



**ING. JULIO CESAR LOPEZ**

**INGENIERIA EN SISTEMAS**

**PROYECTO CALCULO I**

**Universidad Mariano Gálvez de Guatemala**

**CENTRO UNIVERSITARIO PETEN**

**Facultad de ingeniería en sistemas de**

**Información y ciencias de la computación**

**Curso: Calculo I**

**I ciclo sección “A”**

**Titular: Ing. Julio cesar López**

Proyecto Derivadas Exponenciales y Logarítmicas

Nombre: Carné:

Kevin Oliver Mateo Flores 1690-17-17904

Anthony Enrique Chen Dubon 1690-19-9261

Diego Javier Trujillo García 1690-10-6032

Elizabeth Roxana González Castro 1690-20-3141

Yeimi Natalia Contreras Godoy 1690-20-6811

Marlon Josué Sulecio Ramos 1690-20-20962

Kiener Francisco Maldonado Zamora 1690-17-18958

Brayan Rivaldo Obando Casasola 1690-19-22993

José Antonio Solis Góngora 1690-19-10364

Mario Gerardo Muralles Mendoza 1609-18-9195

Pedro Eduardo Albani Díaz 1690-17-7033

Santa Elena, Flores, Petén

**ÍNDICE**

[INTRODUCCIÓN 4](#_Toc73464770)

[OBJETIVO 5](#_Toc73464771)

[FUNCIONES EXPONENCIALES Y LOGARÍTMICA 6](#_Toc73464772)

[¿Qué es la función exponencial y logarítmica? 6](#_Toc73464773)

[¿Cómo se comporta la función exponencial? 6](#_Toc73464774)

[Funciones Exponenciales 6](#_Toc73464775)

[Funciones Logarítmica 8](#_Toc73464776)

[Funcionamiento de librería 10](#_Toc73464777)

[Características 10](#_Toc73464778)

[Análisis y evaluación 10](#_Toc73464779)

[Configuración por defecto 11](#_Toc73464780)

[Métodos de evaluación 11](#_Toc73464781)

[Variables 12](#_Toc73464782)

[Asignación 12](#_Toc73464783)

[Funciones logarítmicas y exponenciales 13](#_Toc73464784)

[JAVA 14](#_Toc73464785)

[CALCULADORA DE DERIVADAS EXPONENCIALES Y ALGORÍTMICAS 14](#_Toc73464786)

[CONCLUSIÓN 16](#_Toc73464787)

[ANEXOS 17](#_Toc73464788)

[27](#_Toc73464789)

[27](#_Toc73464790)

[E-grafía 28](#_Toc73464791)

# **INTRODUCCIÓN**

Como proyecto se desarrolló una calculadora en Java, con la cual se pueden calcular las derivadas exponenciales y logarítmicas, con la finalidad de poner en práctica todo lo aprendido en el curso. También para poner en práctica lo aprendido en programación.

La derivada es el resultado de un límite y representa la pendiente de la recta tangente a la gráfica de la función en un punto.

La función logarítmica es una función inversa a la función exponencial. La función exponencial se encuentra definida para todos los reales.

# **OBJETIVO**

Comprender y aplicar los conceptos de las derivadas exponenciales y logarítmicas, poniendo en práctica lo aprendido en el curso de Cálculo I.

* Identificar la diferencia de las funciones exponenciales y logarítmicas.
* Tener la capacidad de desarrollar un programa que resuelva las derivadas.
* Ampliar los conocimientos en programación orientada a problemas matemáticos.

# **FUNCIONES EXPONENCIALES Y LOGARÍTMICA**

Las funciones exponenciales y = ax funciones logarítmicas logay = x se le denominan funciones transcendentales, ya que son funciones que transcienden el álgebra en el sentido que ninguna puede ser expresada en términos de una secuencia finita de operaciones algebraicas de suma, resta y/o extracción de raíces.

## ¿Qué es la función exponencial y logarítmica?

Las funciones exponenciales y logarítmicas con base son inversas una de otra. Por lo tanto, cuando en una expresión y = ax nos dan “a” y “x” para calcular “y”, estamos en presencia de una función exponencial, pero cuando nos dan “a” e “y” para calcular x, estamos en presencia de una función logarítmica.

## ¿Cómo se comporta la función exponencial?

Toda función exponencial es de la forma f(x)=ax, donde a es la base que siempre será un número mayor de cero y diferente de 1. El exponente x es cualquier número real. Como vemos su variable está en el exponente mientras la base es una constante.

# **Funciones Exponenciales**

Toda función f: R → R+\* tal que f(x) = ax con a ≠ 1 y a > 0, se le denomina función exponencial.

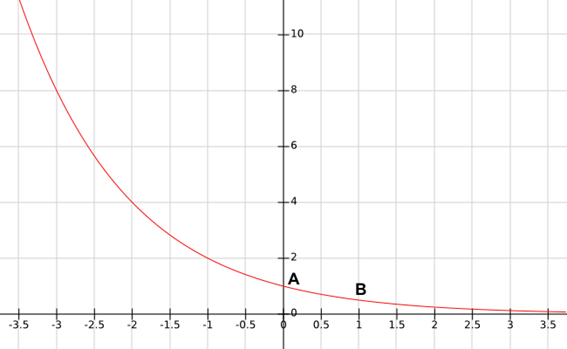
* Como a0= 1, la curva pasa por el punto (0,1).
* Como a1= a, la curva pasa por el punto (1,a).

El valor de y en la función f(x) = ax para cualquier número del conjunto R siempre es un número positivo y nunca puede valer cero, ya que no hay ningún número x que sustituido en la expresión de la función de cómo resultado cero. Por ello la curva siempre está “por encima” del eje x (no lo corta).

* Cuando a > 1 la curva es estrictamente creciente.
* Cuando a < 1 la curva es estrictamente decreciente.

**Ejemplo**: Sea f: R → R+\* tal que f(x) = (1/2)x. Realizar la representación gráfica de la misma.

Haciendo la representación gráfica para el intervalo, – 3 ≤ x ≤ 3 se tiene:



Veamos que:

* La curva pasa por el punto A(0,1).
* La curva pasa por el punto B(1,1/2)
* La Curva está “por encima” del eje x y no lo corta.
* La función es estrictamente decreciente ya que a < 1, con a = 1/2.

# **Funciones Logarítmica**

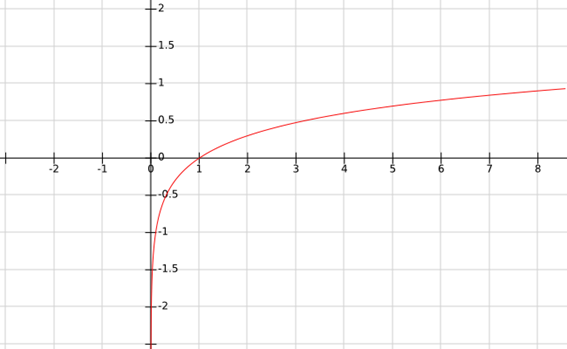
Toda función f: R → R+\* tal que logaf(x) = ax con a ≠ 1 y a > 0, se le denomina **función logarítmica**. Esta función es la inversa de la función de la exponencial en base a, dado que:

logaf(y) = x    ↔     ax= y

* La función logarítmica sólo existe para x > 0 (sin incluir el cero). Por tanto, su dominio es el intervalo (0,+∞).
* Cuando x = 1, la función logarítmica se anula, ya que logaf(1) = 0, en cualquier base.
* La función logarítmica de la base es siempre igual a 1.
* La curva es continua, y es creciente para a > 1 y decreciente para a < 1.

**Ejemplo**: Sea f: R → R+\* tal que y = log(x) , realizar la representación gráfica de la misma.

Haciendo la representación gráfica para el intervalo -1/2 ≤ x ≤ 8, se tiene:



Veamos que:

* La Curva está “a la derecha” del eje “y” y no lo corta.
* La función es creciente ya que a > 1, con a = 10.

# **Funcionamiento de librería**

Jep es una biblioteca de Java para analizar y evaluar expresiones matemáticas. Con este paquete, puede tomar fórmulas como cadenas y evaluarlas instantáneamente. Muchas funciones y constantes matemáticas comunes están integradas y listas para usar. Pero, además, puede ampliar Jep con sus propias variables, constantes y funciones definidas por el usuario.

## Características

* Tamaño pequeño (menos de 400 KB como archivo jar)
* Evaluación rápida
* Alta precisión con BigDecimals
* Incluye funciones y operadores matemáticos comunes
* Admite expresiones booleanas
* Arquitectura extensible y configurable
* Soporte para cadenas, vectores y números complejos.
* Soporte para multiplicación implícita
* Permite variables declaradas o no declaradas
* Compatible con Java 1.7
* Admite caracteres Unicode
* Amplia documentación y pruebas unitarias> 1400
* Localización

## Análisis y evaluación

La evaluación de una expresión de cadena con Jep es un proceso de dos pasos, como se muestra en la siguiente figura. Primero se *analiza* la expresión, llevándola de una *representación de cadena* a una *representación de árbol*. La representación de árbol es una representación estructurada de la expresión que permite una evaluación simple y rápida de la expresión en un segundo paso. Las siguientes secciones discutirán estos dos pasos con más detalle.

## Configuración por defecto

Si crea una instancia de Jep con la configuración predeterminada, puede analizar expresiones con todas las funciones enumeradas en la sección Funciones. También puede utilizar las siguientes constantes:

|  |  |
| --- | --- |
| **Pi** | 3.1415 ... La razón entre la circunferencia de un círculo y su diámetro (Math.PI) |
| **mi** | 2.718 ... Constante de Euler en doble precisión (Math.E) |
| **cierto** | La constante booleana (Boolean.TRUE) |
| **falso** | La constante booleana (Boolean.FALSE) |
| **I** | La unidad imaginaria compleja |

## Métodos de evaluación

Hay tres métodos disponibles para evaluar una expresión:

* [**Object evaluate()**](http://www.singularsys.com/jep/doc/javadoc/com/singularsys/jep/Jep.html#evaluate())**:** Evalúa la última expresión analizada y devuelve el resultado como un objeto.
* [**Object evaluate(Node root)**](http://www.singularsys.com/jep/doc/javadoc/com/singularsys/jep/Jep.html#evaluate(com.singularsys.jep.parser.Node))**:** Evalúa una expresión pasada por su nodo raíz y devuelve el resultado como un objeto.
* [**double evaluateD()**](http://www.singularsys.com/jep/doc/javadoc/com/singularsys/jep/Jep.html#evaluateD())**:** Evalúa la última expresión analizada y devuelve el resultado como un valor doble. Si el resultado no se puede convertir en un doble, se lanza una EvaluationException.

Es posible que no siempre sepa de qué tipo es el resultado de la expresión. Por ejemplo, podría ser Double, Vector, Boolean o String, según la expresión analizada. Puede usar el instanceofoperador para identificar el tipo de resultado y luego convertir el resultado en la clase apropiada.

## Variables

Las variables están representadas por la clase [Variable](http://www.singularsys.com/jep/doc/javadoc/com/singularsys/jep/Variable.html) y se almacenan en una [VariableTable](http://www.singularsys.com/jep/doc/javadoc/com/singularsys/jep/VariableTable.html) . El valor de una variable se puede establecer usando

[Jep.addVariable (nombre de cadena, valor de objeto)](http://www.singularsys.com/jep/doc/javadoc/com/singularsys/jep/Jep.html#addVariable-java.lang.String-java.lang.Object-)

y recuperado con

[Objeto Jep.getVariableValue (nombre de cadena)](http://www.singularsys.com/jep/doc/javadoc/com/singularsys/jep/Jep.html#getVariableValue-java.lang.String-)

El addVariable()método también se puede utilizar para cambiar el valor de las variables existentes. Para doublevalores, use el [método Jep.addVariable (String, double)](http://www.singularsys.com/jep/doc/javadoc/com/singularsys/jep/Jep.html#addVariable-java.lang.String-double-) y para números complejos, use el [método Jep.addVariable (String, double, double)](http://www.singularsys.com/jep/doc/javadoc/com/singularsys/jep/Jep.html#addVariable-java.lang.String-double-double-) . Otro método, [getVariable (String)](http://www.singularsys.com/jep/doc/javadoc/com/singularsys/jep/Jep.html" \l "getVariable-java.lang.String-) , devuelve el objeto Variable. Esta clase permite observar variables y ofrece un método ligeramente más rápido para obtener y establecer valores de variables.

## Asignación

La asignación permite que los valores de las variables se establezcan utilizando el = operador en ecuaciones, por lo que es posible asignar valores con expresiones como

x = 3

luego use los valores asignados en una expresión como

y = x ^ 2

y ytendrá el valor 9. La asignación está habilitada de forma predeterminada. Para deshabilitar las funciones de asignación, se debe llamar al [método setAllowAssignment (boolean)](http://www.singularsys.com/jep/doc/javadoc/com/singularsys/jep/Jep.html#setAllowAssignment(boolean)) del objeto principal de Jep. Es necesario llamar a [setAllowUndeclared (true)](http://www.singularsys.com/jep/doc/javadoc/com/singularsys/jep/Jep.html" \l "setAllowUndeclared(boolean)" \t "_blank) para que Jep analice las ecuaciones de asignación.

## Funciones logarítmicas y exponenciales

Todas las funciones aceptan argumentos de los tipos Double y Complex.

|  |  |  |
| --- | --- | --- |
| **Descripción** | **Nombre de la función** | **Nombre de la clase** |
| Logaritmo natural 1 | en (x) | [Logaritmo natural](http://www.singularsys.com/jep/doc/javadoc/com/singularsys/jep/functions/NaturalLogarithm.html) |
| Base logarítmica 10 1 | registro (x) | [Logaritmo](http://www.singularsys.com/jep/doc/javadoc/com/singularsys/jep/functions/Logarithm.html) |
| Base logarítmica 2 1 | lg (x) | [LogBase2](http://www.singularsys.com/jep/doc/javadoc/com/singularsys/jep/functions/LogBase2.html) |
| Exponencial (e ^ x) | exp (x) | [Exp](http://www.singularsys.com/jep/doc/javadoc/com/singularsys/jep/functions/Exp.html) |
| Poder 1 | pow (x) | [Energía](http://www.singularsys.com/jep/doc/javadoc/com/singularsys/jep/functions/Power.html) |

## JAVA

Java es un [lenguaje de programación](https://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n) y una [plataforma informática](https://es.wikipedia.org/wiki/Plataforma_inform%C3%A1tica) que fue comercializada por primera vez en 1995 por [Sun Microsystems](https://es.wikipedia.org/wiki/Sun_Microsystems" \o "Sun Microsystems). Hay muchas [aplicaciones](https://es.wikipedia.org/wiki/Aplicacion) y [sitios web](https://es.wikipedia.org/wiki/Sitios_web) que no funcionarán, probablemente, a menos que tengan Java instalado y cada día se crean más. Java es rápido, seguro y fiable.

Desde [ordenadores portátiles](https://es.wikipedia.org/wiki/Ordenadores_port%C3%A1tiles) hasta centros de datos, desde [consolas para juegos](https://es.wikipedia.org/wiki/Consolas_para_juegos) hasta computadoras avanzadas, desde [teléfonos móviles](https://es.wikipedia.org/wiki/Tel%C3%A9fonos_m%C3%B3viles) hasta [Internet](https://es.wikipedia.org/wiki/Internet), Java está en todas partes, si es ejecutado en una plataforma no tiene que ser [recompilado](https://es.wikipedia.org/wiki/Compilaci%C3%B3n_en_tiempo_de_ejecuci%C3%B3n) para correr en otra. Java es, a partir de 2012, uno de los lenguajes de programación más populares en uso, particularmente para aplicaciones de [cliente-servidor](https://es.wikipedia.org/wiki/Cliente-servidor) de [web](https://es.wikipedia.org/wiki/Web), con unos diez millones de usuarios reportados.

## CALCULADORA DE DERIVADAS EXPONENCIALES Y ALGORÍTMICAS

La principal funcionalidad de la calculadora es para poder resolver los problemas de funciones exponenciales y logarítmicas con la finalidad de poder tener resultados confiables.

La calculadora está hecha en Java y Netbeans, se utilizó estos programas ya que son más factibles a los resultados que queremos obtener en la calculadora.

Decidimos hacer el programa en Java debido a su interfaz gráfica, entre todos investigaron para mandar links para darnos una idea de cómo hacer el programa. Decidimos reunirnos en Meet para apoyarnos y despejar dudas que nos iban surgiendo. Nos repartimos el trabajo para que todos pudiéramos aportar algo.

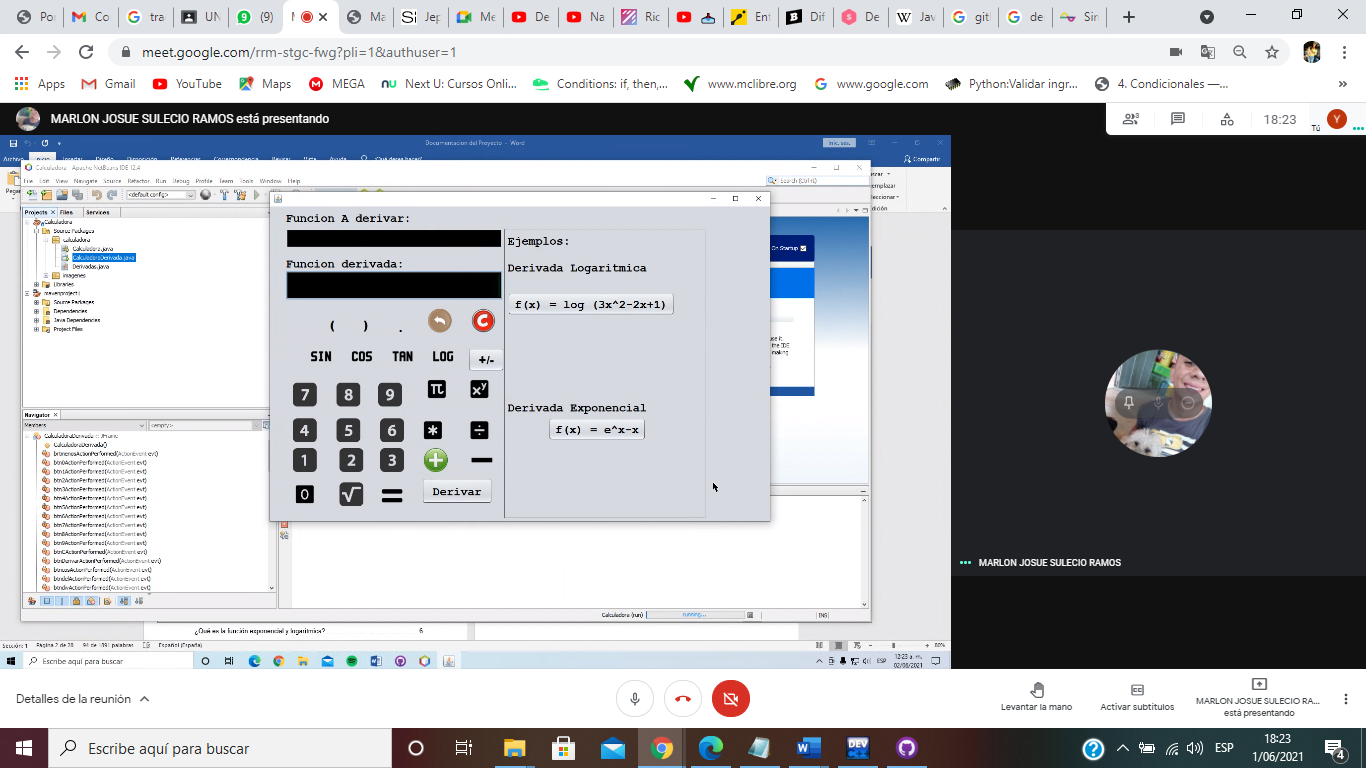
También se decidió trabajar en GitHub Desktop ya que de esta manera todos los integrantes del grupo iban poder acceder a los documentos que estamos trabajando, de esta manera cada uno podía acceder, trabajar y guardar los cambios que se habían realizado.

Al principio el programa estaba así:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Luego se decidió hacerla en forma de una calculadora para que fuera más fácil a la hora de querer utilizarla y tuviera una mejor visibilidad.

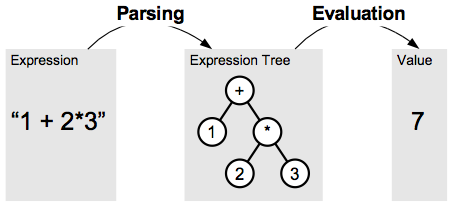


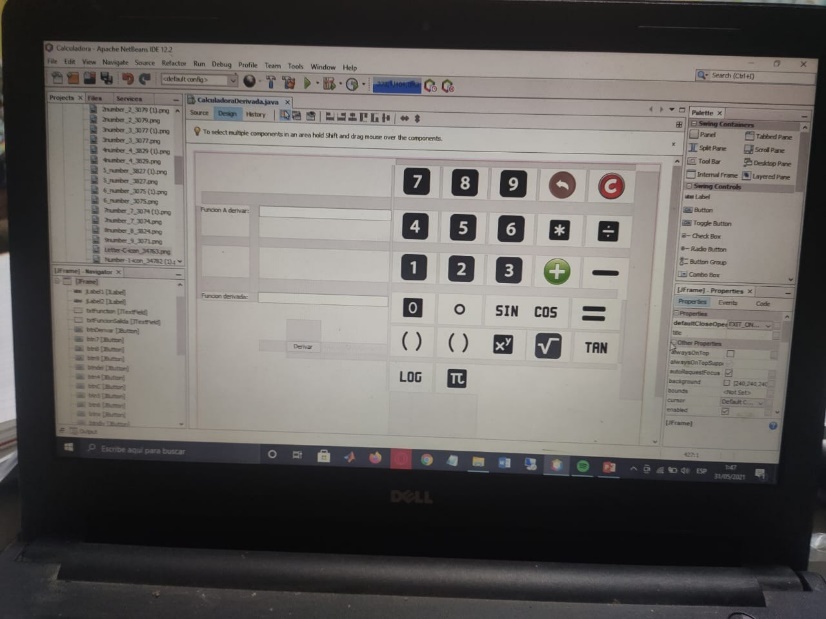
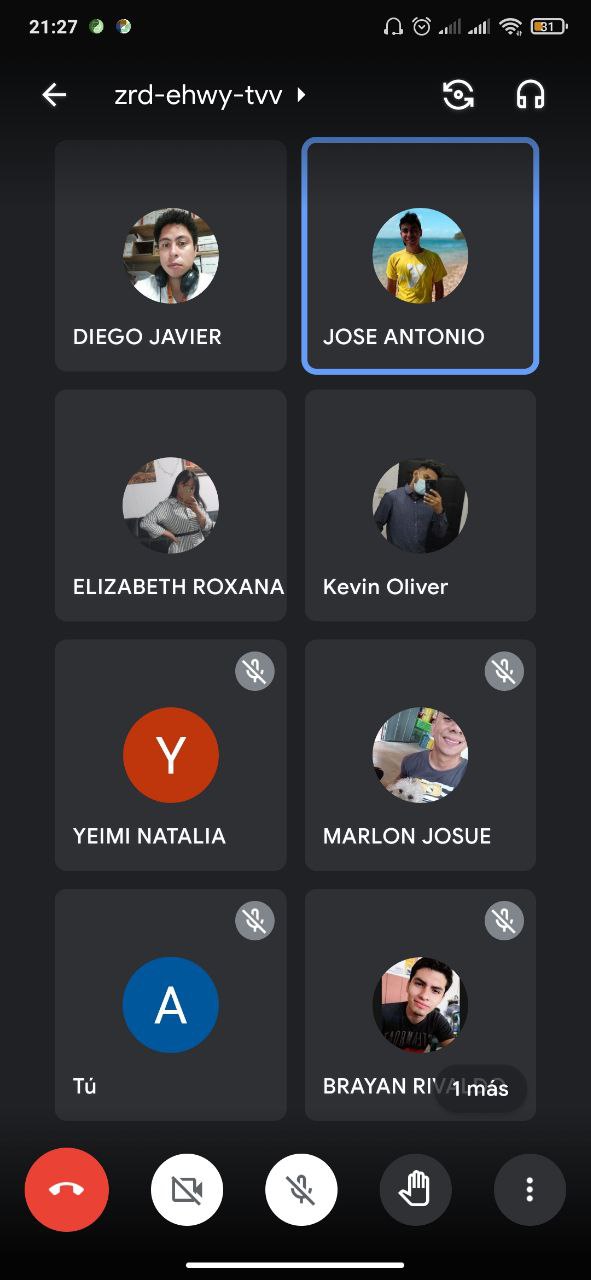
# **CONCLUSIÓN**

* Finalmente comprendimos cuales son las diferencias de las funciones exponenciales y logarítmica.
* Es claro que al poder desarrollar el programa de la calculadora nos enriquecimos de nuevos conocimientos.
* De ciertamente aportar los pocos conocimientos que tenemos en programación orientada y con la ayuda del grupo se lograron estos resultados.

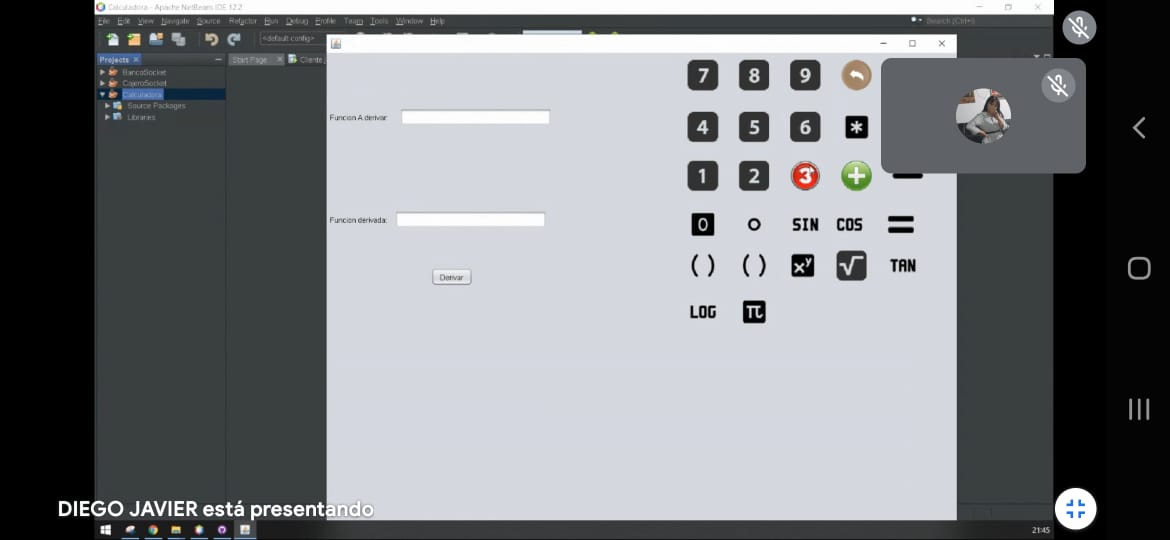
# **ANEXOS**

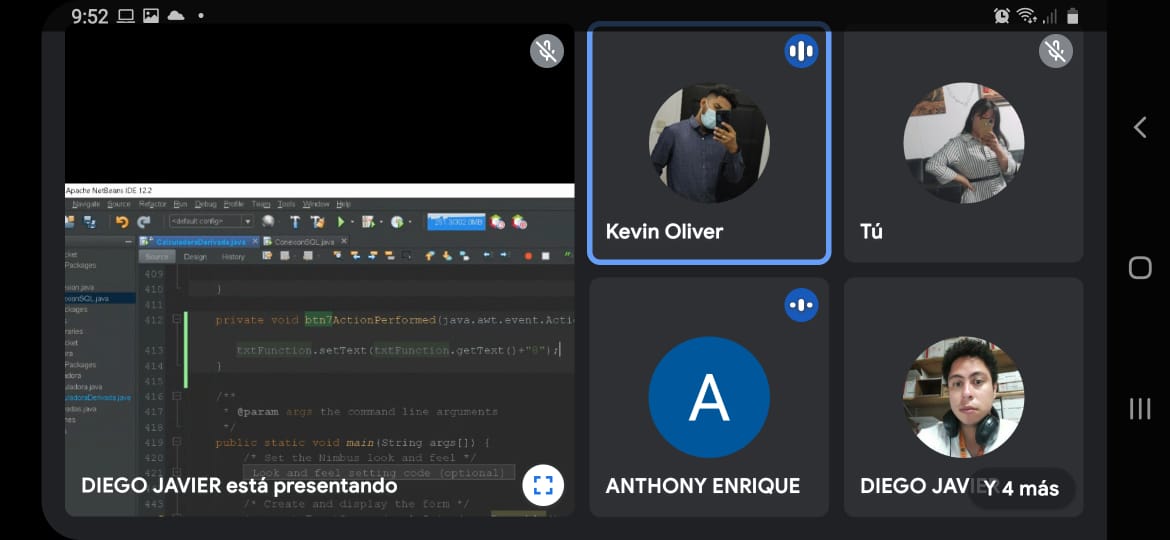
**Análisis y evaluación**

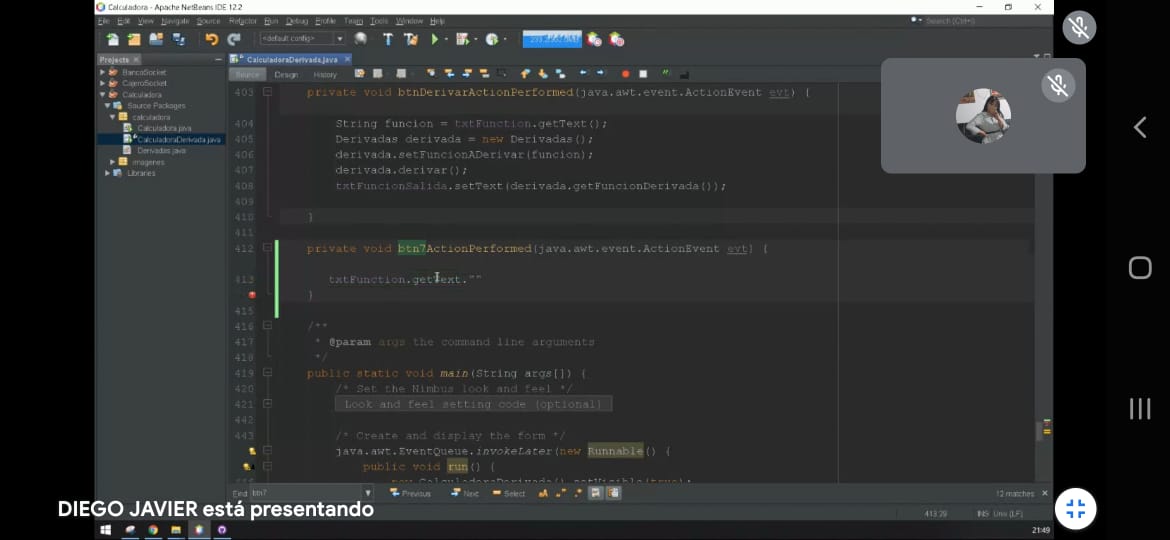


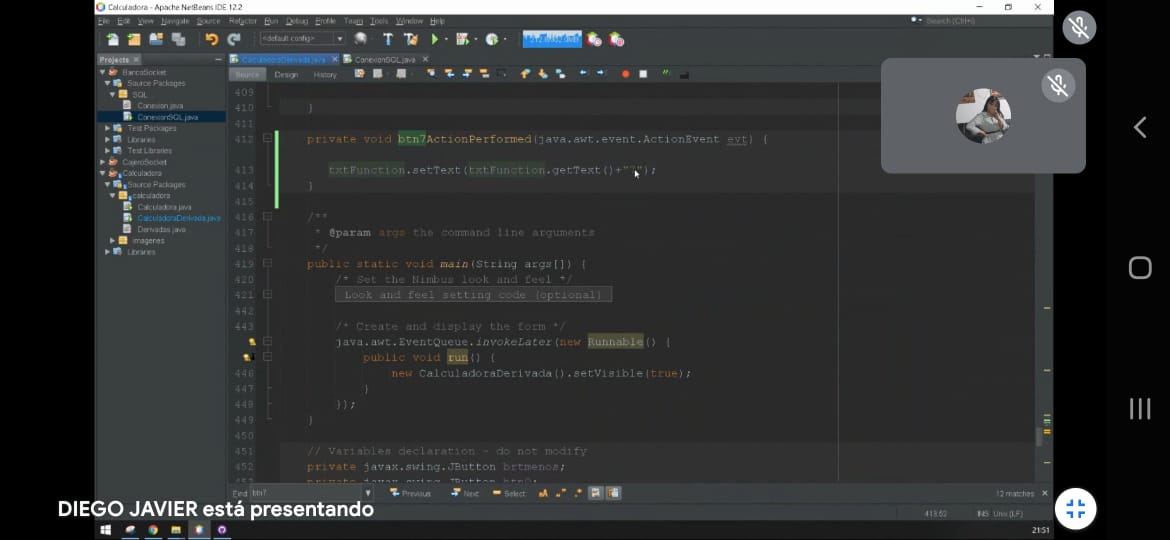


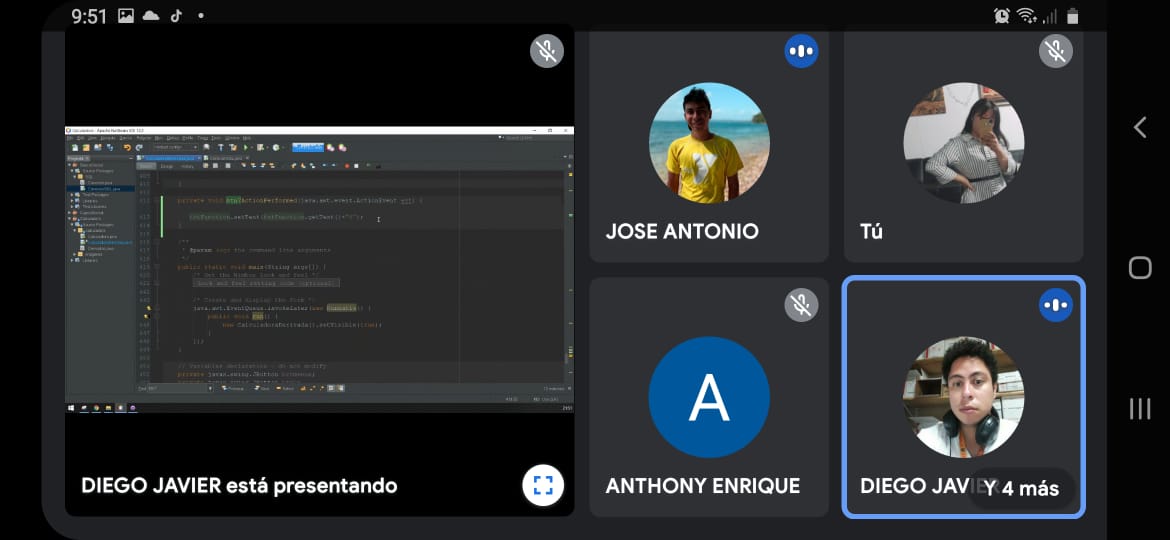
**Imágenes de avances de proyecto**

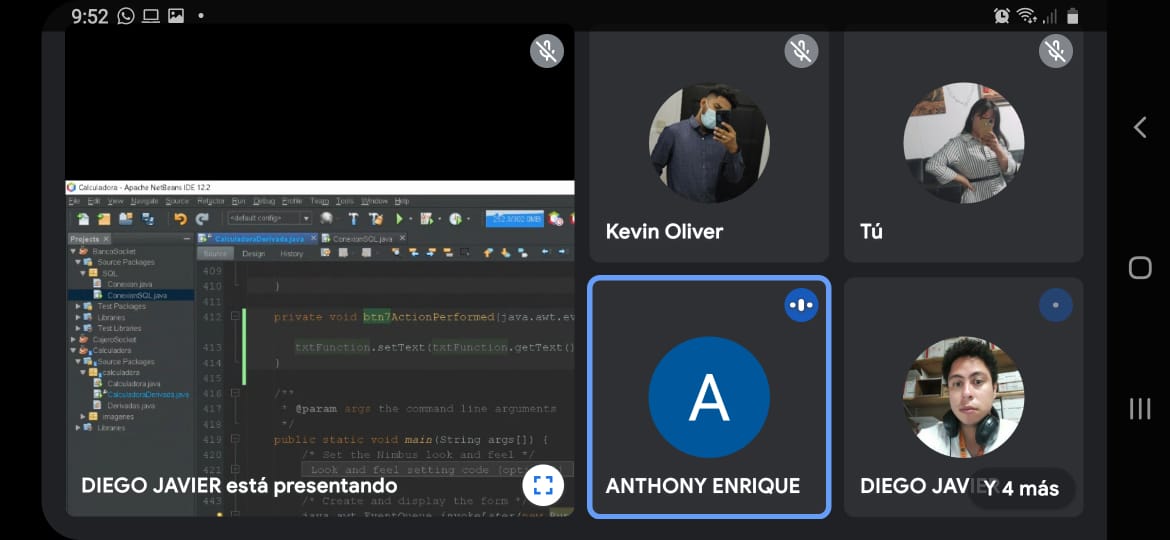
****

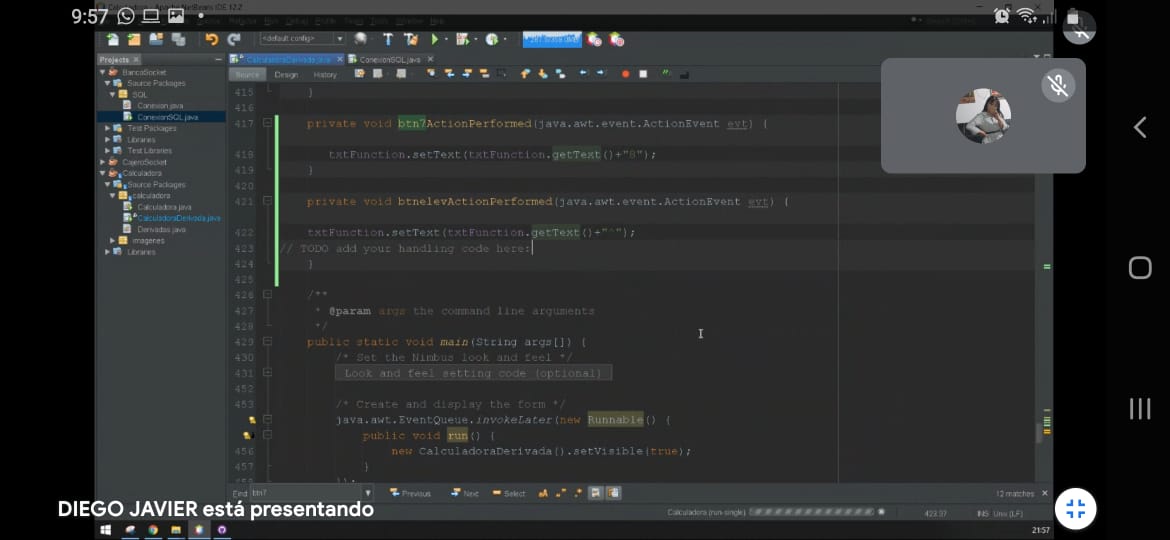


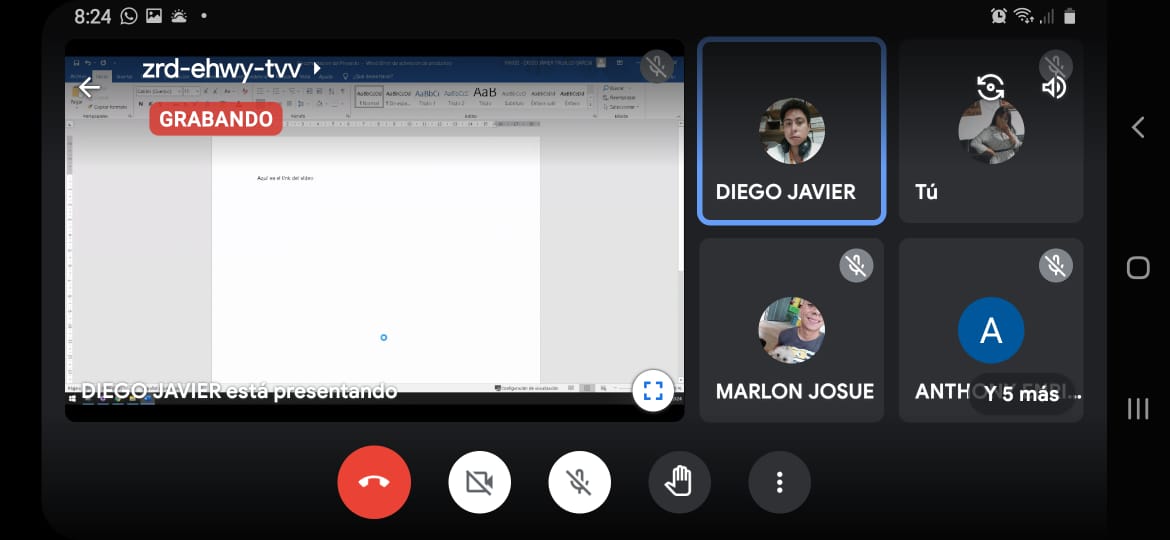


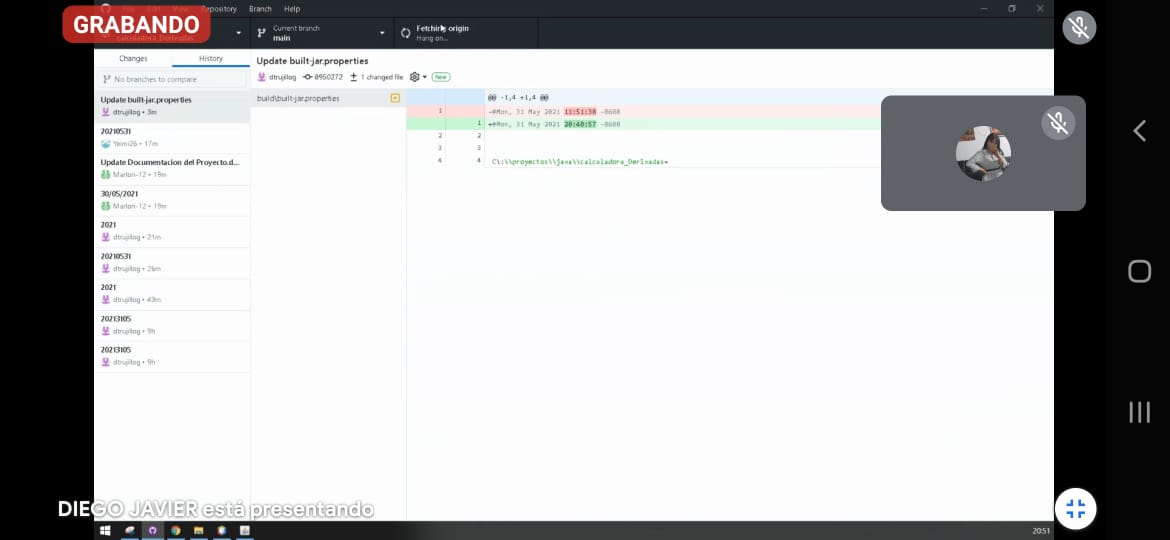


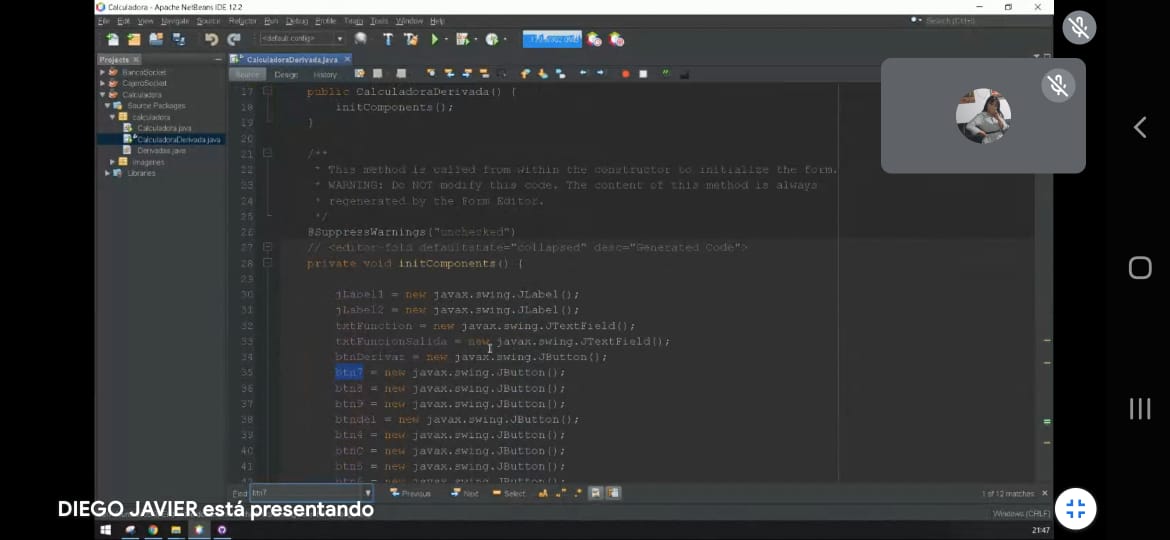


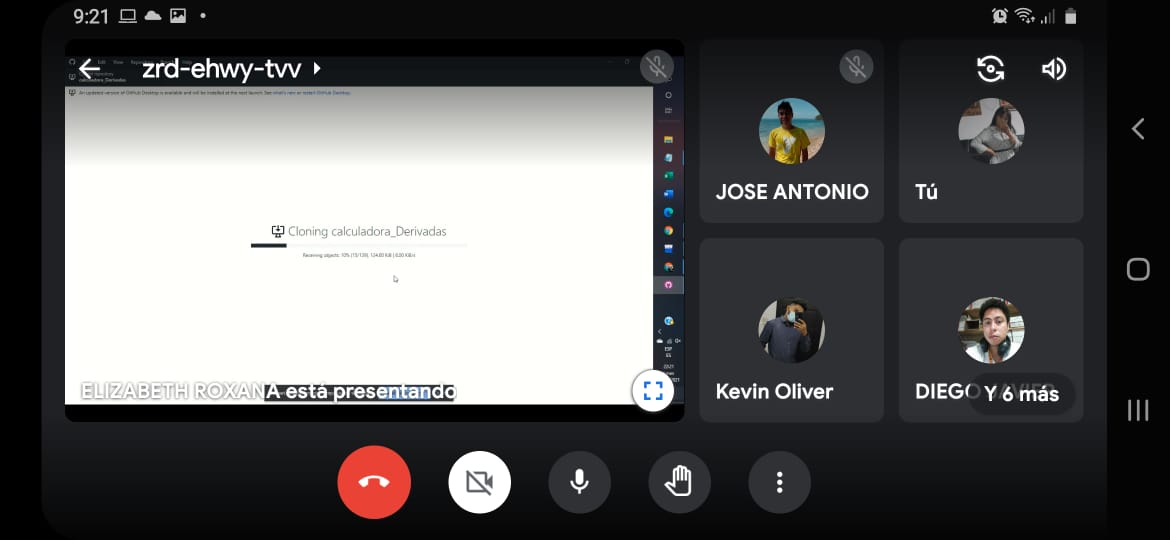


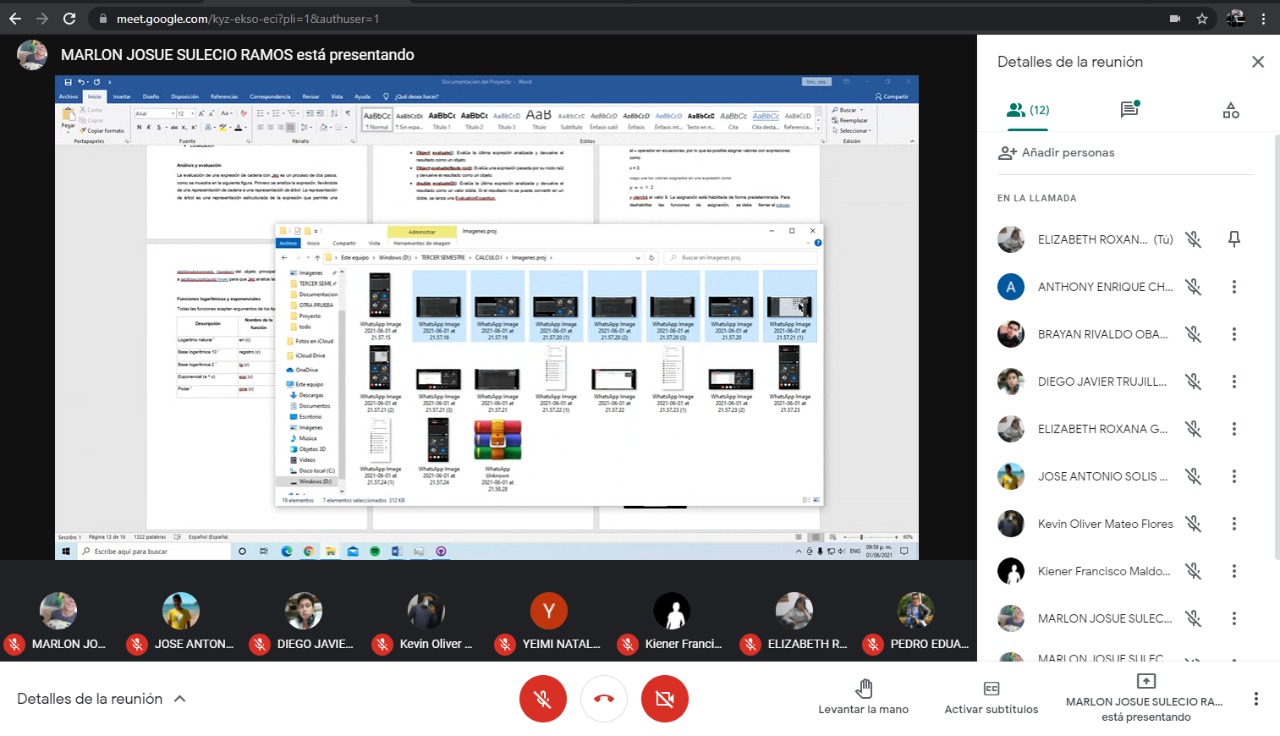


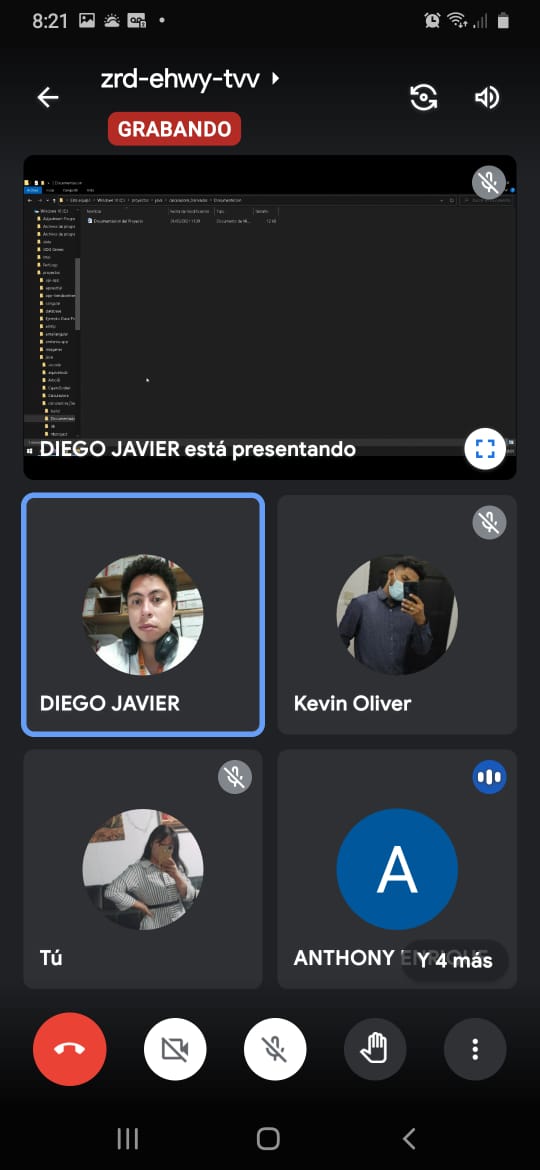
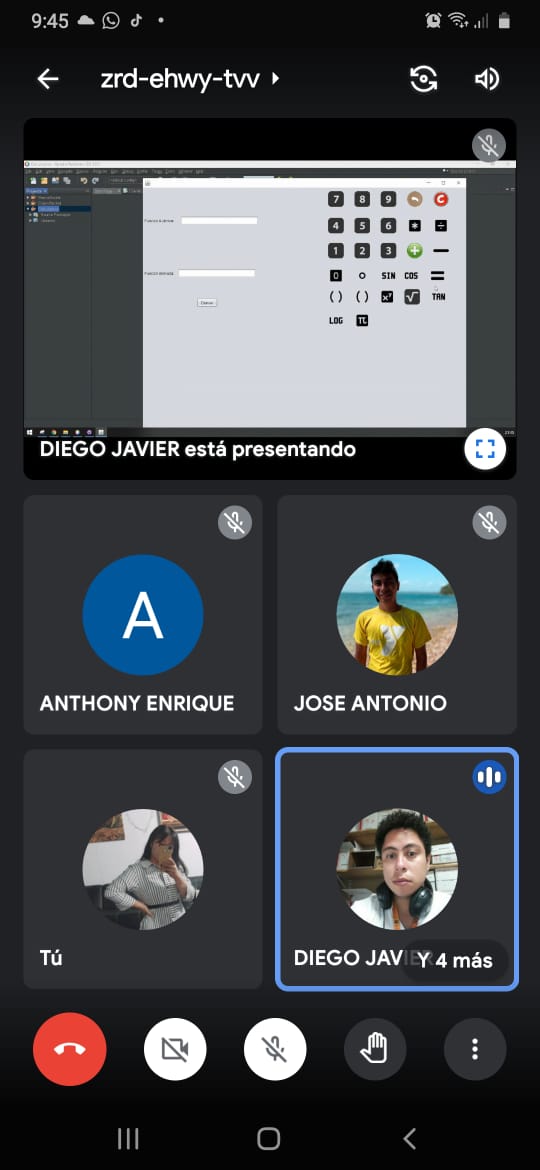


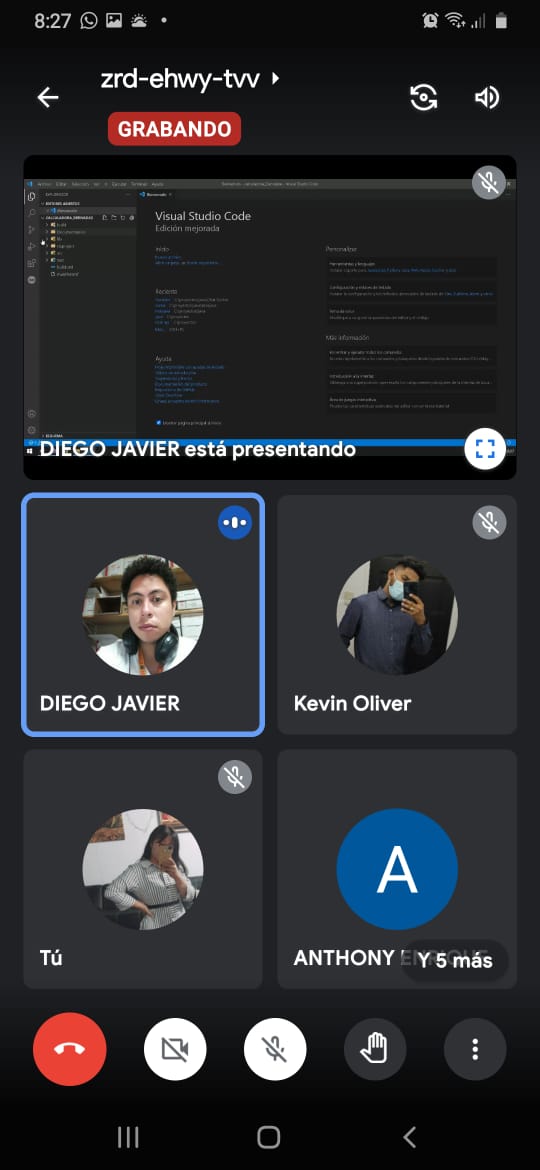




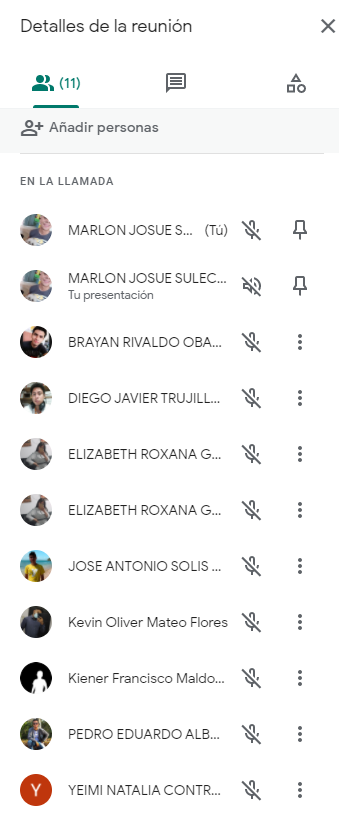
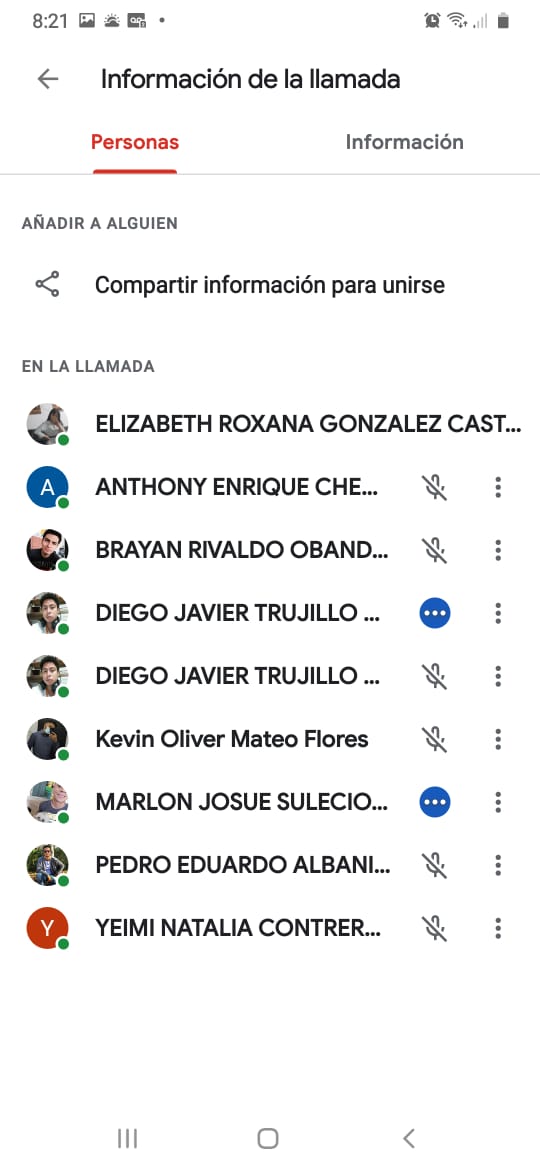








**Reuniones para proyecto**



# 

# 

# E-grafía

* <https://matematicaspr.com/l2dj/blog/funciones-exponenciales>

Funciones Exponenciales

Creado por: [l2dj](https://matematicaspr.com/blog/user/l2dj)

Día: July 28, 2012

Vistas: 163172

* Funciones exponenciales y logarítmicas fuente <https://miprofe.com/funciones-exponenciales-y-logaritmicas/>
* Jep Java - Math Expression Parser - Singular Systems. (s. f.). Jep Java. Recuperado 2 de junio de 2021, de <http://www.singularsys.com/jep/>
* https://icon-icons.com/es/

Imágenes para el diseño de botones