

UNIVERSIDAD ANDINA DEL CUSCO  
FACULTAD DE INGENIERIA Y ARQUITECTURA  
ESCUELA PROFESIONAL DE INGENIERIA DE SISTEMAS



---

**CALCULADORA JAVA POO**

---

ASIGNATURA: Algoritmica y Laboratorio de Programacion II

DDOCENTE: Espetia Huamanga Hugo

INTEGRANTES:

- Cristian Del Carpio Oviedo
- Reny Luque Rios

PERU-CUSCO

2019-II

## **PRESENTACIÓN**

Los Alumnos de la Facultad de Ingenieria de Sistemas tenemos el agrado de presentar el informe a nuestro Ing. HUGO ESPETIA HUAMANGA, dando a conocer sobre la Calculadora en Java POO, ya que el inicio del curso de Laboratorio de Programacion 2 nos propone la programacion Orientada a Objetos (POO)

Java es un lenguaje orientado a objetos y para programar aplicaciones robustas, eficientes y escalables en Java es necesario entender cómo funcionan los objetos, es por eso que esta vez te voy a enseñar los 4 elementos importantes que debes conocer para entender y aplicar la Programación Orientada a Objetos en Java, también conocida también por sus siglas como POO.

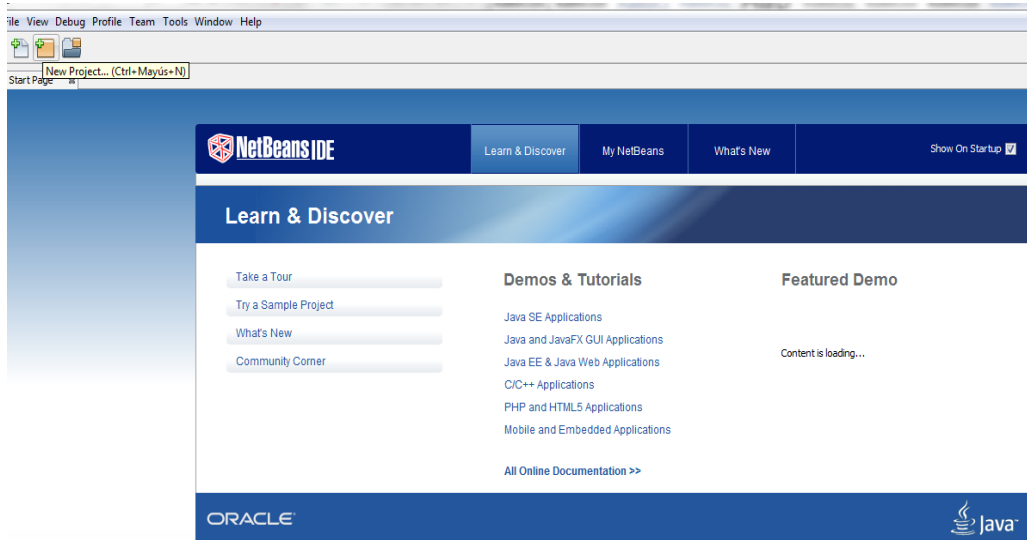
La POO es un paradigma de programación (un modelo), la cual permite relacionar los objetos del mundo real y de alguna manera llevar estos conceptos a la programación.

Y como se puede relacionar los objetos del mundo real con la programación?, pues bueno lo que se intenta es crear sistemas basados no sólo en código sino más bien pensar en objetos que se comuniquen entre sí, reutilizar código y de alguna manera modularizar un programa., donde el estudiante debe manejar estos terminos y tecnicas para el diseño y la condtruccion de algoritmos para la solucion a problemas de media complejidad.

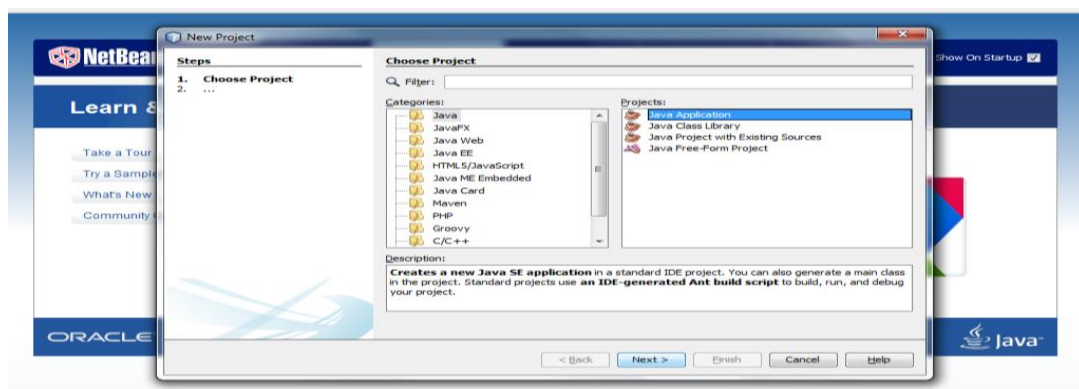
## CALCULADORA JAVA POO

### Desarrollo.

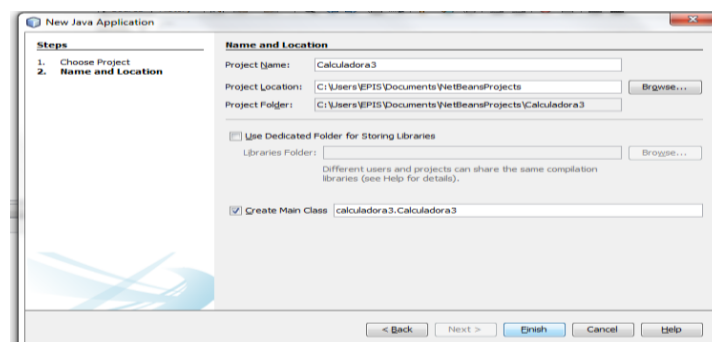
Para comensar se escoje el lenguaje de programacion, que en este caso es Java; para comensar con la creacion de la calculadora, teniendo ya en programa abierto, se escoge la opcion nuevo proyecto.



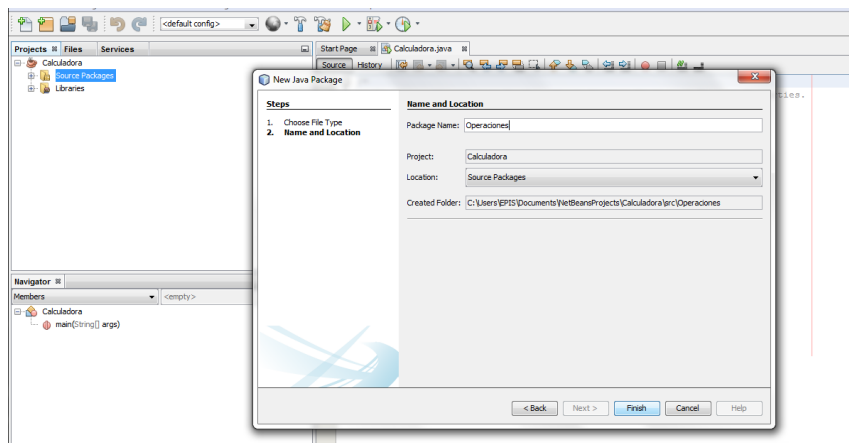
Despues escogemos la carpeta de java y junto con ella la aplicaci3n de Java Application y precionamos next.



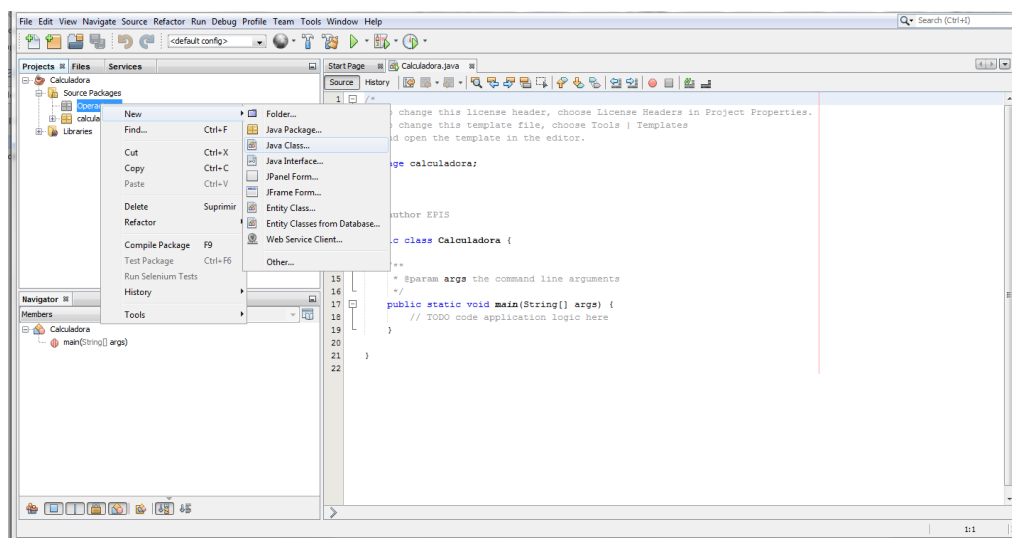
Luego colocamos el nombre del proyecto que desarrollaremos que en este caso sera la calculadora y presionamos Finish.



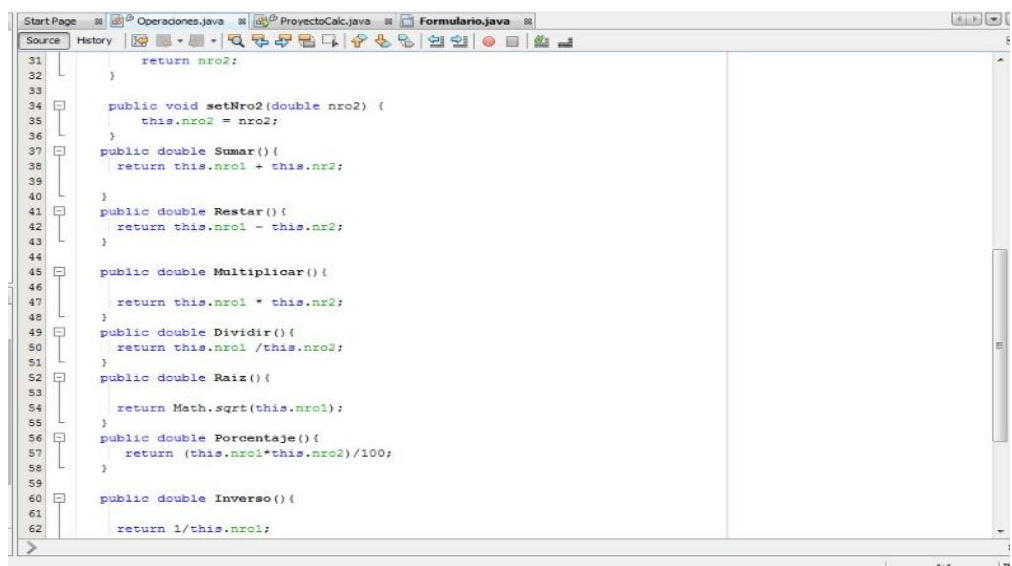
para inicializar el proceso y creacion de la calculadora en java creamos la clase Operaciones.



Declaramos la clase en java en el paquete de operaciones.

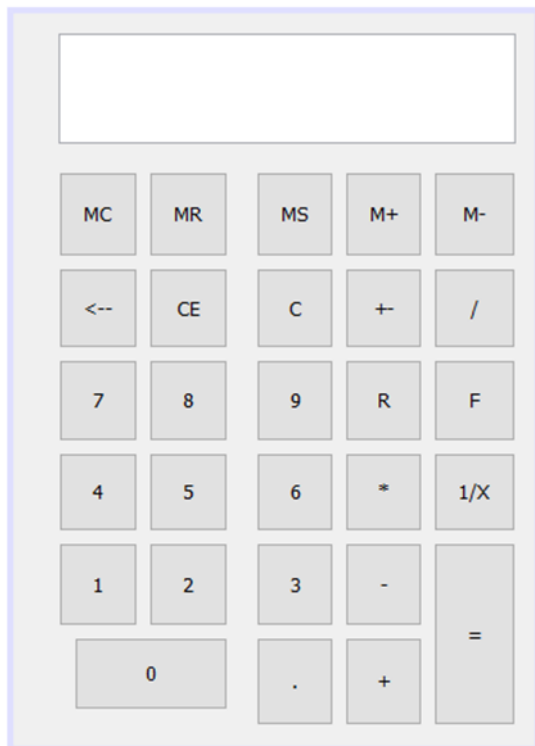


se le adiciona los atributos y metodos.



Despues se diseña la calculadora en Java Forms .

implementando botones, y espacios de lectura



Despues se diseña el codigo de cada BOTON

```
32  [+] Generated Code
350  static String operacion = " ";
351  static double nro;
352  [-] private void btn1ActionPerformed(java.awt.event.ActionEvent evt) {
353  |   txtCalculadora.setText(txtCalculadora.getText()+"1");
354  | }
355
356  [-] private void btn2ActionPerformed(java.awt.event.ActionEvent evt) {
357  |   txtCalculadora.setText(txtCalculadora.getText()+"2");
358  | }
359
360  [-] private void btn3ActionPerformed(java.awt.event.ActionEvent evt) {
361  |   txtCalculadora.setText(txtCalculadora.getText()+"3");
362  | }
363
364  [-] private void btn4ActionPerformed(java.awt.event.ActionEvent evt) {
365  |   txtCalculadora.setText(txtCalculadora.getText()+"4");
366  | }
367
368  [-] private void btn5ActionPerformed(java.awt.event.ActionEvent evt) {
369  |   txtCalculadora.setText(txtCalculadora.getText()+"5");
370  | }
371
372  [-] private void btn6ActionPerformed(java.awt.event.ActionEvent evt) {
373  |   txtCalculadora.setText(txtCalculadora.getText()+"6");
374  | }
375
```

```

376 private void btn7ActionPerformed(java.awt.event.ActionEvent evt) {
377     txtCalculadora.setText(txtCalculadora.getText()+"7");
378 }
379
380 private void btn8ActionPerformed(java.awt.event.ActionEvent evt) {
381     txtCalculadora.setText(txtCalculadora.getText()+"8");
382 }
383
384 private void btn9ActionPerformed(java.awt.event.ActionEvent evt) {
385     txtCalculadora.setText(txtCalculadora.getText()+"9");
386 }
387
388 private void btn0ActionPerformed(java.awt.event.ActionEvent evt) {
389     txtCalculadora.setText(txtCalculadora.getText()+"0");
390 }
391
392 private void btnSumarActionPerformed(java.awt.event.ActionEvent evt) {
393     nro = Double.parseDouble(txtCalculadora.getText());
394     operacion="+";
395     txtCalculadora.setText("");
396 }
397
398 private void btnRestarActionPerformed(java.awt.event.ActionEvent evt) {
399     nro = Double.parseDouble(txtCalculadora.getText());
400     operacion="-";
401     txtCalculadora.setText("");
402 }
403
404 private void btnMultiplicarActionPerformed(java.awt.event.ActionEvent evt) {
405

```

## Conclusiones:

- La calculadora hecha en Java responde a los objetivos que fueron planteados en el curso de Laboratorio de Programacion 2, al resolver problemas de mediana complejidad con programacion Orientada a Objetos.

FUNCIONALIDAD	SI	NO
Sumar	X	
Restar	X	
Dividir	X	
Multiplicar	X	
Raiz		X

## Bibliografía

Cruz, E. (6 de Marzo de 2017). *Calculadora POO con C#*. Recuperado el 21 de Agosto de 2019, de Calculadora POO con C#:

<https://programacion22.wordpress.com/2017/03/06/calculadora-poo-con-c/>

Prado, C. E. (s.f.). *Calculadora en Java con herencia de clases*. Recuperado el 21 de Agosto de 2019, de Calculadora en Java con herencia de clases:

<https://devcode.la/tutoriales/calculadora-en-java/>

## **CONCLUSIONES**

- La programacion nos permitio optimizar del codigo generado de la Calculadora gracias a que mediante tecnicas de programacion,
- El codigo que sea generado de manera que sea reutilizable.



- Logramos tener tecnicas por lo cual podemos establecer una solucion ante un problema real.

## Referencias

- Cruz, E. (6 de Marzo de 2017). *Calculadora POO con C#*. Recuperado el 21 de Agosto de 2019, de Calculadora POO con C#:  
<https://programacion22.wordpress.com/2017/03/06/calculadora-poo-con-c/>
- Prado, C. E. (s.f.). *Calculadora en Java con herencia de clases*. Recuperado el 21 de Agosto de 2019, de Calculadora en Java con herencia de clases:  
<https://devcode.la/tutoriales/calculadora-en-java/>

