

# Universidad Autónoma De Baja California



Programación Orientada a Objetos

Nombre: López Mercado Brayan

Matricula: 1280838

Proyecto: Mathematical Insanity Beta v1.0


Fecha: 13/06/2020



# Descripción Del Proyecto



# ¿Qué hará el proyecto?

- El proyecto tiene la meta de ser una herramienta para estudiar y practicar los temas de ecuaciones cuadráticas, vectores, números complejos, integrales definidas y aritmética simple.
- 



# ¿Cómo funciona?

- La forma en la que funciona es haciendo uso del generador de números aleatorios, más específicamente, el constructor *Random*.
- 



# ¿Qué acciones se llevarán a cabo?

- Las acciones que se llevan a cabo de manera interna sin contar el código que genera la parte visual, contiene métodos para solucionar los problemas que el mismo programa genera, por ejemplo, existe un método para resolución de producto cruz entre vectores.




# Detalles en general, información extra, etc.

- El programa, aunque tenga una interfaz excesivamente simple, no se ve tan mal comparado a como se vería si se utilizara en la consola del IDE.
- El programa originalmente era más ambicioso en cuanto a funcionalidades, pero se descartaron por falta de tiempo y falta de conocimiento sobre cómo hacer el código para resolver el problema de manera óptima.
- El nombre de Mathematical Insanity se debe a que el programador estaba escuchando la canción *"It's Almost Easy"* de Avenged Sevenfold mientras estaba pensando en el nombre.



# ¿Cuál es la relación entre los usuarios y los datos?



- La única relación que existe entre los usuarios y los del programa son los datos que captura el usuario al momento de resolver los problemas, todos los demás datos son generados por el programa.
- 





# Perfil De Usuario



- 
- 
- Rango de edad entre 16 y 20 años
  - Nivel educativo de preparatorio o tronco común de universidad
  - El estilo de aprendizaje no influye en nada al momento de usar el programa, aunque los que sean del tipo auditivo tendrán dificultades durante largos periodos de tiempo.
  - Cualquier persona puede usar el programa, a no ser que tenga discapacidad visual.
  - La única experiencia con equipo de cómputo requerida es saber usar el teclado y el mouse para moverse en el programa.
  - Si el usuario ya hecho uso de plataformas de quiz en internet, no debería existir dificultad para utilizar el programa, en caso de que no, solo debe capturar datos en una casilla de texto, es como llenar un formulario en línea.
  - Las preferencias de los usuarios objetivos tienen como base el gusto por las matemáticas y hacer uso de programas para la realización de problemas, no tendría caso que el usuario haga uso del programa en caso de que no le agrade mucho la idea de usar un programa que genera los problemas de forma aleatoria o simplemente que no le agrade resolver problemas fuera del horario de clase.



# Usuario Jugador

- Las funcionalidades a las que tiene acceso el usuario del programa son: acceso al selector de dificultad y los tipos de problemas que se generan. No existen funcionalidades que sean exclusivas del administrador.

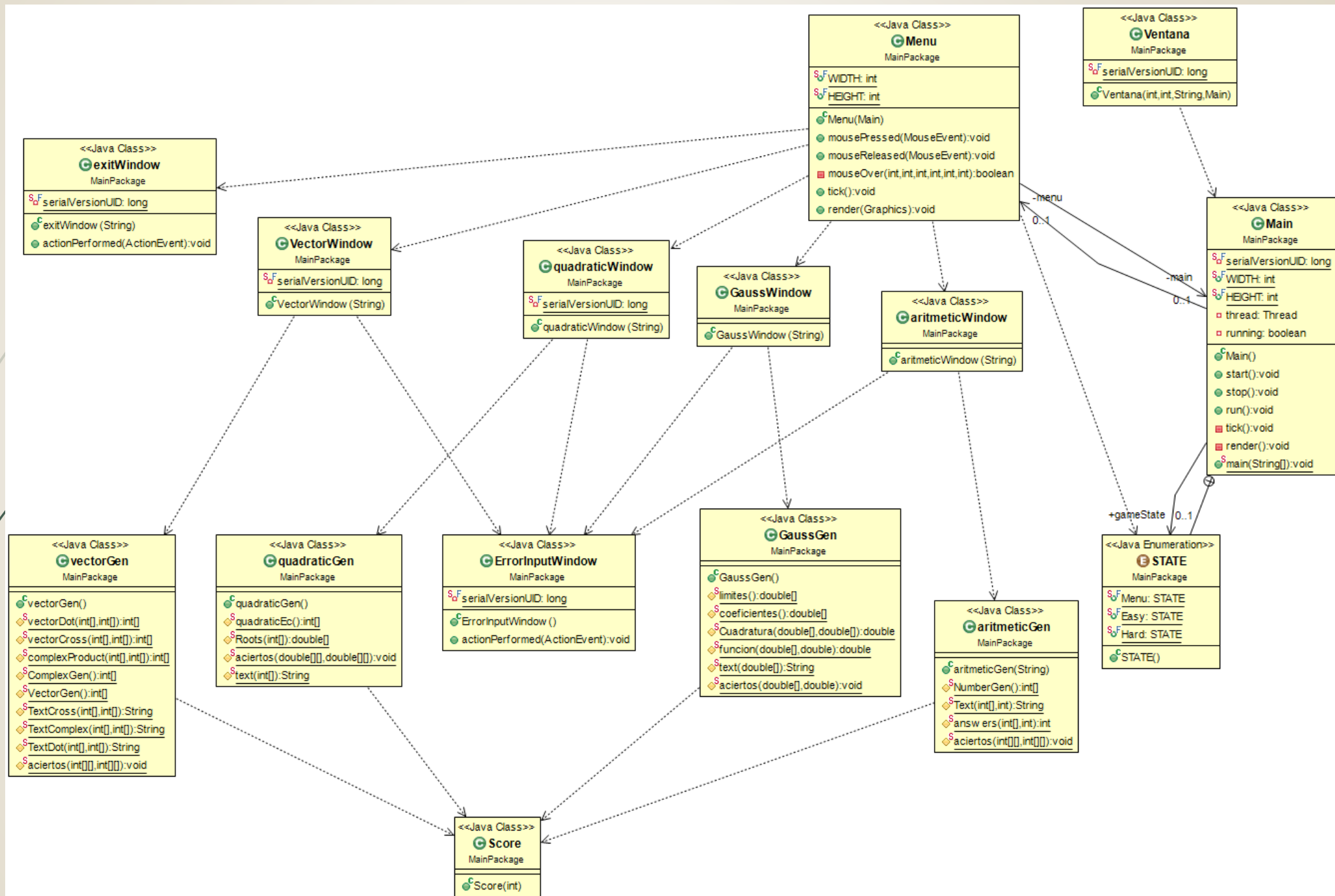


# Lógica Del Sistema

- La lógica del programa resumida de manera sencilla es: “*genero un problema y ahora resuélvelo no importa como lo hagas*”, o de una manera más detallada: se genera un vector con los números que forma parte del problema, se envían a un método que muestra el problema, se envía al método correspondiente para determinar la solución, y luego se le pide al usuario la respuesta del problema planteado y al final se muestra una ventana con la puntuación obtenida.



# Arquitectura De Clases





# Arquitectura de Objetos

- Main
  - Menú
    - VectorWindow
      - ErrorInputWindow
      - Score
    - GaussWindow
      - ErrorInputWindow
      - Score
    - QuadraticWindow
      - ErrorInputWindow
      - Score
    - AritmeticWindow
      - ErrorInputWindow
      - Score





# Conceptos de POO Utilizados

- Clases Normales
  - Interfaces
  - Jerarquía
  - Asociación
  - Polimorfismo
  - Manejo de eventos
  - Concurrencia (un solo hilo)
- 