        }
    }
    if (aux2Fx.length == 4) {
        if (Math.abs(aux2Fx[0]) == 0) {
```

```
                flag = 1;
            }
            if (Math.abs(aux2Fx[1]) != 0) {
                flag = 1;
            }
            if (Math.abs(aux2Fx[2]) != 0) {
                flag = 1;
            }
            if (Math.abs(aux2Fx[3]) == 0) {
                flag = 1;
            }
            if (flag == 0) {
                if(((aux2Fx[3]/aux2Fx[0])*-1) < 0){
                    raiz = Math.pow((aux2Fx[3]/aux2Fx[0]), 1.0/3.0)*-1;
                }
                if(((aux2Fx[3]/aux2Fx[0])*-1) >= 0){
                    raiz = Math.pow((aux2Fx[3]/aux2Fx[0])*-1, 1.0/3.0);
                }
            }
        }
        if (aux2Fx.length == 3) {
            if (flag == 0) {
                double verific_1 = (aux2Fx[1]*aux2Fx[1])-(4*aux2Fx[0]*aux2Fx[2]);
                if (verific_1 < 0) {
                    flag = 1;
                }
                if (verific_1 >= 0) {
                    double Res1 = Math.pow(verific_1, 1.0/2.0);
                    R_2G[0] = ((aux2Fx[1]*-1) + Res1)/(2*aux2Fx[0]);
                    R_2G[1] = ((aux2Fx[1]*-1) - Res1)/(2*aux2Fx[0]);
                }
//                datoCorteF2  += verific_1+"n\n";
            }
        }
//        datoCorteF2  += auxFx.length+"P\n";
//        datoCorteF2  += aux2Fx.length+"P\n";
//        datoCorteF2  += flag+"P\n";
        if(flag == 0){
            int R0_2 = 0;
            int R0 = auxFx.length - aux2Fx.length;
            if (R0 > 0) {
                R0_2 = 1;
            }
            datoCorteF2  += "PUNTO DE CORTE CON EL EJE X\n";
            if (aux2Fx.length != 3) {
                datoCorteF2  += "nro. de raices reales  "+(2+R0_2)+"\n";
                datoCorteF2  += EncadenarRaizSIMPLE(raiz);
            }
            if (aux2Fx.length == 3) {
                datoCorteF2  += EncadenarRaizSIMPLE(R_2G[0]);
                datoCorteF2  += EncadenarRaizSIMPLE(R_2G[1]);
            }
            if (aux2Fx.length == 5) {
                datoCorteF2  += EncadenarRaizSIMPLE(raiz*-1);
            }
            if (R0 > 0) {
                datoCorteF2  += EncadenarRaizSIMPLE(0);
            }
            dato[2] = datoCorteF2;
        }
        if(flag == 1){
            Graeffe FG0 = new Graeffe(aux2Fx);
            int R0 = auxFx.length - aux2Fx.length;
            if (R0 > 0) {
                R0 = 1;
            }
            datoCorteF2  += "PUNTO DE CORTE CON EL EJE X\n";
```

```
    double raizFx[] = FG0.mostrarRaices(R0);
    datoCorteF2 += "nro. de raices reales " + (Graeffe.numReales + R0) + "\n";
    datoCorteF2 += EncadenarRaiz(raizFx) + "\n";
    if(Graeffe.numComplejas > 0){
        datoCorteF2 += "nro. de raices no reales " + Graeffe.numComplejas;
    }
    dato[2] = datoCorteF2;
}
double R_PCFd1[] = new double[1];
double R_PCFd3[] = new double[2];
String datoPrimeraDerivada;
datoPrimeraDerivada = "\nPRIMERA DERIVADA DE F'(x)\n";
String[] Fd1 = derivarFx(auxFx);
for (int i = 0; i < Fd1.length; i++) {
    datoPrimeraDerivada += Fd1[i] + " ";
}
dato[3] = datoPrimeraDerivada;
String datoPuntosCriticosFd1 = "\nPuntos Criticos\n";

double[] PCFd1 = ReducionCoeficientes(Fd1);
int flag1 = 1;
if (PCFd1.length == 3 || PCFd1.length == 4 || PCFd1.length == 5) {
    flag1 = 0;
}
if (PCFd1.length == 5) {
    if (Math.abs(PCFd1[0]) == 0) { // x: 4
        flag1 = 1;
    }
    if (Math.abs(PCFd1[1]) != 0) { // x: 3
        flag1 = 1;
    }
    if (Math.abs(PCFd1[2]) != 0) { // x: 2
        flag1 = 1;
    }
    if (Math.abs(PCFd1[3]) != 0) { // x: 1
        flag1 = 1;
    }
    if (Math.abs(PCFd1[4]) == 0) { // x: 0
        flag1 = 1;
    }
    if (flag1 == 0) {
        if(((PCFd1[4]/PCFd1[0])*-1) < 0){
            flag1 = 1;
        }
        if(((PCFd1[4]/PCFd1[0])*-1) >= 0){
            R_PCFd3[0] = Math.pow((PCFd1[4]/PCFd1[0])*-1, 1.0/4.0);
            R_PCFd3[1] = Math.pow((PCFd1[4]/PCFd1[0])*-1, 1.0/4.0)*-1;
        }
    }
}
if (PCFd1.length == 4) {
    if (Math.abs(PCFd1[0]) == 0) {
        flag1 = 1;
    }
    if (Math.abs(PCFd1[1]) != 0) {
        flag1 = 1;
    }
    if (Math.abs(PCFd1[2]) != 0) {
        flag1 = 1;
    }
    if (Math.abs(PCFd1[3]) == 0) {
        flag1 = 1;
    }
    if (flag1 == 0) {
        if(((PCFd1[3]/PCFd1[0])*-1) < 0){
            R_PCFd1[0] = Math.pow((PCFd1[3]/PCFd1[0]), 1.0/3.0)*-1;
        }
```

```
        if(((PCFd1[3]/PCFd1[0])*-1) >= 0){
            R_PCFd1[0] = Math.pow((PCFd1[3]/PCFd1[0])*-1, 1.0/3.0);
        }
    }
}
if (PCFd1.length == 3) {
    if (flag1 == 0) {
        double verific_1 = (PCFd1[1]*PCFd1[1])-(4*PCFd1[0]*PCFd1[2]);
        if (verific_1 < 0) {
            flag1 = 1;
        }
        if (verific_1 >= 0) {
            double Res1 = Math.pow(verific_1, 1.0/2.0);
            R_PCFd3[0] = ((PCFd1[1]*-1) + Res1)/(2*PCFd1[0]);
            R_PCFd3[1] = ((PCFd1[1]*-1) - Res1)/(2*PCFd1[0]);
        }
    }
}
if (flag1 == 0) {
    int R1 = Fd1.length - PCFd1.length;
    raizFd1 = R_PCFd1;
    if(PCFd1.length == 3 || PCFd1.length == 5) {
        raizFd1 = R_PCFd3;
    }
    if (R1 > 0) {
        double R_PCFd_2[] = new double[raizFd1.length+1];
        for(int i = 0; i < raizFd1.length; i++) {
            R_PCFd_2[i] = raizFd1[i];
        }
        R_PCFd_2[raizFd1.length] = 0;
        raizFd1 = R_PCFd_2;
    }
    for(int i = 0; i < raizFd1.length; i++) {
        datoPuntosCriticosFd1 += "x = " + Math.rint(raizFd1[i]*10000)/10000 + "\n";
    }
    dato[4] = datoPuntosCriticosFd1;
}
if (flag1 == 1) {
    Graeffe FG1 = new Graeffe(PCFd1);
    int R1 = Fd1.length - PCFd1.length;
    if (R1 > 0) {
        R1 = 1;
    }
    raizFd1 = FG1.mostrarRaices(R1);
    for (int i = 0; i < raizFd1.length; i++) {
        datoPuntosCriticosFd1 += " x = " + Math.rint(raizFd1[i]*10000)/10000 + "\n";
    }
    dato[4] = datoPuntosCriticosFd1;
}
double R_PIFd2[] = new double[1];
double R_PIFd4[] = new double[2];
String datoSegundaDerivada = "SEGUNDA DERIVADA DE F''(x)\n";
String[] Fd2 = derivarFx(Fd1);
for (int i = 0; i < Fd2.length; i++) {
    datoSegundaDerivada += Fd2[i] + " ";
}
dato[5] = datoSegundaDerivada;
String datosPuntodeInflexion = "\nPuntos de Inflexión\n";
double[] PIFd1 = ReducionCoeficientes(Fd2);
int flag2 = 1;
if (PIFd1.length == 4 || PIFd1.length == 5) {
    flag2 = 0;
}
if(PIFd1.length == 4){
    if(Math.abs(PIFd1[0]) ==0){
        flag2 = 1;
    }
```

```
    if(Math.abs(PIFd1[1]) !=0){
        flag2 = 1;
    }
    if(Math.abs(PIFd1[2]) !=0){
        flag2 = 1;
    }
    if(Math.abs(PIFd1[3]) ==0){
        flag2 = 1;
    }
    if(flag2 ==0){
        if(((PIFd1[3]/PIFd1[0])*-1) < 0){
            R_PIFd2[0] = Math.pow((PIFd1[3]/PIFd1[0]), 1.0/3.0)*-1;
        }
        if(((PIFd1[3]/PIFd1[0])*-1) >= 0){
            R_PIFd2[0] = Math.pow((PIFd1[3]/PIFd1[0])*-1, 1.0/3.0);
        }
    }
}
if(PIFd1.length == 5){
    if(Math.abs(PIFd1[0]) ==0){
        flag2 = 1;
    }
    if(Math.abs(PIFd1[1]) !=0){
        flag2 = 1;
    }
    if(Math.abs(PIFd1[2]) !=0){
        flag2 = 1;
    }
    if(Math.abs(PIFd1[3]) !=0){
        flag2 = 1;
    }
    if(Math.abs(PIFd1[4]) ==0){
        flag2 = 1;
    }
    if(flag2 ==0){
        if(((PIFd1[4]/PIFd1[0])*-1) < 0){
            flag2 = 1;
        }
        if(((PIFd1[4]/PIFd1[0])*-1) >= 0){
            R_PIFd4[0] = Math.pow((PIFd1[4]/PIFd1[0])*-1, 1.0/4.0);
            R_PIFd4[1] = Math.pow((PIFd1[4]/PIFd1[0])*-1, 1.0/4.0)*-1;
        }
    }
}
if (PIFd1.length == 3) {
    if (flag2 == 0) {
        double verific_1 = (PIFd1[1]*PIFd1[1])-(4*PIFd1[0]*PIFd1[2]);
        if (verific_1 < 0) {
            flag2 = 1;
        }
        if (verific_1 >= 0) {
            double Res1 = Math.pow(verific_1, 1.0/2.0);
            R_PCFd3[0] = ((PIFd1[1]*-1) + Res1)/(2*PIFd1[0]);
            R_PCFd3[1] = ((PIFd1[1]*-1) - Res1)/(2*PIFd1[0]);
        }
    }
}
if(flag2 == 0){
    int R2 = Fd2.length - PIFd1.length;
    raizFd2 = R_PIFd2;
    if(PIFd1.length == 3 || PIFd1.length == 5 ) {
        raizFd2 = R_PIFd4;
    }
    if (R2 > 0) {
        double R_PCFd_5[] = new double[raizFd2.length+1];
        for(int i = 0; i < raizFd2.length; i++) {
            R_PCFd_5[i] = raizFd2[i];
```

```
                }
                R_PCFd_5[raizFd2.length] = 0;
                raizFd2 = R_PCFd_5;
            }
            for (int i = 0; i < raizFd2.length; i++) {
                datosPuntodeInflexion += "x = "+Math.rint(raizFd2[i]*10000)/10000 +"\n";
            }
            dato[6] = datosPuntodeInflexion;
        }
        if(flag2 == 1){
            Graeffe FG2 = new Graeffe(PIFd1);
            int R2 = Fd2.length - PIFd1.length;
            if (R2 > 0) {
                R2 = 1;
            }
            raizFd2 = FG2.mostrarRaices(R2);
            for (int i = 0; i < raizFd2.length; i++) {
                datosPuntodeInflexion += "x = " + Math.rint(raizFd2[i]*10000)/10000 + "\n";
            }
            dato[6] = datosPuntodeInflexion;
        }
        a.add(dato);
        a.add(DerivacionPolinomica.DerivadaEnesima(Fd2, raizFd1));
        a.add(DerivacionPolinomica.DerivadaEnesima(Fd2, raizFd2));
        return a;
    }
}
```