

CAJA DE POLINOMIOS WEB MÓVIL



DORIAN YESID FERNÁNDEZ ESTRELLA
DIEGO FERNANDO OCORÓ STANFORD

UNIVERSIDAD DE NARIÑO
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA DE SISTEMAS
SAN JUAN DE PASTO

2015

CAJA DE POLINOMIOS WEB MÓVIL

DORIAN YESID FERNÁNDEZ ESTRELLA
DIEGO FERNANDO OCORÓ STANFORD

Trabajo presentado como requisito parcial para optar al Título de
Ingeniero de Sistemas

DIRECTOR
M.Sc. JESUS INSUASTI PORTILLA

UNIVERSIDAD DE NARIÑO
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA DE SISTEMAS
SAN JUAN DE PASTO

2015

NOTA DE RESPONSABILIDAD

Las ideas y conclusiones aportadas en este Trabajo de Grado son Responsabilidad de los autores.

Artículo 1 del Acuerdo No. 324 de octubre 11 de 1966, emanado por el Honorable Concejo Directivo de la Universidad de Nariño.

“La Universidad de Nariño no se hace responsable de las opiniones o resultados obtenidos en el presente trabajo y para su publicación priman las normas sobre el derecho de autor”.

Artículo 13, Acuerdo N. 005 de 2010 emanado del Honorable Consejo Académico.

Nota de aceptación

Firma del presidente del jurado

Firma del jurado

Firma del jurado

Firma del asesor del proyecto

San Juan de Pasto, Septiembre de 2015

DEDICATORIA

“Y sabemos que a los que aman a Dios, todas las cosas les ayudan a bien, esto es, a los que conforme a sus propósitos son llamados”, Romanos 8-28.

Quiero dedicarle este trabajo a Dios que nos ha dado la vida y fortaleza para terminar este trabajo de aplicación, por ser nuestro guía hacia el éxito y brindarnos su amor sincero.

A mis padres, quienes a lo largo de mi vida han velado por mi bienestar y educación siendo mi apoyo en todo momento. Depositando su entera confianza en cada reto que se me presentaba sin dudar ni un solo momento en mi inteligencia y capacidad.

A mi familia, por estar ahí cuando más los necesité y en especial a mi tío Paulino Ocoró, por su apoyo económico y moral durante estos años de estudio.

A mi novia y compañera Yurani Landazuri quien me apoyo y alentó para continuar en los momentos que parecía que me iba a rendir.

Diego Fernando Ocoró Stanford

DEDICATORIA

A Dios, como ser supremo, por todas las bendiciones recibidas, por ser la luz, guía y apoyo espiritual para la culminación de este trabajo de grado.

A mis padres, que siempre han estado incondicionalmente en los diferentes aspectos de mi vida, llenándome de fuerza para siempre seguir adelante, por su esfuerzo, amor, apoyo, comprensión, sobre todo por estar siempre a mi lado guiándome por el buen camino de la vida logrando que este sueño se haga realidad.

A mi familia, por todo el apoyo moral, económico, por creer siempre en mí, especialmente a mis hermanos(as) que en lazos de hermandad, amistad y amor han hecho que cada día se llene de una sonrisa en mi corazón llenándome de vitalidad para así finalizar esta etapa de mi vida.

Dorian Yesid Fernández Estrella

AGRADECIMIENTOS

Al Ingeniero Jesus Insuasti, por manifestarnos su interés en dirigir nuestro trabajo de grado.

Al profesor Oscar Fernando Soto, por contribuirnos todo el material educativo de la Caja de Polinomios y aportes precisos en los temas tratados.

Al ingeniero Vicente Chamorro, por su valioso apoyo en la construcción de este documento.

Al ingeniero Oscar Revelo, por su colaboración, revisiones puntuales y aportes precisos en este trabajo.

A la Ingeniera Sandra Vallejo, por los aportes hechos al trabajo en temas de ingeniería de software.

A nuestra Alma Mater, la Universidad de Nariño, por darnos tantos espacios y oportunidades de adquirir conocimiento.

A nuestros docentes de la Universidad de Nariño, que compartieron sus conocimientos, dentro y fuera de clase, haciendo posible que nuestra formación profesional se resumiera en satisfacciones académicas.

A nuestros compañeros de estudio, con quienes compartimos maravillosos momentos y construimos grandes cosas.

RESUMEN

Caja de polinomios *Web Móvil*, es una aplicación que permite apoyar el aprendizaje de la operatividad con polinomios, con características *multiplataforma*, *multiusuario*, *multi-acceso* y *facilidad de uso*, para la resolución del problema que se particulariza en el estudio de la operatividad elemental con polinomios y con su factorización.

En el desarrollo del proyecto fue necesario la identificación de las bondades que tiene la caja de polinomios cuando se trata de operar con ellos, se elaboró una lista de requerimientos para construir un modelo virtual de la caja de polinomios, se diseñó la arquitectura de la herramienta caja de polinomios en plataforma web y dispositivos móviles y se realizó la aplicación de acuerdo a la especificación de requerimientos y diseño arquitectónico de la caja de polinomio en plataforma web y dispositivos móviles.

El trabajo se justifica porque posee valor teórico, utilidad práctica, relevancia social, es conveniente y por los beneficios que genera. Se sustentó en las metodologías ágiles específicamente Programación extrema (*XP*), fue publicado en la Universidad de Nariño, haciendo uso de los servidores del grupo de investigación GALERAS .NET con lo que se hace ágil y eficiente el acceso a la aplicación *Web*.

Caja de polinomio *Web Móvil* se presenta como solución innovadora y referente en cuanto a la necesidad de expandir el uso de nuevas metodologías de aprendizaje, teniendo como base la tecnología y manifestándose como apoyo a los usuarios de las instituciones educativas que pertenezcan a básica secundaria.

Palabras Claves: aplicación *Web*, aplicación *Móvil*, Caja de polinomios, metodologías ágiles, multiplataforma, modelo virtual.

Abstract

Web Mobil Polynomial Box is an application which allow to support the learning of polynomial effectiveness. This application has characteristics such as cross-platform, multi-user, multi-access, and usability to problems resolution in the study of elemental polynomial effectiveness and with its factoring.

In the development of the project it was necessary the identification of the goodness at the polynomial box. A requirement list was prepared to create a virtual model of the polynomial box. Also, the architecture of the tool of polynomial box was designed in the web platform and mobile devices. In addition, the application was performed according to the specification of requirements and architectural design of the polynomial box in web platform and mobile devices.

The work is justified because of its theoretical value, practical utility and social relevance. It was based on agile and efficient methodologies such as Extreme programming (XP), it was published at Universidad de Nariño, using severes of the research group GALERAS .NET.

Web Mobile Polynomial Box is introduced as an innovative regarding solutions in terms of the need to expand the use of new learning methodologies using as a base the technology and manifesting as a support to the users in the educational institutions which belong to secondary school.

Key words: *Web application, mobile application, polynomial box, agile methodologies, multi-platform, virtual model.*

CONTENIDO

1	MARCO TEÓRICO.....	32
1.1	ANTECEDENTES	32
1.2	¿QUÉ ES UN POLINOMIO?	33
1.3	CAJA DE POLINOMIOS	40
1.3.1	Representación geométrica de polinomios	40
1.3.2	Caso de los coeficientes negativos.	42
1.3.3	Suma.....	43
1.3.4	Resta.....	46
1.3.5	Optimización del uso del plano cartesiano.	48
1.3.6	División.....	52
1.3.7	Factorización.....	55
1.4	HERRAMIENTAS MATEMÁTICAS PARA OPERAR POLINOMIOS.....	57
1.4.1	Aplicaciones web	57
1.4.2	Aplicaciones de escritorio.....	58
1.4.3	Aplicaciones móviles	58
1.5	FRAMEWORK:	58
1.6	APACHE CORDOVA:	59
1.7	JQUERY:.....	59
1.8	¿QUÉ ES UNA APLICACIÓN?	59
1.8.1	Tipos de Aplicaciones.	60
1.9	LENGUAJE DE PROGRAMACIÓN.....	63
1.9.1	Tipos de lenguajes de programación	63
1.10	PARADIGMAS DE PROGRAMACIÓN.....	64
1.10.1	Programación por procedimientos:.....	65
1.10.2	Programación modular:	65
1.10.3	Paradigma funcional:.....	65
1.10.4	Programación orientada a objetos (OOP):	65
1.11	HERRAMIENTAS DE DESARROLLO	66

1.11.1	Visual Studio (IDE):	66
1.12	SISTEMAS OPERATIVOS	67
2	DESARROLLO CAJA DE POLINOMIOS WEB MÓVIL	70
2.1	HISTORIAS DE USUARIOS	70
2.2.1	Historias de usuarios a partir del desarrollador	70
2.2.2	Historias de usuarios a partir del cliente	75
2.3.	Diagramas casos de usos	80
2.3.1.	Caso de uso Iniciar operaciones	80
2.3.2.	Caso de uso Operar	81
2.4.	Diagramas de secuencia	82
2.4.1.	Diagrama de secuencia elegir operación	82
2.4.2	Diagrama de secuencia operar	83
2.5.	Diagramas de actividades	84
2.5.1	Diagrama de actividad suma	84
2.5.2	Diagrama de actividad resta	85
2.5.3	Diagrama de actividad multiplicación	86
2.5.4	Diagrama de actividad división	87
2.5.5	Diagrama de actividad factorización	88
2.6	Código principal de la aplicación	89
2.6.1	Agregar ficha en la caja de polinomios	89
2.6.2	Establecer la propiedad draggable a las fichas de la caja de polinomios	90
2.6.3	Establecer la propiedad droppable	91
2.6.4	Traducción simbólica de polinomios	102
3	INTERFACES Y FUNCIONAMIENTO	104
3.1	MENÚ PRINCIPAL	107
3.2	OPERACIÓN SUMA	108
3.3	OPERACIÓN RESTA	112
3.4	OPERACIÓN MULTIPLICACIÓN	115
3.5	OPERACIÓN DIVISIÓN	119
3.6	OPERACIÓN FACTORIZACIÓN	123

3.7	TUTORIAL	125
3.8	ACERCA DE	126
4	CONCLUSIONES.....	127
5	RECOMENDACIONES	128
6	BIBLIOGRAFÍA	129

LISTA DE TABLAS

	Pág.
Tabla 1. Grado de un polinomio.....	34
Tabla 2. Polinomio ordenado	34
Tabla 3. Polinomio completo.....	34
Tabla 4. Aplicaciones Web Ventajas y Desventajas	61
Tabla 5. Aplicaciones nativas ventajas y desventajas	62
Tabla 6. Aplicaciones híbrida ventajas y desventajas.....	63
Tabla 7. Historia de usuario fichas caja de polinomios web móvil	70
Tabla 8. Historia de usuario espacio de trabajo	71
Tabla 9. Historia de usuario traducción simbólica.....	71
Tabla 10. Historia de usuario ayudas.....	72
Tabla 11. Historia de usuario suma caja de polinomios web móvil.....	72
Tabla 12. Historia de usuario resta caja de polinomios web móvil.....	73
Tabla 13. Historia de usuario división caja de polinomios web móvil.....	73
Tabla 14. Historia de usuario multiplicación caja de polinomios web móvil	74
Tabla 15. Historia de usuario factorización caja de polinomios web móvil.....	74
Tabla 16. Historia de usuario mover fichas.....	75
Tabla 17. Historia de usuario eliminar fichas	75
Tabla 18. Historia de usuario escoger operación.....	76
Tabla 19. Historia de usuario reiniciar operación.....	76
Tabla 20. Historia de usuario animaciones guías	77
Tabla 21. Historia de usuario suma usuario.....	77
Tabla 22. Historia de usuario resta usuario	78
Tabla 23. Historia de usuario división usuario.....	78
Tabla 24. Historia de usuario multiplicación usuario	79
Tabla 25. Historia de usuario factorización usuario	79

LISTA DE FIGURAS

	Pág.
Figura 1 – Esquema XP	29
Figura 2. Ley de signos.....	37
Figura 3. Multiplicación polinomio por monomio	37
Figura 4. Multiplicación polinomio por polinomio.....	38
Figura 5. División polinomios iteración 1	38
Figura 6. División polinomios iteración 2.....	39
Figura 7. División polinomios iteración 3.....	39
Figura 8. División polinomios iteración 4.....	40
Figura 9. Factorización simbólica polinomios.....	40
Figura 10. Rectángulos de Lilian.....	41
Figura 11. Cajas de Adrián	42
Figura 12. Tablero caja de polinomios	43
Figura 13. Suma de polinomios escribiendo $P(x)$ caja de polinomios	44
Figura 14. Suma de polinomios escribiendo $Q(x)$ caja de polinomios.....	44
Figura 15. Suma de polinomios $P(x)+Q(x)$ caja de polinomios	45
Figura 16. Resultado Suma caja de polinomios.....	45
Figura 17. Resta Escribiendo $P(X)$ caja de polinomios	46
Figura 18. Resta Escribiendo $Q(X)$ caja de polinomios.....	47
Figura 19. Resta trasladar fichas del sustraendo caja de polinomios	47
Figura 20. Resultado resta caja de polinomios	48
Figura 21. Tablero para la multiplicación caja de polinomios.....	48
Figura 22. Ubicación de las fichas Multiplicación caja de polinomios	49
Figura 23. Ubicando el primer factor $P(x)$ caja de polinomios.....	50
Figura 24. Ubicando segundo factor $Q(x)$ caja de polinomios.....	50
Figura 25. Multiplicación completando el rectángulo caja de polinomios	50
Figura 26. Multiplicación quitando ceros caja de polinomios	51

Figura 27. Multiplicación ubicando factores $P(x)$, $Q(x)$ caja de polinomios	51
Figura 28. Multiplicación quitando ceros caja de polinomios – 2	52
Figura 29. División escribiendo los polinomios	53
Figura 30. División ubicar fichas banda imaginaria.....	53
Figura 31. División escribir polinomio ejemplo 2.....	54
Figura 32. División ubicar fichas banda imaginaria ejemplo 2	54
Figura 33. División completar cuadrado y quitar ceros ejemplo 2.....	55
Figura 34. Encuadre mínimoal	56
Figura 35. Encuadre mínimoal viable.....	56
Figura 36. Encuadre mínimoal terminación ejemplo	57
Figura 37. Imagen de inicio de Windows 8.1	69
Figura 38. Caso de uso iniciar operación.....	80
Figura 39. Caso de uso operar	81
Figura 40. Diagrama de secuencia elegir operación.....	82
Figura 41. Diagrama de secuencia operar	83
Figura 42. Diagrama de actividad suma	84
Figura 43. Diagrama de actividad resta	85
Figura 44. Diagrama de actividad multiplicación.....	86
Figura 45. Diagrama de actividad división	87
Figura 46. Diagrama de actividad factorización	88
Figura 47. Icono app.....	104
Figura 48. Imagen de inicio app.....	105
Figura 49. Tablero caja de polinomio	105
Figura 50. Fichas caja de polinomio	106
Figura 51. Representación de basurero.....	106
Figura 52. Botón acción a seguir	107
Figura 53. Menú principal	108
Figura 54. Operación suma bienvenida	109
Figura 55. Escribir polinomios suma	110
Figura 56. Elimina ceros suma.....	110

Figura 57. Final eliminar ceros suma	111
Figura 58. Verificar resultado suma	111
Figura 59. Operación resta bienvenida	112
Figura 60. Escribir polinomios resta.....	113
Figura 61. Movimiento fichas $Q(x)$ a $P(x)$	113
Figura 62. Termino movimiento fichas $Q(x)$ a $P(x)$	114
Figura 63. Eliminar ceros resta	114
Figura 64. Verificar resultado resta	115
Figura 65. Operación multiplicación bienvenida.....	115
Figura 66. Escribir polinomios multiplicación	116
Figura 67. Ventana emergente explicativa multiplicación	117
Figura 68. Completar rectangulo multiplicación	117
Figura 69. Eliminar ceros multiplicación.....	118
Figura 70. Verificar resultado multiplicación	118
Figura 71. Operación división bienvenida	119
Figura 72. Armar divisor.....	120
Figura 73. Polinomio dividendo.....	120
Figura 74. Ubicar fichas franjas imaginarias	121
Figura 75. Completar rectangulo división.....	121
Figura 76. Eliminar ceros división	122
Figura 77. Verificar resultado división	122
Figura 78. Operación factorización bienvenida	123
Figura 79. Escribir polinomio factorización	124
Figura 80. Completar rectángulo factorización.....	124
Figura 81. Interfaz tutorial	125
Figura 82. Acerca de.....	126

MARCAS REGISTRADAS

.NET Framework es una marca registrada de *Microsoft Corporation*

APACHE CORDOVA es una marca registrada de *ASF (Apache Software Foundation)*.

ANDROID es una marca registrada de *Google Inc.*

UML es una marca registrada de *OMG (Object Management Group)*, consorcio dedicado al cuidado y al establecimiento de diversos estándares de tecnologías orientadas a objetos.

VISUAL STUDIO es una marca registrada de *Microsoft Corporation*.

C# es una marca registrada de *Microsoft Corporation*.

JAVA es una marca registrada de *Oracle Corporation*.

COMPUTADOR INTEL CORE I3 es una marca registrada de *Intel Corporation*

AVVIO 765 es una marca registrada de *Brightstar Corporation*

MOTO G es una marca registrada de *Motorola*.

MICROSOFT WINDOWS 8.1 es una línea de sistemas operativos registradas por *Microsoft Corporation*.

GLOSARIO

- **ALGEBRISTA:** persona que estudia, profesa o sabe el álgebra.
- **ALGORITMO:** en matemáticas, lógica, ciencias de la computación y disciplinas relacionadas, un algoritmo (del griego y latín, *dixit algorithmus* y este a su vez del matemático persa *Al-Juarismi*) es un conjunto prescrito de instrucciones o reglas bien definidas, ordenadas y finitas que permite realizar una actividad mediante pasos sucesivos que no generen dudas a quien deba realizar dicha actividad. Dados un estado inicial y una entrada, siguiendo los pasos sucesivos se llega a un estado final y se obtiene una solución. Los algoritmos son el objeto de estudio de la algoritmia.
- **API:** la interfaz de programación de aplicaciones, abreviada como *API* (del inglés: Application Programming Interface), es el conjunto de subrutinas, funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción. Son usadas generalmente en las bibliotecas de programación.
- **ARTIFICIO:** uno de los usos más extendidos de la palabra es para indicar un arte, una habilidad o el ingenio con el cual se realiza algo.
- **BIBLIOTECA:** una biblioteca es un conjunto de recursos (erg. estructuras de datos, funciones, etc.) que el usuario puede usar en sus programas. El efecto de incluir una biblioteca es básicamente extender el conjunto de recursos que provee el lenguaje, sin ninguna lógica o jerarquía subyacente, sencillamente agrego componentes nuevos a este.
- **COMPILADOR:** un compilador es un programa informático que traduce un programa escrito en un lenguaje de programación a otro lenguaje de programación. Usualmente el segundo lenguaje es lenguaje de máquina, pero también puede ser un código intermedio (*bytecode*), o simplemente texto. Este proceso de traducción se conoce como compilación.
- **CSS:** (*Cascading Style Sheets*) hojas de estilo en cascada es la tecnología desarrollada por el *World Wide Web Consortium (W3C)* con el fin de separar la estructura de la presentación.

Es una tecnología que nos permite crear páginas web de una manera más exacta, logrando resultados más amenos al usuario; estas hojas de estilo nos permiten aplicar estilos a nuestras páginas *html* por ejemplo: colores, bordes espacio entre caracteres, etc.

- **DOM:** el *Document Object Model* o *DOM* ('Modelo de Objetos del Documento' o 'Modelo en Objetos para la Representación de Documentos') es esencialmente una interfaz de programación de aplicaciones (*API*) que proporciona un conjunto estándar de objetos para representar documentos *HTML* y *XML*, un modelo estándar sobre cómo pueden combinarse dichos objetos, y una interfaz estándar para acceder a ellos y manipularlos. A través del *DOM*, los programas pueden acceder y modificar el contenido, estructura y estilo de los documentos *HTML* y *XML*, que es para lo que se diseñó principalmente.
- **ENSAMBLADOR:** el término ensamblador se refiere a un tipo de programa informático que se encarga de traducir un fichero fuente escrito en un lenguaje ensamblador, a un fichero objeto que contiene código máquina, ejecutable directamente por el microprocesador.
- **FRAMEWORK:** la palabra inglesa "*framework*" (marco de trabajo) define, en términos generales, un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar.
- **HTML:** *HyperText Markup Language* (lenguaje de marcas de hipertexto). Es el lenguaje básico de la *Web* para crear documentos y aplicaciones para que todos puedan usar en cualquier lugar.
Lenguaje de marcado y estándar de referencia para la creación de páginas web que define una estructura básica y un código llamado *HTML*.
- **LENGUAJE INTERPRETADO:** es el lenguaje cuyo código no necesita ser pre procesado mediante un compilador, eso significa que el ordenador es capaz de ejecutar la sucesión de instrucciones dadas por el programador sin necesidad de leer y traducir exhaustivamente todo el código.
- **MULTIPLATAFORMA:** es un término usado para referirse a los programas, sistemas operativos, lenguajes de programación, u otra clase de software, que puedan funcionar en diversas plataformas. Por ejemplo, una aplicación multiplataforma podría ejecutarse en *Windows* en un procesador *x86*, en

GNU/Linux en un procesador *x86*, y en *Mac OS X* en uno *x86* (solo para equipos *Apple*) o en un *PowerPC*.

- **PARADIGMA DE PROGRAMACIÓN:** es una propuesta tecnológica adoptada por una comunidad de programadores y desarrolladores cuyo núcleo central es incuestionable en cuanto que únicamente trata de resolver uno o varios problemas claramente delimitados; la resolución de estos problemas debe suponer consecuentemente un avance significativo en al menos un parámetro que afecte a la ingeniería de software.

- **PARALELEPÍPEDO:** es un prisma cuyas bases son paralelogramos y tiene seis caras paralelas dos a dos. Se clasifican en:

Paralelepípedo recto o rectangular: es aquel que posee aristas laterales perpendiculares a las bases y las caras son rectangulares.

Paralelepípedo oblicuo: varía respecto al del paralelepípedo recto solo en que la altura debe medirse en la perpendicular levantada desde el plano que contiene a base inferior hasta algún punto de la base superior.

Paralelepípedo cubo: es el paralelepípedo recto o rectangular cuyas aristas son todas iguales.

- **PLANO CARTESIANO:** es un sistema de referencia que se encuentra conformado por dos rectas numéricas, una horizontal y otra vertical, que se cortan en un determinado punto.
- **PLATAFORMA:** una plataforma es una combinación de hardware y software usada para ejecutar aplicaciones; en su forma más simple consiste únicamente de un sistema operativo, una arquitectura, o una combinación de ambos. La plataforma más conocida es probablemente *Microsoft Windows* en una arquitectura *x86*; otras plataformas conocidas son *GNU/Linux* y *Mac OS X* (que ya de por sí son multiplataforma).
- **PORTABILIDAD:** capacidad de un programa o sistema de ejecutarse en diferentes plataformas o arquitecturas con mínimas modificaciones.
- **SDK:** un *kit* de desarrollo de *software* o *SDK* (siglas en inglés de *software development kit*) es generalmente un conjunto de herramientas de desarrollo de software que le permite al programador o desarrollador de *software* crear aplicaciones para un sistema concreto, por ejemplo ciertos

- paquetes de *software*, *frameworks*, plataformas de *hardware*, computadoras, videoconsolas, sistemas operativos, etcétera.

Es algo tan sencillo como una interfaz de programación de aplicaciones o *API* (del inglés *application programming interface*) creada para permitir el uso de cierto lenguaje de programación, o puede, también, incluir *hardware* sofisticado para comunicarse con un determinado sistema embebido.

- **SOFTWARE LIBRE:** el *software* libre es la denominación del *software* que respeta la libertad de todos los usuarios que adquirieron el producto y, por tanto, una vez obtenido el mismo puede ser usado, copiado, estudiado, modificado, y redistribuido libremente.
- **XML:** *Extensible Markup Language* (Lenguaje de Marcas Extensible) es un lenguaje de marcas desarrollado por el *World Wide Web Consortium (W3C)* utilizado para almacenar e intercambiar una amplia variedad de datos en la web.

INTRODUCCIÓN

La educación Matemática se considera imprescindible para el desarrollo cultural de la humanidad, ya que es una disciplina que permite al ser humano incorporarse al mundo natural y hacer posible su desarrollo, mientras avanza de una función completamente individual a una función cognitiva social.

El docente de matemáticas, por su parte, debe tener como línea de base fundamental, el mejorar su desempeño didáctico para lograr el mejoramiento del conocimiento matemático, tan deficiente que hay en el país, como lo demuestran los resultados obtenidos por Colombia en las pruebas *PISA*¹ y por los estudiantes de educación media de Colombia en las pruebas Saber 11 del último año.

El uso creciente de las *TIC*, en especial de la *web* y de los dispositivos *móviles*, es una oportunidad que se debe aprovechar para desarrollar aplicaciones didácticas dirigidas a apoyar decididamente los métodos de enseñanza de todas las áreas del conocimiento y, en particular, el de las temidas y en ocasiones no queridas matemáticas. La enorme acogida de los dispositivos móviles y su facilidad de acceso, así como la cada vez creciente cobertura vía *Internet*, son oportunidades que no deben desperdiciarse para desarrollar aplicaciones didácticas portables, funcionales y de fácil manejo.

La Caja de Polinomios, es una herramienta física, didáctica y lúdica que permite operar con polinomios. Fue desarrollada por los profesores Fernando Soto y Libardo Jácome de la Universidad de Nariño y es referenciada más adelante en este trabajo. El dispositivo ha sido probado en instituciones locales como ITSIM y colegio San José Bethlemitas, y expuesto con éxito a nivel internacional, logrando alta acogida en la comunidad estudiantil como una nueva metodología para la resolución de operaciones con polinomios.

Este proyecto tuvo como resultado la creación de la Caja de Polinomios virtual, para plataformas web y dispositivos Móviles que pretende conjugar actividades físicas, mentales, comunicativas, emocionales y sociales, que estimulen la capacidad creativa en los estudiantes a través del uso de figuras geométricas tales como rectángulos o cuadrados que se caracterizan por las medidas de sus lados y la medida de su área y como apoyo a las metodologías de enseñanza de las matemáticas en las aulas de clase mejorando la motivación e interés de los estudiantes.

¹ ICFES. (2013). COLOMBIA EN PISA 2012 Informe nacional de resultados Resumen ejecutivo. Bogotá: ISBN de la versión electrónica: 978-958-11-0627-1.

PLANTEAMIENTO DEL PROBLEMA

En la instalación del Foro Educativo Nacional 2014 “*Matemáticas para todos y con todos*”, la Ministra de Educación Gina Parody manifestó: “Nos va mejor en tercero que en quinto en Pruebas Saber. En matemáticas, el nivel es insuficiente. Está en 20% y el avanzado en 23%, mientras que en quinto el insuficiente llega al 37% y el avanzado disminuye al 13%. En noveno lo que sucede es que nos mediocrizamos, el insuficiente ya no está en 37%, sino en 20%, pero el mínimo, que es el nivel que sigue, crece y llegamos casi al 60%, lo que significa que tenemos casi el 70% de los estudiantes en inferior y mínimo”.

Los resultados promedios por institución de las pruebas **Saber 11 2014-2**, realizados por el ICFES indican que el puntaje promedio de Matemáticas ≈ 48.89 , que no llegan ni siquiera a la mitad de los 100 puntos posibles. (ICFES, Interactivo, 2014)

En el documento **COLOMBIA EN PISA 2012 - Informe Nacional de Resultados - Resumen ejecutivo**, se lee que en los últimos resultados *PISA* 2012 participaron 65 países entre los cuales ocho son latinoamericanos (Brasil, Argentina, Colombia, Chile, Costa Rica, México, Perú y Uruguay) e indica: “En matemáticas, el puntaje de Colombia (376) es inferior a los obtenidos por 61 países y no es estadísticamente diferente de los observados en los países que obtuvieron los tres puntajes más bajos: Catar, Indonesia y Perú.” Lo que también nos muestra que dentro de los países latinoamericanos ocupamos el antepenúltimo lugar sólo por encima de Perú. (ICFES, COLOMBIA EN PISA 2012 Informe nacional de resultados Resumen ejecutivo, 2013)

En el mismo documento se afirma que: “En matemáticas, el crecimiento anual ha sido de 1,1 puntos y en ciencias de 1,8; ambos positivos pero no significativos estadísticamente. En el caso particular de lectura, el crecimiento anual fue de 3 puntos, que es estadísticamente significativo, a pesar de la desaceleración que se observa entre 2009 y 2012.” (ICFES, COLOMBIA EN PISA 2012 Informe nacional de resultados Resumen ejecutivo, 2013)

Basados en lo anterior y con la experiencia propia de varios años de estudio, así como la de muchos compañeros, atrevemos a afirmar que el aprendizaje de las matemáticas presenta una gran dificultad y que se particulariza en el estudio de la operatividad elemental con polinomios y con su factorización.

FORMULACIÓN DEL PROBLEMA

¿Cómo contribuir con los estudiantes y profesores de educación básica secundaria para que el aprendizaje de la operatividad con polinomios y su factorización sea más eficiente, amigable, didáctico, lúdico y novedoso?

SISTEMATIZACIÓN DEL PROBLEMA

- ¿Cómo identificar las funcionalidades de la Caja de Polinomios?
- ¿Cómo aprovechar las características lúdicas de la Caja de Polinomios cuando se trata de operar con ellos?
- ¿Cómo adaptar las características de la caja de polinomios a una aplicación computacional diseñada para la web y dispositivos móviles?
- ¿Cómo aprovechar el uso creciente de las TIC, en particular de la web y de los dispositivos móviles, para facilitar el aprendizaje de las matemáticas y, en particular, de la operatividad con polinomios y su factorización?

ALCANCE Y DELIMITACIÓN

La aplicación será construida para trabajar con las cuatro operaciones básicas con polinomios de hasta cuarto grado con una variable: suma, resta, multiplicación y división. De igual manera, contará con un módulo con el que se puede trabajar la factorización de un polinomio. La aplicación se desarrollará para plataformas web y dispositivos Móviles con acceso a Internet. Se espera que su uso recreará el conocimiento en torno de estas temáticas logrando mejor comunicación entre los participantes (población básica secundaria). La fluidez verbal y algorítmica se beneficiarán al recurrir a estrategias como la de constituir variables o sumar ceros, permitiendo un entendimiento claro y preciso de estos y otros recursos.

Para la realización de este trabajo se sigue la metodología *XP (Programación Extrema)*.

LÍNEA DE INVESTIGACIÓN

El trabajo pertenece a la línea de investigación: **Software y manejo de Información** y se inscribe en el grupo de investigación **GALERAS.NET** del Departamento de Sistemas, Facultad de Ingeniería de la Universidad de Nariño.

MODALIDAD

El trabajo corresponde a la modalidad **TRABAJO DE APLICACIÓN**.

OBJETIVOS

OBJETIVO GENERAL

Crear un modelo virtual de la caja de polinomios y desarrollar una aplicación *web* móvil para su manejo, que permita apoyar el aprendizaje de la operatividad con polinomios.

OBJETIVOS ESPECÍFICOS

- Identificar las bondades que tiene la caja de polinomios cuando se trata de operar con ellos.
- Elaborar listas de requerimientos para construir un modelo virtual de la Caja de Polinomios.
- Diseñar la arquitectura de la herramienta Caja de polinomios en plataforma *Web* y dispositivos *Móviles*.
- Desarrollar la aplicación de acuerdo a la especificación de requerimientos y el diseño arquitectónico de la Caja de Polinomio en plataforma *Web* y dispositivos *Móviles*.

JUSTIFICACIÓN

Los resultados obtenidos por Colombia en las pruebas *PISA* y por los estudiantes de educación media de Colombia en las pruebas Saber 11 en el último año indican una grave deficiencia en el conocimiento matemático y, por supuesto en el proceso de aprendizaje.

La caja de polinomios es una alternativa didáctica, lúdica y divertida que ayuda a fortalecer la enseñanza y aprendizaje del manejo de operaciones con polinomios tanto a profesores como alumnos, una prueba de esto son los resultados arrojados por el estudio al implementarla en el colegio ITSIM de la ciudad de Pasto, en el grado octavo (8-1, 8-2), de los cuales un grupo fue denominado “grupo de control” que recibía sus clases normalmente y el segundo grupo fue denominado “grupo experimental” respectivamente, en cada uno de ellos se tuvo la intensidad horaria de cuatro horas semanales. Al final, el estudio indicó que los resultados del grupo de control eran tan satisfactorios al igual que los obtenidos por el grupo experimental con una excepción en los resultados, en operaciones más complejas como la de división se obtuvo mejor resultado el grupo experimental el cual desarrolló la temática con la caja de polinomios.

Otra evidencia de la los buenos resultados obtenidos con el manejo de operaciones con polinomios utilizando la caja de polinomios es el resultado de la investigación con grupos con discapacidad auditiva y personas sin ninguna discapacidad o limitante físico, en el cual a nivel general se obtuvieron resultados positivos tanto así que los resultados obtenidos por los estudiantes con discapacidad auditiva fueron similares a los obtenidos por los estudiantes sin discapacidad.

En la actualidad los dispositivos móviles están abarcando el mercado mundial, en el diario vivir la mayoría de las personas cuentan con uno haciendo uso de diversas aplicaciones que le ayuden a agilizar diferentes procesos, tales como: comunicación, cálculos matemáticos, entretenimiento, entre otros; esto debido a su funcionalidad y portabilidad ya que se tiene acceso desde cualquier lugar. En vista de lo anterior las tecnologías móviles y el uso creciente de las *TIC* en especial la *Web*, es una oportunidad para difundir recursos educativos y procesos que ayuden a fortalecer la enseñanza en las aulas de clase.

La existencia mínima de aplicaciones lúdicas y amenas que ayuden a fortalecer y promover el interés por el aprendizaje de la operatividad con polinomios, ha llevado a la necesidad de sistematizar estos procesos, mediante el desarrollo de

aplicaciones didácticas las cuales puedan solventar las carencias en la conducción de los procesos antes referidos orientados a la educación básica secundaria.

Por lo anterior, desarrollar **La Caja de Polinomios** como una aplicación para plataformas web y dispositivos móviles contribuirá a estudiantes y profesores en el aprendizaje de la operatividad con polinomios.

METODOLOGÍA

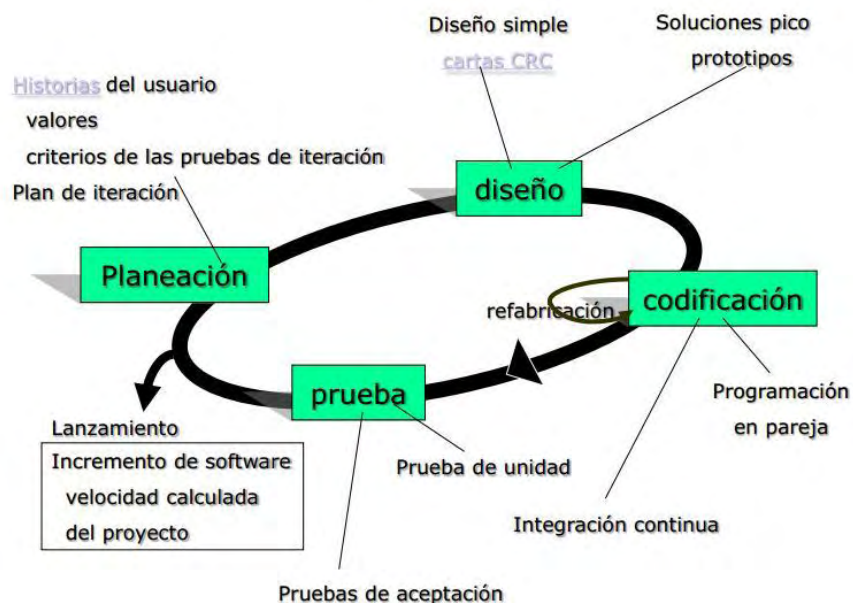
Para la realización de este proyecto se tuvieron en cuenta las siguientes metodologías: *XP*, *RUP* (), *SCRUM*, *PSP*, *TSP* de las cuales se eligió *XP* por ser una de las más utilizada en el desarrollo Ágil y adaptarse a las condiciones expuestas del trabajo.

Introducción a *XP*

XP resalta una serie de valores y principios que deben tenerse en cuenta y practicar durante el desarrollo del trabajo.

En la figura 1, Planeación, Diseño, Codificación y Prueba representan las disciplinas que agrupan actividades por su naturaleza, con el objetivo de satisfacer al cliente.

Figura 1 – Esquema *XP*



A continuación se indica las actividades más importantes de cada una de las fases de la metodología *XP*.

Planeación

1. Realizar historias de usuarios con el objetivo de identificar cuáles son los principales requerimientos del proyecto.
2. Realizar un *Release Planning*; es decir, una proyección con el fin de determinar los tiempos de ejecución ideales para el proyecto.
3. Determinar el número de Iteraciones a realizar en toda la ejecución del proyecto.
4. Determinar la velocidad de realización del proyecto.
5. Determinar las parejas de trabajo, esto con el objetivo de codificar y analizar al mismo tiempo lo que permite obtener una aplicación con gran calidad.
6. Determinar las fechas de las reuniones las cuales pueden ser diarias o día por medio con el fin de que los desarrolladores expongan sus problemas.

Diseño

1. Realizar un diseño simple, que permita tener una descripción del proyecto fácilmente entendible.
2. Realizar un glosario de términos que permita especificar los nombres de métodos y clase.
3. Determinar riesgos en el proyecto.
4. Determinar funcionalidades extras.
5. Realizar una refactorización del proyecto con el objetivo de mejorar las estructuras y codificación ya creadas sin alterar su funcionalidad.
6. Determinar las tarjetas C.R.C.

Codificación

En esta etapa se desarrollan las historias de usuarios que fueron planteadas en la planificación y posteriormente diseñadas teniendo en cuenta que se requiere de la aprobación del cliente en cuanto a la funcionalidad de cada una de estas. Para lograr con éxito esta etapa se requiere:

1. Realizar un *test* que permita probar el correcto funcionamiento de la historia de usuario a codificar.

2. Abrir un repositorio de trabajo donde las parejas de programadores publiquen cada pocas horas sus códigos implementados y corregidos junto a los *test* que deben pasar.

Prueba

En esta fase del proyecto se tendrá en cuenta los *test* para comprobar el funcionamiento de la codificación que se halla implementado, esto permitirá evaluar las distintas tareas en las que ha sido dividida la historia de usuario. Para asegurar el funcionamiento final de una determinada historia de usuario se crearan, *Test* de aceptación; estos *test* son creados y usados por los clientes para comprobar que las distintas historias de usuario cumplen su cometido.

1 MARCO TEÓRICO

1.1 ANTECEDENTES

En la actualidad se encuentran diferentes herramientas que permiten llevar a cabo la operatividad con polinomios y su factorización. En esta sección se referenciará algunos de estos elementos con el propósito de acoger experiencia y conocimiento que sirvan como base para esta investigación.

En el ámbito nacional

En Colombia se han hecho trabajos que permiten la operatividad con polinomios, entre ellos suma, resta, multiplicación y su factorización. Algunos de estos se referencian a continuación:

Caja de Polinomios: es un proyecto colectivo de docentes y estudiantes de la Universidad de Nariño encabezada por Oscar Fernando Soto y Libardo Jácome² cuyo origen se establece en el siglo X, con los trabajos del árabe *Tabit ben Querra* el *Harani*; su principal fundamento es la resolución de problemas algebraicos, específicamente operatividad con polinomios y su factorización, dando a conocer las ventajas que trae su uso, al ser acogido por las comunidades estudiantiles tanto de bachillerato como de los primeros semestres de las universidades.

En este libro se puede encontrar explicación y ejercicios resueltos de los algoritmos de factorización, multiplicación, adición, división, y sustracción con la nomenclatura propia de la Caja de Polinomio.

Para el presente trabajo, Caja de Polinomios, ayuda a manera de guía en la resolución de la línea de investigación ya que resuelve problemas como: construir fichas equivalentes en área a otra de diferentes dimensiones, operatividad con polinomios y su factorización de una variable y hasta grado tres entre otros.

Caja de polinomios *web* v.1.1: es un sitio *web* de apoyo al estudio de la operatividad con polinomios y su factorización desarrollado por *Edwin Insuasty*, *Oscar Fernando Soto* y *Jesús Insuasti*,³ adscritos a los grupos de Investigación *GESCAS* y *Galeras.NET* de la Universidad de Nariño. Esta herramienta tiene sus bases en la Caja de Polinomios y uno de sus principales objetivos es servir como recurso para el trabajo dentro y fuera del aula ya que juegan un papel esencial para despertar sentimientos y actitudes positivas hacia las matemáticas, para desmitificarlas, para propiciar la participación, la integración y vencer los

² JACOME, Libardo. SOTO, Oscar, Caja de Polinomios. Universidad de Nariño, Colombia.

³ INSUASTY, Edwin. SOTO, Oscar, INSUASTI, Jesús Caja de Polinomios Web v.1.1.

<<http://190.254.4.13/CajaPolinomiosWeb/About.>> [Consultado el 24 de marzo 2015]

obstáculos emocionales responsables del aburrimiento; permitiendo ver que las matemáticas son una materia viva, llena de interés y muy útil dentro y fuera del aula.

En este sitio *Web* se puede encontrar una breve descripción de cada una de las fichas y elementos que se utilizan para la operatividad con polinomios y su factorización además de la sección *DEMO* que permite a las personas realizar operaciones a partir de los polinomios ingresados.

Caja de polinomios *Web v1.1*, es un gran aporte al presente trabajo porque permite evidenciar el posicionamiento de los elementos en la *Web*, brinda una orientación de cómo se puede hacer uso de una interfaz de usuario fácil de manejar y amigable ante los ojos del usuario final sin dejar de lado el proceso de desarrollo de las operaciones, además, afianza los conocimientos respecto a operatividad entre polinomios y su factorización.

En el ámbito regional

En la región se encuentra:

AL – KHWARIZM⁴: es un trabajo de grado realizado por los señores Iván Mauricio Argote Puetaman y Maicol Esneider Burbano Iguá en la Universidad de Nariño, enfocado a la implementación de un sistema de apoyo de los procesos de enseñanza y aprendizaje del álgebra de polinomios utilizando la caja de polinomios haciendo uso de las técnicas de aprendizaje como lo son la lúdica y el juego, que contribuye al desarrollo del aprendizaje en el área de las matemáticas.

La mayor diferencia entre los software nombrados anteriormente y la Caja de Polinomios *Web Móvil*, es que la Caja de Polinomios *Web Móvil* abstrae las funcionalidades lógicas del juego en la *Web* para que puedan ser reutilizadas en cualquier plataforma como lo son *Android*, *IOs*, *Windows Phone* entre otras y estén disponibles en cualquier momento.

1.2 ¿QUÉ ES UN POLINOMIO?

Es una expresión algebraica que la constituye un número finito de términos (variables y constantes o coeficientes), un polinomio en si está compuesto de un conjunto de monomios, suele nombrarse con letras mayúsculas (empezando por la P) y se incluye entre paréntesis las variables que éste tiene como por

⁴ ARGOTE, Iván. BURBANO, Maicol, Alkharizmi, Sistema para el apoyo de los procesos de enseñanza – Aprendizaje del álgebra de los polinomios utilizando la caja de polinomios, dirigidos a estudiantes de educación secundaria, basado en la lúdica y el juego, Universidad de Nariño, Pasto, 2010.

ejemplo: $P(x, y) = 2 + 6y$; los polinomios pueden tener más de una variable $15x+2y+4$ es un polinomio con dos variables.

Grado de un polinomio: el grado de un polinomio es el mayor de los grados de sus monomios que lo componen. En la tabla 1 se indica ejemplos del grado de un polinomio.

Tabla 1. Grado de un polinomio

Polinomio	Grado
$P(x) = 2x^2 - 3x + x^5 + 1$	5
$Q(a, b) = -4ab^2 + 2a^2b^2 - a^2 + b^3$	4
$R(g, m, t) = 6m^2t^4 + 3g^2mt - 8g + m$	6

Polinomio ordenado: se denomina polinomio ordenado si cada uno de sus monomios que lo componen están ordenados de mayor a menor grado. En la tabla 2 se indica un ejemplo de polinomio ordenado.

Tabla 2. Polinomio ordenado

Polinomio	Polinomio ordenado
$P(x) = 3x^3 - 3x^2 + x + 12$	Si
$Q(y) = 3y^2 + 2y^5 - y^3 + 3$	No

Polinomio completo: se denomina polinomio completo al polinomio que tenga todos los términos desde el grado mayor hasta el grado cero. En la tabla 3 se indican un ejemplo de polinomio completo.

Tabla 3. Polinomio completo

Polinomio	Polinomio completo
$P(x) = 3x^4 - 3x^3 - 8x^2 + x + 2$	Si
$Q(y) = 2y^5 + 3y^3 + 2y^2 + 8$	No

Los polinomios y precisamente las ecuaciones polinómicas como las funciones polinómicas tienen aplicaciones en una gran variedad de problemas, desde la matemática elemental y el álgebra hasta áreas como la física, química, economía y las ciencias sociales.

En el estudio de los polinomios para este trabajo se hará énfasis en aquellos polinomios compuestos por una sola variable y hasta de grado 4.

Operaciones entre polinomios:

Suma: la suma de un polinomio no difiere a la suma normal de dos cantidades; cabe destacar que la suma de las cantidades para polinomios se hace entre los coeficientes que acompañan las variables es así que se pueden sumar términos cuyo exponente sea de igual valor es decir cuadrados con cuadrados cubos con cubos etc., coeficientes sin variable con aquellos que estén sin variable.

Ejemplo: sea: $P(x) = 3x^3 - 3x^2 + x + 12$ y $Q(x) = 8x^3 - 2x^2 + x + 10$

$$\begin{array}{r} 3x^3 - 3x^2 + x + 12 + \\ 5x^3 - 2x^2 + x + 10 \\ \hline 8x^3 - 5x^2 + 2x + 20 \end{array}$$

Propiedades de la suma de polinomios:

A. Ley de cierre

La suma de dos o más polinomios da otro polinomio. Esto quiere decir que si se suma $P(x) + Q(x) = R(x)$

B. Propiedad asociativa:

Dada la suma de tres polinomios $P(x), Q(x), R(x)$ cumple la propiedad asociativa, si:

$$P(x) + Q(x) + R(x) = (P(x) + Q(x)) + R(x) = P(x) + (Q(x) + R(x))$$

C. Elemento neutro

Si dado un polinomio $P(x)$ se le suma 0 el resultado será el mismo $P(x)$

$$P(x) + 0 = 0 + P(x) = P(x)$$

D. Propiedad conmutativa

El orden de los polinomios no afecta o altera el resultado, sean los polinomios $P(x)$ y $Q(x)$ cumplen la propiedad conmutativa, si:

$$P(x) + Q(x) = Q(x) + P(x)$$

E. Existencia del elemento opuesto

Si a $P(x)$ se suma su opuesto $-P(x)$ el resultado es cero

Resta entre polinomios: al igual que la suma la forma de restar polinomios es agrupando o identificando términos semejantes para proceder a sumar cantidades de dichos términos; tener en cuenta que para la resta el sustraendo debe cambiar sus signos para luego proceder a sumar.

Ejemplo: sea: $P(x) = 3x^3 - 12x^2 + 5x - 10$ y $Q(x) = -5x^3 + 8x^2 + 13x - 5$

$$\begin{array}{r} 3x^3 - 12x^2 + 5x - 10 \\ - (5x^3 - 8x^2 - 13x + 5) \\ \hline 8x^3 - 20x^2 - 8x - 5 \end{array}$$

Multiplicación con polinomios: para la multiplicación con polinomios se tiene las siguientes propiedades importantes:

A. Propiedad asociativa

Dada la multiplicación de tres polinomios $P(x), Q(x), R(x)$ cumple la propiedad asociativa, si:

$$P(x) * Q(x) * R(x) = (P(x) * Q(x)) * R(x) = P(x) * (Q(x) * R(x))$$

B. Elemento neutro

El producto de un polinomio $P(x)$ por la unidad su resultado será el mismo $P(x)$.

$$P(x) * 1 = P(x)$$

C. Propiedad conmutativa

El orden de los polinomios no afecta o altera el resultado, sean los polinomios $P(x)$ y $Q(x)$ cumplen la propiedad conmutativa, si:

$$P(x) * Q(x) = Q(x) * P(x)$$

Nota: se debe tener en cuenta la ley de signos. Ver figura 2:

Figura 2. Ley de signos.

$$\begin{aligned} (+) (+) &= + \\ (+) (-) &= - \\ (-) (+) &= - \\ (-) (-) &= + \end{aligned}$$

En las figuras 3 y 4, se indica ejemplos de la multiplicación de polinomios.

Figura 3. Multiplicación polinomio por monomio.

$$\begin{array}{r} (x^5 + 2x^3 + 6x^2 + 4x + 9) \quad * \\ \quad \quad \quad \quad \quad \quad \quad (x + 5) \\ \hline x^6 + \quad + 2x^4 + 6x^3 + 4x^2 + 9x \\ \quad + 5x^5 + \quad + 10x^3 + 30x^2 + 20x + 45 \\ \hline x^6 + 5x^5 + 2x^4 + 16x^3 + 34x^2 + 29x + 45 \end{array}$$

Figura 4. Multiplicación polinomio por polinomio.

$$\begin{array}{r}
 (x^5 + 2x^3 + 6x^2 + 4x + 9) \quad * \\
 \times (6x^4 + 7x^3 + 4x^2 + 12x + 10) \\
 \hline
 6x^9 + \quad + 12x^7 + 36x^6 + 24x^5 + 54x^4 \\
 \quad + 7x^8 + \quad + 14x^6 + 42x^5 + 28x^4 + 63x^3 \\
 \quad \quad + 4x^7 + \quad + 8x^5 + 24x^4 + 16x^3 + 36x^2 \\
 \quad \quad \quad + 12x^5 + 36x^4 + 72x^3 + 48x^2 + 108x \\
 \quad \quad \quad \quad + 10x^5 + \quad + 20x^3 + 60x^2 + 40x + 90 \\
 \hline
 6x^9 + 7x^8 + 16x^7 + 50x^6 + 96x^5 + 142x^4 + 171x^3 + 144x^2 + 148x + 90
 \end{array}$$

División con polinomios: para dividir dos polinomios se los ordena en forma ascendente o descendente tanto dividendo como divisor; en la parte izquierda se ubica el dividendo y si el polinomio no es completo se deja huecos en los lugares que correspondan, en la parte derecha se ubica el divisor; ahora si se debe tratar de cancelar el primer término que quede del dividendo realizando o aplicando un análisis aritmético.

Ejemplo: sean $P(x) = x^5 + 2x^3 - x - 8$ y $Q(x) = x^2 - 2x + 1$ tenemos que $P(x)/Q(x)$ se resuelve de la siguiente manera:

- Una vez se tiene organizado los polinomios a dividir como corresponde se divide el primer monomio del dividendo entre el primer monomio del divisor.

Tratando de cancelar el primer término se busca un monomio que al multiplicar con el divisor se cancele el primer término del dividendo; para esto se multiplica cada término del polinomio divisor por x^3 el resultado se ubica a la parte izquierda debajo del dividendo de forma ordenada para luego proceder a la resta entre el dividendo, como se muestra en la figura 5.

Figura 5. División polinomios iteración 1

$$\begin{array}{r}
 x^5 \quad + 2x^3 \quad - x - 8 \\
 \underline{-x^5 + 2x^4 - x^3} \\
 2x^4 + x^3 \quad - x - 8
 \end{array}
 \qquad
 \begin{array}{r}
 | \quad x^2 - 2x + 1 \\
 \underline{x^3}
 \end{array}$$

- El siguiente paso es repetir el paso anterior con el resto del dividendo hasta que este sea cero o el dividendo sea de grado menor que el divisor, como se ilustra en la figura 6.

Figura 6. División polinomios iteración 2

$$\begin{array}{r}
 x^5 \qquad + 2x^3 \qquad - x - 8 \\
 \underline{-x^5 + 2x^4 - x^3} \\
 2x^4 + x^3 \qquad - x - 8 \\
 \underline{-2x^4 + 4x^3 - 2x^2} \\
 5x^3 - 2x^2 - x - 8
 \end{array}
 \qquad
 \begin{array}{r}
 \overline{) x^2 - 2x + 1} \\
 \underline{x^3 + 2x^2}
 \end{array}$$

- En la figura 6, el dividendo aún es de grado mayor que el divisor por lo cual se procede a otra iteración, como lo ilustra la figura 7.

Figura 7. División polinomios iteración 3

$$\begin{array}{r}
 x^5 \qquad + 2x^3 \qquad - x - 8 \\
 \underline{-x^5 + 2x^4 - x^3} \\
 2x^4 + x^3 \qquad - x - 8 \\
 \underline{-2x^4 + 4x^3 - 2x^2} \\
 5x^3 - 2x^2 - x - 8 \\
 \underline{-5x^3 + 10x^2 - 5x} \\
 8x^2 - 6x - 8
 \end{array}
 \qquad
 \begin{array}{r}
 \overline{) x^2 - 2x + 1} \\
 \underline{x^3 + 2x^2 + 5x}
 \end{array}$$

- Finalmente, en la figura 8, se mira que el dividendo es de menor grado que el divisor de esta manera termina la división entre los dos polinomios y se puede observar que la división no fue exacta y se tiene como resultado $P(x)/Q(x)$ un cociente $x^3 + 2x^2 + 5x + 8$ y un residuo $10x - 16$

Figura 8. División polinomios iteración 4

$$\begin{array}{r}
 x^5 \qquad + 2x^3 \qquad - x - 8 \\
 \underline{-x^5 + 2x^4 - x^3} \\
 2x^4 + x^3 \\
 \underline{-2x^4 + 4x^3 - 2x^2} \\
 5x^3 - 2x^2 - x \\
 \underline{-5x^3 + 10x^2 - 5x} \\
 8x^2 - 6x - 8 \\
 \underline{-8x^2 + 16x - 8} \\
 10x - 16
 \end{array}
 \qquad
 \begin{array}{r}
 | x^2 - 2x + 1 \\
 \hline
 x^3 + 2x^2 + 5x + 8
 \end{array}$$

Factorización de polinomios: descomponer factorialmente una expresión algebraica es hallar dos o más expresiones algebraicas cuyo producto sea la expresión inicial. En la figura 9, se ilustra ejemplos de factorización de polinomios.

Figura 9. Factorización simbólica polinomios

$$\begin{aligned}
 (x+3)(x+5) &= x^2 + 8x + 15 \\
 (x-3)(x+5) &= x^2 + 2x - 15 \\
 (x+3)(x-5) &= x^2 - 2x - 15 \\
 (x-3)(x-5) &= x^2 - 8x + 15
 \end{aligned}$$

1.3 CAJA DE POLINOMIOS

1.3.1 Representación geométrica de polinomios

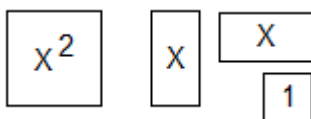
1.3.1.1 Generalidades: con el estudio de la representación geométrica de polinomios se quiere lograr que los estudiantes tengan un alto nivel en el proceso de desarrollo de operaciones polinómicas ya que representa una meta a conseguir y constituye una señal de madures en las matemáticas como disciplina. De

acuerdo a las diferentes investigaciones sobre la representación geométrica de polinomios se proponen una diversidad de definiciones entre las cuales están. *Rectángulos de Lilian I, Cajas de Adrián y los Rectángulos de Libardo.*

1.3.1.2 Rectángulos de Lilian I. En el postgrado de enseñanza de la Matemática en el Año 1997 en la Universidad de Nariño, Lilian Narvaez explicó sobre el uso de rectángulos que permiten multiplicar dos polinomios de la forma $ax + b$, donde a y b están en $\mathbb{Z}^+ \cup \{0\}$ y factorizar polinomios de la forma $ax^2 + bx + c$ donde a, b, c están en $\mathbb{Z}^+ \cup \{0\}$, haciendo una interpretación de la geometría de *Tabit ben Qurra el Harrani*, algebrista árabe que murió en 901.

La interpretación geométrica de *Tabit ben Qurra* permite adoptar X^2 como un cuadrado de lado x , la variable x está representada por un rectángulo de lados x y 1 y el 1 es un cuadrado de lado 1 tal y como se muestra en *figura 10*.

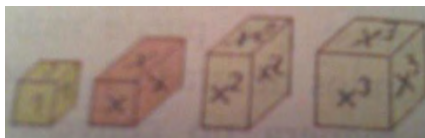
Figura 10. Rectángulos de Lilian



1.3.1.3 Cajas de Adrián: hasta el surgimiento de los *Rectángulos de Lilian I*, se podía multiplicar dos polinomios de coeficientes enteros con grado 1 y factorizar polinomios de la forma $ax^2 + bx + c$ con a, b y c coeficientes enteros, sin embargo, en el año 1998, *Libardo Jacome* dictó un curso de Matemáticas en la Licenciatura de Educación Básica de la Universidad de Nariño en el municipio de Gualmatan y en dicho curso se estudió los *Rectángulos de Lilian* y como trabajo creativo se planteó multiplicar tres polinomios de grado 1 con coeficientes enteros no negativos; el estudiante *Adrián Moran* resuelve el problema proponiendo cuatro paralelepípedos; rectángulos básicos que en el libro *CAJA DE POLINOMIOS* se le denominó como *Cajas de Adrian* con la intención de dar crédito al estudiante que propuso la idea.

Las *Cajas de Adrian*, admiten manejar polinomios de grado 3 permitiendo incluir una tercera dimensión a las fichas; estas cajas tienen como volumen X^3, X^2, X y 1 cuyas aristas son respectivamente $x, x, x; x, x, 1; x, 1, 1; 1, 1, 1$, su esquema gráfico se dispone a continuación en la figura 11 y consagra la idea de *Tabit ben Qurra* de homogenizar los términos de un polinomio de grado tres a través de paralelepípedos donde lo importante es su volumen:

Figura 11. Cajas de Adrián



1.3.1.4 Rectángulos de Libardo: con las “Cajas de Adrián” sale la pregunta de cómo multiplicar un polinomio de grado 1 cuatro veces por el mismo, lo que requiere de una nueva ficha para la representación geométrica X^4 ; siguiendo la idea de las “Cajas de Adrián” se requiere de una cuarta dimensión lo cual ya es imposible porque hasta ahora el mundo se ve y vive en una tercera dimensión.

El 8 de febrero de 1999, Libardo Jácome, catedrático de la Universidad de Nariño concibe la posibilidad de multiplicar n polinomios de grado uno con coeficientes enteros no negativos y la factorización de polinomios de la forma:

$$p(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n, a_i \in \mathbb{Z}^+ \cup \{0\}$$

Esto se logró con el artificio denominado cambio de variable y haciendo uso de únicamente de rectángulos o tarjetas, remplazando un paralelepípedo por una tarjeta, un cubo por otro, un hipercubo por otro.

1.3.2 Caso de los coeficientes negativos. En un estudio con un grado de dificultad más elevado los docentes Libardo Jacome y Oscar Fernando Soto en el libro Caja de Polinomios plantean la forma de multiplicar dos o más polinomios con coeficientes enteros sin restricción alguna, es decir, se pueden escoger para algunos de los términos coeficientes negativos. La forma creativa de hacerlo, es una aplicación evidente del plano cartesiano, Renato Descartes.

Se debe disponer de un plano cartesiano como el que se propone en la figura 12, en donde los cuadrantes primero y tercero representen las fichas con valores positivos y los cuadrantes segundo y cuarto por su parte integran a las fichas con valores negativos, con esto queda solucionado el problema de los coeficientes negativos y se abre paso a una nueva metodología de resolución de operaciones con polinomios.

Figura 12. Tablero caja de polinomios

<i>Negativo</i>	<i>Positivo</i>
<i>Positivo</i>	<i>Negativo</i>

1.3.3 Suma. Disponiendo del tablero e identificando el signo correspondiente a cada cuadrante del plano cartesiano se procede a ubicar los dos polinomios a sumar teniendo en cuenta los pasos siguientes:

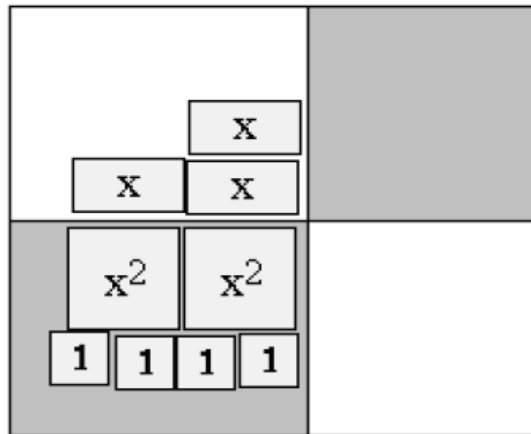
- I. Ubicar $P(x)$ en el segundo y tercer cuadrante.
- II. Ubicar $Q(x)$ en el primero y cuarto cuadrante.
- III. Retirar los pares de fichas que equivalgan algebraicamente a cero.
- IV. Leer el polinomio resultante.
- V. Escribir en el cuaderno de trabajo la expresión obtenida.

Ejemplo: Efectuar la adición de los polinomios.

$$P(x) = 2X^2 - 3X + 4$$
$$Q(x) = -X^2 + X - 2$$

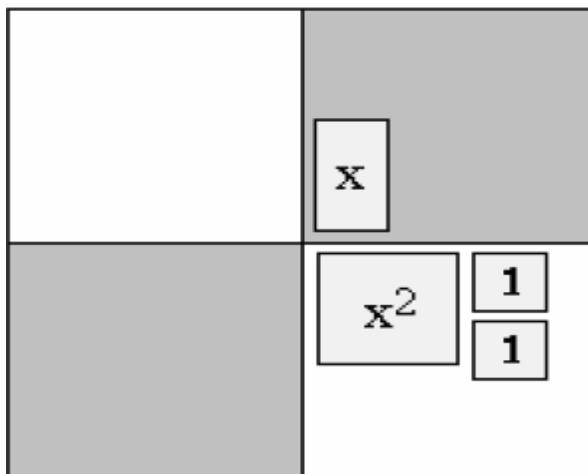
- I. Escribir el polinomio $P(x) = 2X^2 - 3X + 4$ en los cuadrantes segundo y tercero. Este paso se ilustra en la figura 13.

Figura 13. Suma de polinomios escribiendo P(x) caja de polinomios



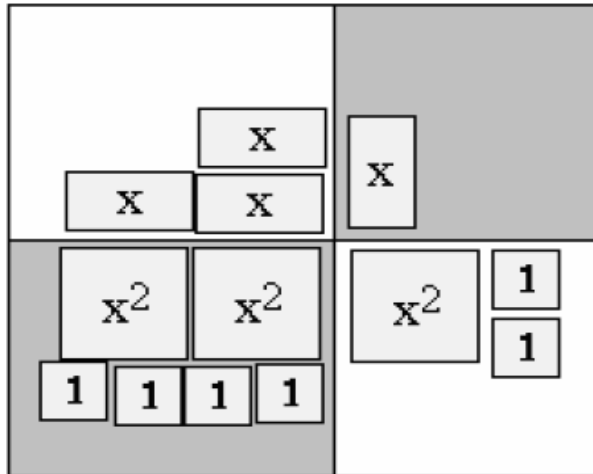
- II. Escribir el polinomio $Q(x) = -x^2 + x - 2$ en los cuadrantes primero y cuarto, como se ilustra en la figura 14.

Figura 14. Suma de polinomios escribiendo Q(x) caja de polinomios



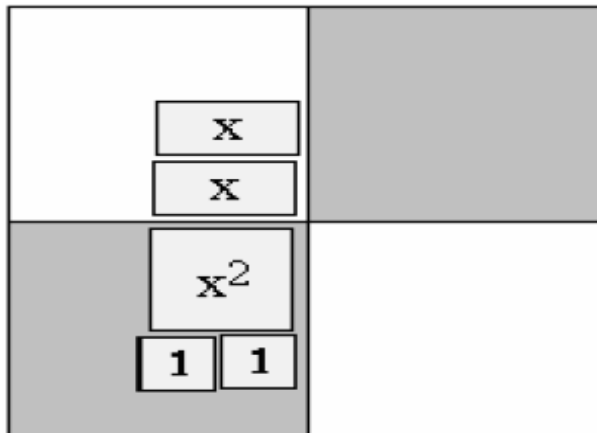
Al ejecutar los dos pasos anteriores el tablero toma la siguiente apariencia. Ver figura 15.

Figura 15. Suma de polinomios $P(x)+Q(x)$ caja de polinomios



- III. Retirar del plano, si lo hubiere, los pares de fichas que equivalgan algebraicamente a cero, como lo ilustra la figura 16.

Figura 16. Resultado Suma caja de polinomios



- IV. Leer el polinomio que queda escrito en todo el tablero, tal y como se ve en el plano anterior; es decir $X^2 - 2X + 2$. De esta forma, se obtiene que $(2X^2 - 3X + 4) + (-X^2 + X - 2) = X^2 - 2X + 2$

Debido a que el álgebra es un juego operatorio simbólico es necesario indicar mediante símbolos lo que de manera concreta se ha realizado en el tablero.

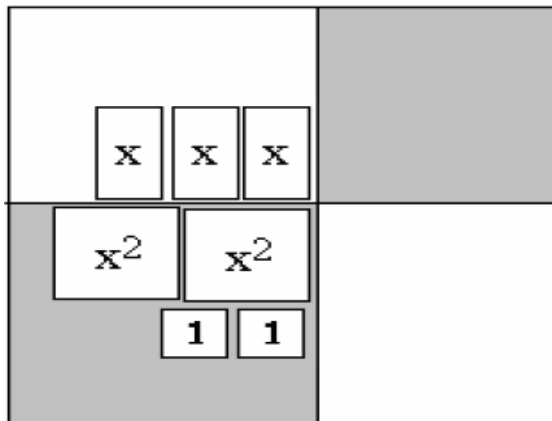
1.3.4 Resta. Para calcular la diferencia entre $P(x)$ y $Q(x)$, $P(x) - Q(x)$ se procede de manera similar al cálculo de sumas siguiendo los siguientes pasos:

- I. Escribir el minuendo $P(x)$ en los cuadrantes Segundo y Tercero.
- II. Escribir el sustraendo $Q(x)$ en los cuadrantes Primero y Cuarto.
- III. Dado que restar es sinónimo de quitar, las fichas ubicadas en el primer cuadrante correspondientes a $Q(x)$ deben cambiarse de signo, lo que equivale a trasladarlas al segundo cuadrante, de igual forma se procede con las fichas ubicadas en el cuarto cuadrante que deben trasladarse al tercer cuadrante.
- IV. Se retiran del tablero, los ceros que hayan configurado.
- V. La diferencia está constituida por las fichas que finalmente quedan en el tablero.

Ejemplo: Calcular la diferencia entre $P(x) = 2x^2 - 3x + 2$ y $Q(x) = x^2 - 2x + 1$

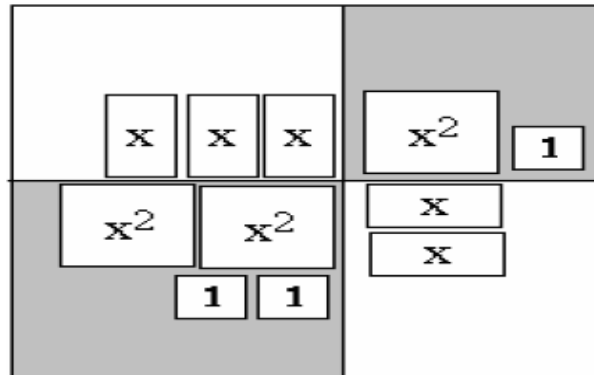
- I. Escribir el polinomio $P(x) = 2x^2 - 3x + 2$ (minuendo) en los cuadrantes segundo y tercero. Ver figura 17.

Figura 17. Resta Escribiendo $P(x)$ caja de polinomios



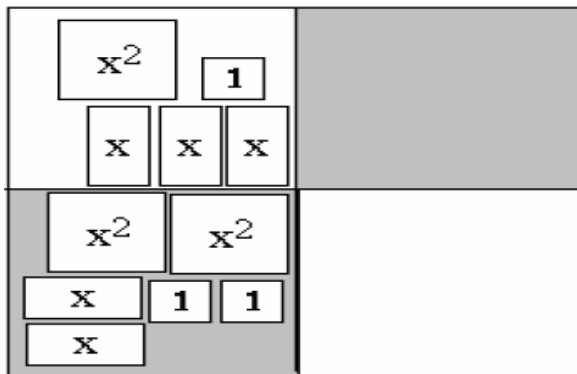
- II. Escribir el polinomio $Q(x) = x^2 - 2x + 1$ (sustraendo) en los cuadrantes primero y cuarto. Ver figura 18.

Figura 18. Resta Escribiendo Q(X) caja de polinomios



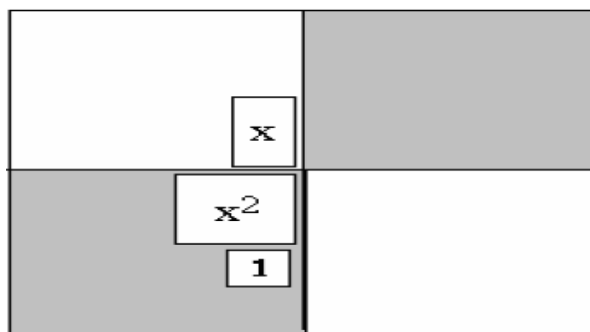
- III. Trasladar las fichas del sustraendo: las del primer cuadrante al segundo y las del cuarto cuadrante al tercero, como lo indica la figura 19.

Figura 19. Resta trasladar fichas del sustraendo caja de polinomios



- IV. Retirar del plano las fichas que representan ceros. Ver el resultado de este paso en la figura 20.

Figura 20. Resultado resta caja de polinomios

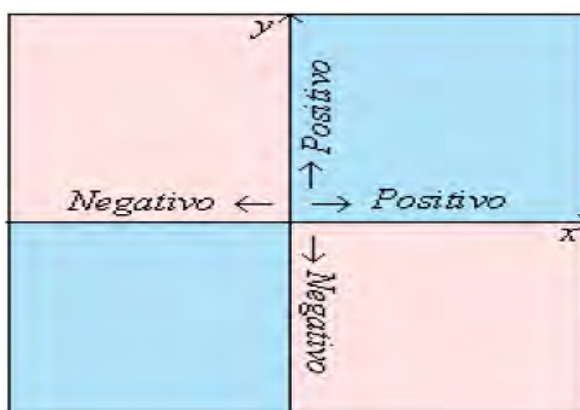


V. Leer el polinomio resultante, en este caso: $P(x) - Q(x) = x^2 - x + 1$

De lo anterior se puede deducir que la diferencia entre $P(x) = 2x^2 - 3x + 2$ y $Q(x) = x^2 - 2x + 1$ es $P(x) - Q(x) = x^2 - x + 1$.

1.3.5 Optimización del uso del plano cartesiano. Como un avance más para llevar acabo la multiplicación de polinomios utilizando el álgebra los docentes Oscar Fernando Soto y Libardo Jacome hicieron un estudio de como optimizar el uso del plano cartesiano y dedujeron que el plano cartesiano está configurado a partir de dos ejes perpendiculares que se cortan en el origen de coordenadas, cada uno de los ejes contiene dos direcciones opuestas: positiva y negativa como se muestra en la siguiente figura 21.

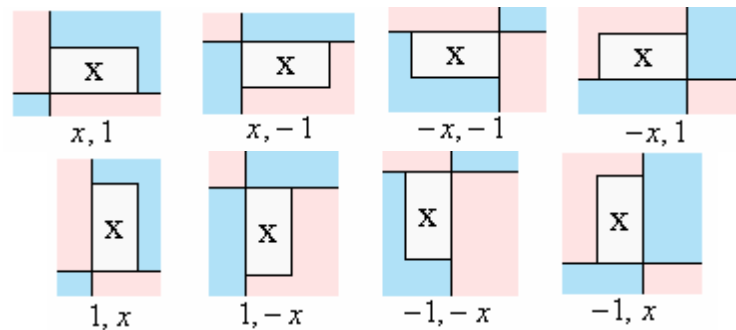
Figura 21. Tablero para la multiplicación caja de polinomios



Para cada rectángulo que se construye sobre este plano, las dimensiones se leerán como base y altura en su orden, siendo que la base corresponde al lado paralelo al eje x y la altura, el lado paralelo al eje y, y en concordancia con la

orientación de los ejes. De esta manera, la ubicación de una ficha o de un rectángulo a partir del origen y haciendo uso de los ejes coordenados establece unas dimensiones distintas como se muestra con las siguientes disposiciones de la ficha x. Ver figura 22.

Figura 22. Ubicación de las fichas Multiplicación caja de polinomios



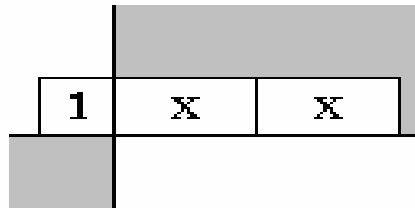
1.3.5.1 Multiplicación. La aplicación de la multiplicación requiere de mayor habilidad y poder de abstracción y se sujeta a recordar que del origen de coordenadas a la izquierda, el eje x toma valores negativos, Igual sucede con el eje y del origen hacia abajo.

El plano cartesiano con sus ejes de coordenadas se constituye en una guía esencial para el cálculo de productos. El producto $P(x).Q(x)$ corresponden al valor algebraico relativo de las fichas que configuran un rectángulo de base $P(x)$ y de altura $Q(x)$ o viceversa. La lectura del producto se realiza, después de retirar los pares de fichas que algebraicamente equivalen a cero y que se ubican, para recordarlo, en cuadrantes de colores distintos, si es que los hubiere.

El siguiente ejemplo ilustra el procedimiento para calcular productos de la forma $(ax + d).(cx + d)$ puesto que estudia el cálculo de $(2x - 1).(x + 2)$ indicando dos caminos.

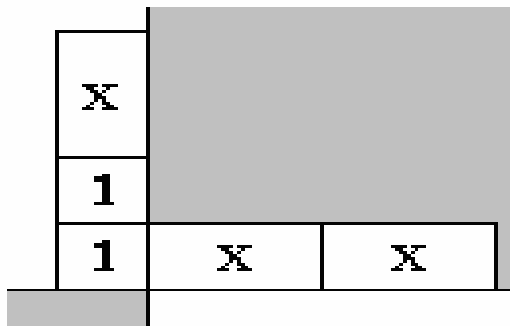
- **Primer Camino:** se toma como base el polinomio $P(x) = (2x - 1)$, ubicando dicho polinomio a partir del origen y haciendo uso adecuado de los ejes coordenados, como se muestra en la figura23.

Figura 23. Ubicando el primer factor $P(x)$ caja de polinomios



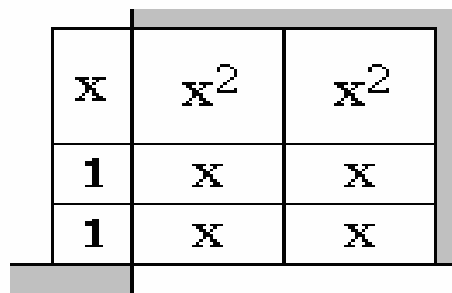
Una vez hecho eso y utilizando el criterio de que fichas adyacentes deben tener la misma dimensión en su frontera común, se configura la altura del rectángulo cuya dimensión está dada por el factor $Q(x) = (x + 2)$; este paso se señala en la figura 24.

Figura 24. Ubicando segundo factor $Q(x)$ caja de polinomios



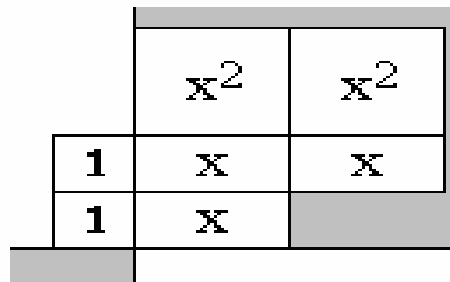
Ahora, se arma completamente el rectángulo utilizando tantas fichas como sea necesario. El rectángulo completo se dispone en la figura 25.

Figura 25. Multiplicación completando el rectángulo caja de polinomios



Finalmente se procede a retirar fichas que algebraicamente equivalen a cero, en este caso, un par de fichas rotuladas con x y se procede a leer la respuesta teniendo en cuenta la ubicación de las fichas en sus respectivos cuadrantes; el tablero resultante se ilustra en la figura 26.

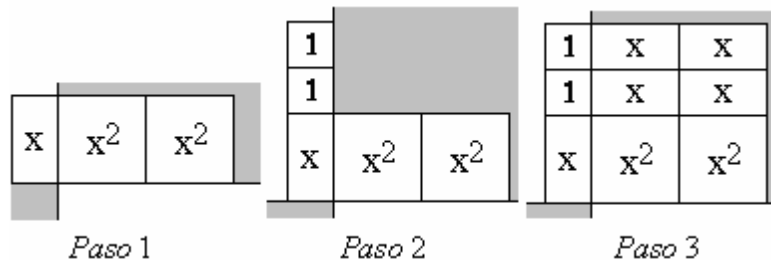
Figura 26. Multiplicación quitando ceros caja de polinomios



Y en consecuencia se tiene que: $(2x - 1) \cdot (x + 2) = 2x^2 + 3x - 2$

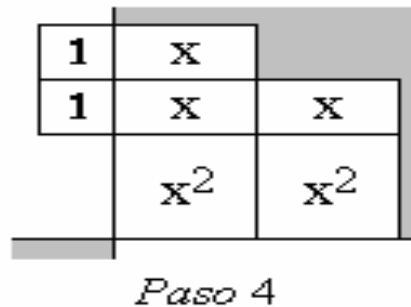
- **Segundo camino:** se establece como base el polinomio $P(x) = (2x - 1)$ pero en este caso, como se muestra en la figura 27, se tiene el sumando x de la altura $Q(x) = (x + 2)$, a partir de allí, se completa el rectángulo teniendo en cuenta el criterio de que fichas adyacentes deben coincidir en la dimensión de su frontera común:

Figura 27. Multiplicación ubicando factores $P(x)$, $Q(x)$ caja de polinomios



Después de completado el rectángulo se procede a retirar las parejas de fichas que algebraicamente equivalen a cero, como lo indica la figura 28, de esta forma se permite leer el producto de manera simple.

Figura 28. Multiplicación quitando ceros caja de polinomios – 2



Así: $P(x).Q(x) = (2x - 1).(x + 2) = 2x^2 + 3x - 2$

Este camino alternativo para calcular el producto tiene en cuenta desde un comienzo la composición del otro factor y es que se debe recordar que en las operaciones binarias cada una de las dos componentes tiene igual importancia.

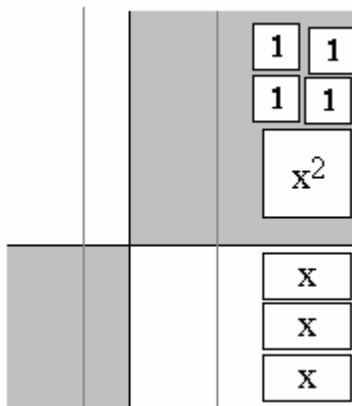
1.3.6 División. La división es sinónimo de repetir o de distribuir y efectivamente es lo que se hace cuando se trata de calcular el cociente $P(x) \div Q(x)$; en este caso se debe construir con el dividendo $P(x)$ un rectángulo de base $Q(x)$ teniendo como recurso la agregación de pares de fichas que equivalen a cero (Comodín). El cociente de la división es la altura de dicho rectángulo, mientras que el residuo, si existe, está constituido por fichas 1 ya que se efectuaran divisiones del tipo $ax^2 + bx + c \div (dx + e)$.

El siguiente ejemplo ayuda a comprender mejor la división de polinomios.

Calcula el cociente $x^2 - 3x + 4 \div (x - 1)$.

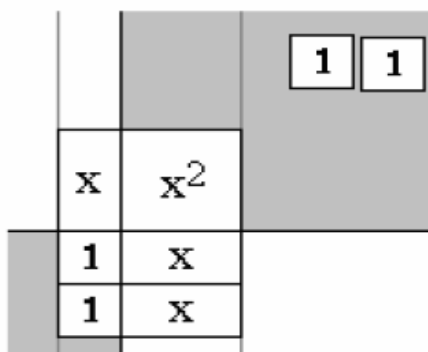
- **Paso 1:** Escribir el polinomio dividendo $x^2 - 3x + 4$. En el plano Cartesiano y establecer sobre el plano unas guías imaginarias de anchura equivalente al divisor $x - 1$ como se indica en la figura 29.

Figura 29. División escribiendo los polinomios



- Paso 2:** Ubicar las fichas que representan al polinomio sobre la banda imaginaria, teniendo en cuenta que se debe utilizar al menor número posible de pares de fichas que representen cero y respetando el color del cuadrante que les corresponde, es decir, el polinomio que se debe leer en el tablero, una vez efectuado el paso dos se obliga a ser el dividendo $P(x)$, como se muestra en la figura 30.

Figura 30. División ubicar fichas banda imaginaria

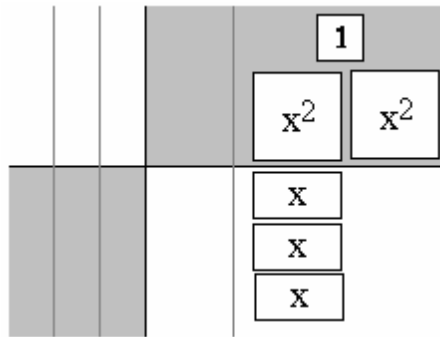


En este ejemplo no hubo necesidad de agregar ceros, y es claro que el cociente es $x - 2$ y el residuo es 2.

Calcula el cociente $2x^2 - 3x + 1 \div (x - 2)$.

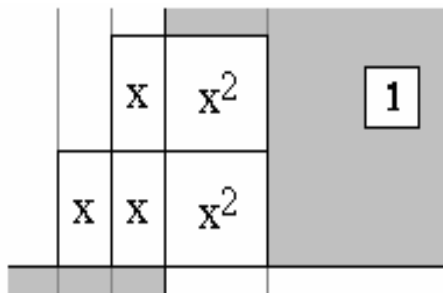
- **Paso 1:** se escribe el polinomio dividido $2x^2 - 3x + 1$ en el plano cartesiano, polinomio que se organizara en el siguiente paso alrededor del origen de coordenadas y se establece la franja de anchura $x - 2$ y que es el divisor, como lo indica la figura 31.

Figura 31. División escribir polinomio ejemplo 2



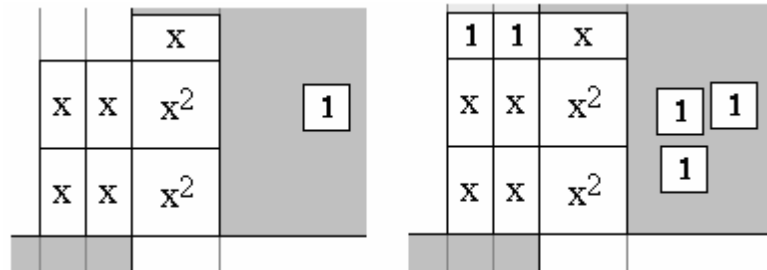
- **Paso 2:** se organizan las fichas alrededor del origen preservando el signo que les corresponde en concordancia con el color del cuadrante y buscando que para completar el rectángulo falten, la menor cantidad de pares de fichas equivalentes algebraicamente a cero, como lo indica la figura 32.

Figura 32. División ubicar fichas banda imaginaria ejemplo 2



- **Paso 3:** Se completa el rectángulo con los dos siguientes movimientos, sobre los cuales se hace evidente que toca agregar tres parejas de ceros. Ver figura 33.

Figura 33. División completar cuadrado y quitar ceros ejemplo 2



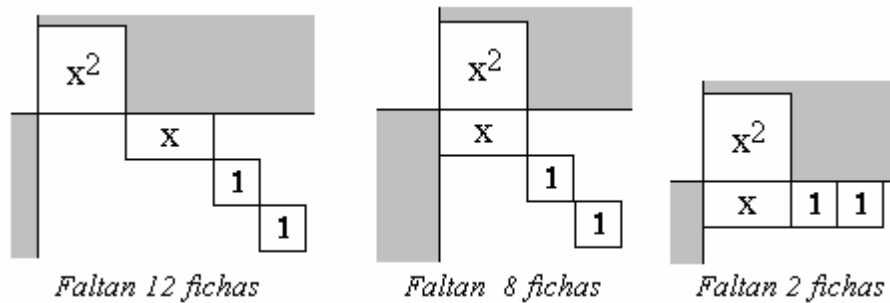
De este modo, el cociente es $2x + 1$ y el residuo es 3 y en concordancia con el algoritmo de la división se escribe $2x^2 - 3x + 1 = (x - 2)(2x - 1) + 1$.

1.3.7 Factorización. En una de las recientes guías expuestas por el grupo de Investigación GESCAS, en cabeza del profesor Fernando Soto, se explica con gran claridad el proceso de factorización de Polinomios de Grado dos.

Factorizar un Polinomio utilizando la Caja de Polinomios equivale a disponer una representación rectangular del mismo, siempre que esto sea posible. Para realizar esta tarea a veces es necesario la agregación de ceros; es decir, de parejas de fichas del mismo rótulo que ubicadas en cuadrantes del mismo color equivalen algebraicamente a cero. Para ese efecto, se requiere disponer el menor número de fichas que representan al polinomio en cuestión en un encuadre minimal viable.

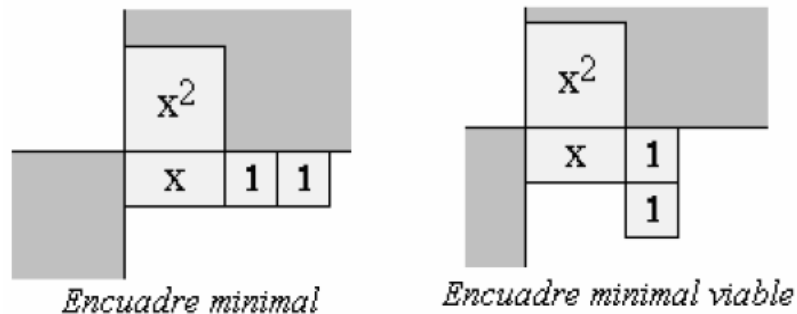
Un encuadre mínimo, es aquella disposición de un polinomio $P(x)$ de forma que al completar un rectángulo requiera del menor número de fichas. En la gráfica 34, se representan tres disposiciones del polinomio $P(x)=x^2-x-2$, en el tablero; de los cuales el tercero corresponde a un cuadrante mínimo:

Figura 34. Encuadre mínimo



Un encuadre mínima viable es aquel encuadre mínimo que requiere de un número par de fichas que equivalen algebraicamente a cero y que completan el rectángulo que representa al polinomio $P(x)$. En la gráfica 35, se representa dos encuadres mínimos de $P(x) = x^2 - x - 2$, de los cuales el segundo es viable, pues requiere de dos fichas x para completar el rectángulo, pero algebraicamente, la agregación de estas fichas equivalen a sumar cero.

Figura 35. Encuadre mínimo viable



Al completar el rectángulo que representa al polinomio $P(x)$ a partir de un encuadre mínimo viable, se ha factorizado, su factorización es el producto de las dimensiones de dos lados consecutivos del rectángulo. Por ejemplo a partir de la disposición mínima viable que representa $P(x) = x^2 - x - 2$, se obtiene el resultado final de la operación, como lo indica la figura 36.

Figura 36. Encuadre mínimo terminación ejemplo

	x^2	x
x	x	1
x	x	1

Y siendo las dimensiones de dos lados consecutivos de este rectángulo $x + 1$ y $x - 2$ se consigue que $P(x) = x^2 - x - 2 = (x + 1)(x - 2)$.

1.4 HERRAMIENTAS MATEMÁTICAS PARA OPERAR POLINOMIOS

1.4.1 Aplicaciones web

- **Experymente:** es un blog para compartir apuntes animados, exámenes, documentales, noticias del mundo de la ciencia y enlaces a sitios de interés de la física, y las matemáticas⁵.
- **Schollaris la web que resuelve problemas:** web en la que ayuda a resolver algunos problemas de aritmética, algebra entre estos calculadora (suma, resta, multiplicación, raíces de polinomios) de polinomios⁶.
- **www.ingenieriaycalculos.com:** web en la que brindan una calculadora en línea a los usuarios para operar (suma, resta, multiplicación, raíces de polinomios) de polinomios⁷.

⁵ EXPERYMENTE. < <http://experymente.blogspot.com.co/2013/11/calculadora-de-division-de-polinomios.html>.> [Consultado el 15 de julio 2015].

⁶ SCHOLLARIS. < <http://schollaris.com.mx/sitemap.php>> [Consultado el 15 de julio 2015].

⁷ POLINOMIOS. < <http://www.ingenieriaycalculos.com/matematicas/algebra/polinomios>> [Consultado el 15 de julio 2015].

1.4.2 Aplicaciones de escritorio

- **MATLAB** (*MATrix LABoratory*, "laboratorio de matrices"): es una herramienta de software matemático que permite el manejo de funciones matemáticas para álgebra lineal, estadística, análisis de Fourier, integración numérica, y la resolución de ecuaciones diferenciales ordinarias entre muchas otras; este software ofrece al usuario un entorno de desarrollo integrado (IDE) con un lenguaje de alto nivel para el cálculo numérico visualización y desarrollo de aplicaciones⁸.
- **Derive**: fue un sistema de gran alcance para hacer matemáticas simbólicas y numéricas en su PC. Este software procesa variables algebraicas, expresiones, ecuaciones, funciones, vectores, matrices y expresiones booleanas.

1.4.3 Aplicaciones móviles

- **Factorización de polinomios**: es una calculadora para móviles de uso libre capaz de factorizar un polinomio le ayuda al usuario a resolver ecuaciones polinómicas, elaborar gráficos entre otras funciones⁹.
- **FactorizationA**: es una calculadora que permite factorizar polinomios¹⁰.
- **Polynomial Analyser**: aplicación que permite al usuario operar polinomios (suma, resta, multiplicación, división) y factorizar polinomios¹¹.

1.5 FRAMEWORK: Es una estructura *software* compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación el cual facilita un esquema para el desarrollo e implementación de aplicaciones.

Un *framework* adiciona funcionalidades extendidas a un lenguaje de programación, automatiza muchos patrones en la programación para un

⁸ MATLAB. < <http://www.mathworks.com/products/matlab/> > [Consultado el 15 de julio 2015].

⁹ FACTORIZACIÓN, Polinomios. < https://play.google.com/store/apps/details?id=an.PolFac&hl=es_419 > [Consultado el 15 de julio 2015].

¹⁰ FACTORIZACIÓN.

<<https://play.google.com/store/apps/details?id=haesaka.jp.ne.ocn.www2.factorizationa>> [Consultado el 15 de julio 2015].

¹¹ POLYNOMIAL, Analyser. < <https://play.google.com/store/apps/details?id=com.dspsoft.polyanalyser> > [Consultado el 15 de julio 2015].

determinado objetivo, los *frameworks* proporciona una estructura al código haciendo que sea más entendible y sostenible.

Existen varios tipos de *frameworks* como por ejemplo orientados a la interfaz de usuario, como *Java Server Faces*, orientados a la parte de control de eventos entre muchos otros.

1.6 APACHE CORDOVA: es un marco de trabajo para el desarrollo de aplicaciones *móviles* este *software* es libre y de código abierto bajo la licencia de *apache*. *Apache Cordova*, permite el desarrollo multiplataforma haciendo uso de tecnologías como *HTML5*, *CSS3* y *JavaScript* y evita la codificación nativa para cada una de las plataformas existentes.

1.7 JQUERY: Biblioteca de *JavaScript* creada inicialmente por *John Resig* de *software libre* y código abierto que permite el recorrido y manipulación de documentos, manejo de eventos, animación, *Ajax* de forma más simple con una *API* fácil de usar que funciona a través de una multitud de navegadores y combina gran versatilidad y extensibilidad facilitando la interacción de los documentos *HTML* y manipulación del *DOM*. Esta es la biblioteca de *JavaScript* más usada.

1.8 ¿QUÉ ES UNA APLICACIÓN?

Es un programa informático creado para llevar a cabo o facilitar una tarea en un dispositivo informático. Cabe destacar que aunque todas las aplicaciones son programas, no todos los programas son aplicaciones. Existe multitud de *software* en el mercado, pero solo se denomina así aquel que ha sido creado con un fin determinado, para realizar tareas concretas. No se considera una aplicación, por ejemplo, un sistema operativo, ni una suite, pues su propósito es general.

Las aplicaciones nacen de alguna necesidad concreta de los usuarios, y se usan para facilitar o permitir la ejecución de ciertas tareas en las que un analista o un programador han detectado una cierta necesidad. Pero las aplicaciones también pueden responder a necesidades lúdicas, además de laborales (todos los juegos, por ejemplo, son considerados aplicaciones).

Las aplicaciones pueden ser desde pequeñísimos programas de apenas unas líneas de código, hasta grandes obras de ingeniería, con miles de horas de trabajo detrás. El tamaño, sin embargo, no define el éxito de una aplicación, sino justamente que cumpla las necesidades del usuario. La historia de las aplicaciones está llena de miles de grandes aplicaciones que han sido sustituidas

por otras a priori mucho menos elaboradas, simplemente porque eran más rápidas, intuitivas, estables o cumplían mejor su función.

En general, una aplicación se diseña para ser usada con uno o varios sistemas operativos, siendo hasta cierto punto dependiente de estos. Pero también existen aplicaciones diseñadas en exclusivas para el uso con un programa o tipo de programas (Ejemplos: Barra de los navegadores Web).

1.8.1 Tipos de Aplicaciones. Naturalmente, el campo de las aplicaciones es tan extenso, y las funciones tan dispersas, que se han creado numerosas clasificaciones, según varios criterios. Así, por ejemplo, dependiendo del uso que se le vaya a dar, podemos tener aplicaciones profesionales o personales. También se puede hablar de aplicaciones para Windows, Linux, Android, Apple, PC, Móvil, según el sistema operativo o la plataforma que vayamos a usar. Según su finalidad, pueden ser de entretenimiento, de negocios, empresariales, de sonido, gráficas, de edición web, didácticas etc., e incluso según la rama de las ciencias en el que se usen se puede decir que hay aplicaciones de ingeniería, de telecomunicaciones, matemáticas entre otras, a continuación se describe algunas

- **Aplicaciones Web:** en Ingeniería de *software* se denomina aplicación *web* a aquellas aplicaciones que los usuarios pueden utilizar accediendo a un servidor *Web* a través de *internet* o de una *intranet* mediante un navegador. En otras palabras, es una aplicación *software* que se codifica en un lenguaje soportado por los navegadores *web*.

Las aplicaciones *web* son populares debido a lo práctico del navegador *web* como cliente ligero, a la independencia del sistema operativo, así como a la facilidad para actualizar y mantener aplicaciones *web* sin distribuir e instalar *software* a miles de usuarios potenciales.

En la tabla 4, se indican algunas ventajas y desventajas de las aplicaciones *web*.

Tabla 4. Aplicaciones Web Ventajas y Desventajas

Ventajas	Inconvenientes
El mismo código base reutilizable en múltiples plataformas.	Requiere de conexión a Internet.
Proceso de desarrollo más sencillo y económico.	Acceso muy limitado a los elementos y características del hardware del dispositivo.
No necesitan ninguna aprobación externa para publicarse (a diferencia de las nativas para estar visibles en la app store).	La experiencia del usuario (navegación, interacción...) y el tiempo de respuesta es menor que en una app nativa.
El usuario siempre dispone de la última versión / Pueden utilizarse sitios "responsive" ya diseñados.	Requiere de mayor esfuerzo en promoción y visibilidad.

- **Aplicación móvil:** *software* desarrollado para correr bajo *Smartphone*. Está diseñada para educar entretener o ayudar en la vida diaria a sus usuarios/consumidores. Funcionalidades que antes solo estaban disponibles en PC ahora pueden ser ejecutadas en los dispositivos móviles, con un buen nivel de prestaciones.
- **Aplicación nativa:** una aplicación nativa es la que se desarrolla de forma específica para un determinado sistema operativo, llamado *Software Development Kit* o *SDK*. Cada una de las plataformas, *Android*, *IOS* o *Windows Phone*, tiene un sistema diferente, por si se requiere que una app esté disponible en todas las plataformas se deberán de crear varias *apps* con el lenguaje del sistema operativo seleccionado. Por ejemplo:
 - *App* para **IOS** se desarrollan con lenguaje *Objective-C*
 - *App* para **Android** se desarrollan con lenguaje *Java*
 - *App* para **Windows Phone** se desarrollan en *.Net*

Cuando se habla de desarrollo *móvil* casi siempre se hace referencia a aplicaciones nativas. La principal ventaja con respecto a los otros tipos (*Apps Web* y *Apps Híbridas*), es la posibilidad de acceder a todas las características del *hardware* del *móvil*: Cámara, *GPS*, agenda, dispositivos de almacenamiento entre otros. Esto hace que la experiencia del usuario sea mucho más positiva que con otro tipo de *App*.

Además, las aplicaciones nativas no necesitan conexión a internet para que funcionen. La descarga e instalación de estas *apps* se realiza siempre a través de las tiendas de aplicaciones (*app store* de los fabricantes).

En la tabla 5, se indican algunas ventajas e inconvenientes de las aplicaciones nativas.

Tabla 5. Aplicaciones nativas ventajas y desventajas

Ventajas	Inconvenientes
Acceso completo al dispositivo / Envío de notificaciones o avisos a los usuarios	Diferentes habilidades / Idiomas / herramienta para cada plataforma de destino
Mejor experiencia del usuario	Tienden hacer más caras de desarrollar
Visibilidad en <i>app store</i> / La actualización de la <i>app</i> es constante	El código del cliente no es reutilizable entre las diferentes plataformas

- Aplicación híbrida:** Una aplicación *Web* híbrida es una combinación de *app web* y *app nativa*, se podría decir que recoge lo mejor de cada una de ellas. Las *apps* híbridas se desarrollan con lenguajes propios de la web *App*, es decir, *HTML*, *Javascript* y *CSS* por lo que permite su uso en diferentes plataformas, pero también dan la posibilidad de acceder a gran parte de las características del *hardware* del dispositivo. La principal ventaja es que a pesar de estar desarrollada con *HTML*, *Java* o *CSS*, es posible agrupar los códigos y distribuirla en *app store*.

PhoneGap es uno de los *Frameworks* más utilizados por los programadores para el desarrollo multiplataforma de aplicaciones híbridas. Otro ejemplo de herramienta para desarrollar *apps* híbridas es *Cordova*.

En la tabla 6, se indican algunas ventajas e inconvenientes de las aplicaciones híbridas.

Tabla 6. Aplicaciones híbrida ventajas y desventajas

Ventajas	Inconvenientes
Es posible distribuirla en todas las tiendas de iOS y Android / Instalación nativa pero construida con JavaScript, HTML y CSS	Experiencia del usuario más propia de la aplicación web que de la app nativa
El mismo código base para múltiples plataformas / Acceso a parte del hardware del dispositivo	Diseño visual no siempre relacionado con el sistema operativo en el que se muestre

1.9 LENGUAJE DE PROGRAMACIÓN

Un lenguaje de programación es un sistema estructurado y diseñado principalmente para que las máquinas y computadoras se entiendan entre sí y con los humanos. Contiene un conjunto de acciones consecutivas que el ordenador debe ejecutar.

Estos lenguajes de programación usan diferentes normas o bases y se utiliza para controlar cómo se comporta una máquina (por ejemplo: un ordenador), también pueden usarse para crear programas informáticos, etc.

1.9.1 Tipos de lenguajes de programación

1.9.1.1 Lenguajes de programación de bajo nivel: son aquellos utilizados fundamentalmente para controlar el Hardware del ordenador y dependen totalmente de la máquina y no se pueden utilizar en otras máquinas. Están orientados exclusivamente para la máquina; estos lenguajes son los que ordenan a la máquina operaciones fundamentales para que pueda funcionar. Utiliza básicamente ceros, unos y abreviaturas de letras. También se denominan lenguaje de código de máquina y son los más complicados de utilizar. Con este tipo de lenguajes programan la asignación y liberación de memoria, el uso de punteros, el poder usar pasó por valor y por referencia, la creación de tipos de datos, etc.

1.9.1.2 Los lenguajes de programación de alto nivel: Estos lenguajes son más parecidos al lenguaje natural humano y no dependen de la máquina y sirven fundamentalmente para crear programas informáticos que solucionan diferentes

problemas. Son los más usados por los programadores y por todo el mundo que realiza programas informáticos, por ejemplo:

- **C#.** Es un lenguaje de propósito general orientado a objetos creado por Microsoft para su plataforma *.NET*. Su sintaxis básica deriva de *C/C++* y utiliza el modelo de objetos de la plataforma *.NET* el cual es similar al de *Java* aunque incluye mejoras derivadas de otros lenguajes. *C#* fue diseñado para combinar el control a bajo nivel de lenguajes como *C* y la velocidad de programación de lenguajes como *Visual Basic*.
- **JavaScript.** Es un lenguaje interpretado orientado a los sitios *Web*, con una sintaxis semejante a la del lenguaje *Java*.

El lenguaje fue inventado por *Brendan Eich* en la empresa *Netscape Communications*, que es la que fabricó los primeros navegadores de *Internet* comerciales. Apareció por primera vez en el producto de *Netscape Navigator 2.0*.

Se utiliza en sitios *Web HTML*, para realizar tareas y operaciones en el marco de la aplicación cliente, los autores inicialmente lo llamaron *Mocha* y más tarde *LiveScript* en un anuncio conjunto entre *Sun Microsystems* y *Netscape* el 4 de diciembre de 1995.

En 1997, los autores propusieron *JavaScript* para que fuera adoptado como estándar de la *European Computer Manufactures Association (ECMA)*, que a pesar de su nombre no es europeo sino internacional, con sede en Ginebra. En junio de 1997, fue adoptado como un estándar *ECMA*, con el nombre de *ECMAScript*. Poco después también lo fue como un estándar ISO.

1.10 PARADIGMAS DE PROGRAMACIÓN

Los paradigmas de programación indican las diversas formas que a lo largo de la evolución de los lenguajes, han sido aceptadas como estilos para programar y para resolver los problemas por medio de una computadora.

Se muestra a continuación un resumen de los paradigmas de uso más extendido en la programación.

1.10.1 Programación por procedimientos: es el paradigma original de programación y quizá todavía el de uso más común. En él, el programador se concentra en el procedimiento, en el algoritmo requerido para llevar a cabo el cómputo deseado. Los lenguajes apoyan este paradigma proporcionando recursos para pasar argumentos a las funciones y devolviendo valores de las funciones. *FORTRAN* es el lenguaje de procedimientos original, *Pascal* y *C* son inventos posteriores que siguen la misma idea. La programación estructurada se considera como el componente principal de la programación por procedimientos.

1.10.2 Programación modular: Con los años, en el diseño de programas se dio mayor énfasis al diseño de procedimientos que a la organización de la información. Entre otras cosas esto refleja un aumento en el tamaño de los programas. La programación modular surge como un remedio a esta situación. A menudo, se aplica el término módulo a un conjunto de procedimientos afines junto con los datos que manipulan. Así, el paradigma de la programación modular consiste en:

- a. Establecer los módulos que se requieren para la resolución de un problema.
- b. Dividir el programa de modo que los procedimientos y los datos queden ocultos en módulos. Este paradigma también se conoce como principio de ocultación de procedimientos y datos. Aunque C++ no se diseñó específicamente para desarrollar la programación modular, su concepto de clase proporciona apoyo para el concepto de módulo.

1.10.3 Paradigma funcional: Este tipo de paradigma demuestra un tipo de programación en forma de funciones matemáticas. Los lenguajes puramente funcionales evitan efectos secundarios que pueden ser comunes en otros tipos de paradigmas. El objetivo es conseguir lenguajes expresivos y matemáticamente elegantes, en los que no sea necesario bajar al nivel de la máquina para describir el proceso llevado a cabo por el programa, y evitando el concepto de estado del cómputo. La secuencia de computaciones llevadas a cabo por el programa se regiría única y exclusivamente por la reescritura de definiciones más amplias a otras cada vez más concretas y definidas, usando lo que se denominan "definiciones dirigidas".

1.10.4 Programación orientada a objetos (OOP): el problema con la abstracción de datos es que no hay ninguna distinción entre las propiedades generales y las particulares de un conjunto de objetos. Expresar esta distinción y aprovecharla es lo que define a la *OOP* a través del concepto de herencia. El paradigma de la programación orientada a objetos es, entonces:

- Definir que clases se desean.
- Proporcionar un conjunto completo de operaciones para cada clase.
- Indicar explícitamente lo que los objetos de la clase tienen en común empleando el concepto de herencia.

En algunas áreas las posibilidades de la *OOP* son enormes. Sin embargo, en otras aplicaciones, como las que usan los tipos aritméticos básicos y los cálculos basados en ellos, se requiere únicamente la abstracción de datos y/o programación por procedimientos, por lo que los recursos necesarios para apoyar la *OOP* podrían sobrar.

1.11 HERRAMIENTAS DE DESARROLLO

Las herramientas de desarrollo son aquellos programas o aplicaciones que tengan cierta importancia en el desarrollo de un programa (programación). Pueden ser de importancia vital (como un ensamblador, un compilador o un editor) o de importancia secundaria, como una *IDE* (*Integrated Development Environment* – Entorno de Desarrollo Integrado).

1.11.1 Visual Studio (IDE): *Visual Studio* es un potente Entorno de Desarrollo Integrado (*IDE*) que asegura código de calidad durante todo el ciclo de vida de la aplicación, desde el diseño hasta la implementación. *Visual Studio* es una solución “todo en uno”, para sistemas operativos *Windows*. Soporta varios lenguajes de programación tales como *Visual C++*, *Visual C#*, *Visual J#*, y *Visual Basic .Net*.

Visual Studio, permite a los desarrolladores crear aplicaciones, sitios y aplicaciones *Web*, así como servicios *Web* en cualquier entorno que soporte la plataforma *.NET*. Así se pueden crear aplicaciones que se intercomunican entre estaciones de trabajo, sitios *Web* y dispositivos móviles.

Actualmente, *Visual Studio* provee las herramientas necesarias para desarrollar aplicaciones que necesitan estar siempre conectadas, aplicaciones para desarrolladores independientes, o para entorno empresarial.

Entre estas herramientas, se tiene:

- **Interfaz mejorada.** La interfaz ha sido rediseñada para que el flujo de trabajo sea más sencillo. Se ha optimizado el orden de las pestañas y las herramientas para mejorar al máximo la navegación en la aplicación.
- **Desarrollo con Web Dev mejorado.** Se han incluido nuevas plantillas, herramientas más eficientes de publicación y soporte para los nuevos estándares *HTML5*, *CSS3*, *ASP.NET*. Todo ello permite el desarrollo *Web* en escritorio y en entornos *móviles* para facilitar un mejor *Responsive Design*.

- **Desarrollo en la nube.** Con las aplicaciones nuevas de Visual Studio 2015 se pueden desarrollar aplicaciones enfocadas a la nube mediante servidores virtuales, limitados, para después migrar a servidores reales, esto permite la ejecución y simulación de las aplicaciones desarrolladas, y mejor distribución del almacenamiento y los recursos consumidos.
- **Mejoras en Sharepoint.** Se ha mejorado con plantillas, opciones de implementación y nuevos diseñadores para aprovechar mejor todas las posibilidades de *Sharepoint*. Incluye mejoras en las funciones *ALM*, y una nueva herramienta llamada *LightSwitch*.
- **Organización de proyectos.** Gracias a las importantes mejoras en *ALM*, el proceso de organización de proyectos para desarrollo de aplicaciones complejas es más sencillo. En la herramienta se han incluido formas de seguimiento de los que integran el proyecto. Estos pueden hacer comentarios y también interactuar para que el proyecto pueda ser monitorizado y revisado más ágilmente.

Aparte de generar aplicaciones de escritorio de alto rendimiento, en *Visual Studio* se pueden utilizar las eficaces herramientas de desarrollo basado en componentes y otras tecnologías para simplificar el diseño, desarrollo e implementación en equipos de soluciones empresariales.

1.12 SISTEMAS OPERATIVOS

- **Linux.** El sistema operativo *Linux*, nació en 1991, gracias a un estudiante de la Universidad de *Helsinki*. El éxito de *Linux* se basa en una idea ingeniosa de su creador, *L. Torvalds*: en inscribir su proyecto bajo términos de la licencia *GPL* y proponer a todos los programadores y otros *hackers* de *internet* que le ayudaran.
- El impulso de Linux proviene en gran parte del hueco que llenó en términos de núcleo en el proyecto *GNU*.
- **Android.** Es un sistema operativo basado en *Linux* diseñado principalmente para dispositivos *móviles* con pantalla *táctil*, como teléfonos inteligentes o *Tablet*.

El lanzamiento de *Android* como nueva plataforma para el desarrollo de aplicaciones *móviles* ha causado grandes expectativas y está teniendo una importante aceptación tanto por parte de los usuarios como de la industria.

Las aplicaciones se desarrollan habitualmente en el lenguaje *Java* con *Android Software Development Kit (Android SDK)*. Pero están disponibles otras herramientas de desarrollo, incluyendo un *Kit* de Desarrollo nativo para aplicaciones o extensiones en *C* o *C++*, *Google App Inventor* y un entorno visual para programadores novatos. El desarrollo de aplicaciones para *Android* no requiere aprender lenguajes complejos de programación, todo lo que se necesita es un conocimiento aceptable de *Java* y estar en posesión del *Kit* de Desarrollo de *Software* o *SDK*.

- **Windows 8.** Es el Sistema Operativo actual de *Microsoft*, lanzado oficialmente al público en general el 26 de Octubre del 2011, luego de una serie de lanzamientos fallidos y presentaciones incompletas por problemas y “*Bugs*”. Lo que más destacó fue su cambio en la interfaz, principalmente por la eliminación del botón de inicio “*Home*” que desde sus inicios *Microsoft* incorporaba en la parte inferior izquierda de la pantalla y que este desplegaba un menú más corto en el que se podía acceder a toda carpeta y configuración del sistema también. Al poco tiempo se dieron cuenta que el botón resultó extremadamente querido por todas las generaciones y en la siguiente actualización se vieron obligados a incorporarlo nuevamente. Este sistema operativo no solamente es compatible con computadores y *laptops*, sino también con *tablets*, no se puede dejar de mencionar la compatibilidad con los procesadores *ARM* de baja potencia.

La figura 37, ilustra una interfaz del sistema operativo *Windows 8.1*.

Figura 37. Imagen de inicio de Windows 8.1



Microsoft introduce lo que más adelante sería un término tan popular en la tecnología que realmente no se esperaba, su interfaz y nuevo logo presentan características planas con formas cuadradas y simples que dan un aire de limpieza y facilidad a la hora de utilizar las aplicaciones. Todo esto de la mano de un soporte táctil con el que se puede explotar las características del Windows 8 al máximo. Claro que el Sistema Operativo puede ser utilizado con los periféricos estándar (Ratón o Mouse y Teclado) pero la experiencia se disfruta más con el soporte de gestos multitáctil que el sistema operativo incorpora.

2 DESARROLLO CAJA DE POLINOMIOS WEB MÓVIL

2.1 HISTORIAS DE USUARIOS

Prioridad. Se maneja en una escala de ALTA, MEDIA Y BAJA

2.2.1 Historias de usuarios a partir del desarrollador

2.2.1.1. Historia de usuario Fichas caja de polinomios web móvil

Tabla 7. Historia de usuario fichas caja de polinomios web móvil

Historia de Usuario Nº. 0001				5/08/2015	
Nombre	Diseñar las Fichas que permitan operar los Polinomios en el software	Prioridad	ALTA	Release	01
Descripción	El software debe proveer las fichas correspondientes para el diseño de operaciones en la caja de polinomios hasta de grado cuatro.				
Estimación	1 día	Dependencia	N/A	Iteración	1
Actualización			Base		
Usuario	Dorian Fernández Diego Ocoró	Rol		Desarrollador	

2.2.1.2. Historia de usuario Espacio de trabajo

Tabla 8. Historia de usuario espacio de trabajo

Historia de Usuario Nº. 0002					5/08/2015	
Nombre	Diseñar el Espacio de trabajo		Prioridad	ALTA	Release	01
Descripción	El software debe proveer un espacio de trabajo donde se encontrara tanto el tablero de la Caja de polinomios como las fichas a utilizar.					
Estimación	1 día	Dependencia			Iteración	1
Actualización			Base			
Usuario	Dorian Fernández Diego Ocoró		Rol		Desarrollador	

2.2.1.3. Historia de usuario Traducción simbólica

Tabla 9. Historia de usuario traducción simbólica

Historia de Usuario Nº. 0003					6/15/2015	
Nombre	Mostrar Traducción simbólica		Prioridad	MEDIA	Release	02
Descripción	El software deberá traducir las fichas contenidas en el tablero a lenguaje simbólico correspondiente para cada polinomio $P(x)$ y $Q(x)$.					
Estimación	3 días	Dependencia			Iteración	2
Actualización			Base			
Usuario	Dorian Fernández Diego Ocoró		Rol		Desarrollador	

2.2.1.4. Historia de usuario Ayudas

Tabla 10. Historia de usuario ayudas

Historia de Usuario N°. 0008					6/28/2015	
Nombre	Presentar Documentación soporte	Prioridad	BAJA	Release	04	
Descripción	El software deberá proveer gifs animados para guiar al usuario en cuanto a funcionamiento de la caja de polinomios web móvil para cada una de las operaciones.					
Estimación	2 semana	Dependencia		Iteración	4	
Actualización			Base			
Usuario	Dorian Fernández Diego Ocoró		Rol	Desarrollador		

2.2.1.5. Historia de usuario Suma Caja de polinomios Web Móvil

Tabla 11. Historia de usuario suma caja de polinomios web móvil

Historia de Usuario N°. 0010					6/28/2015	
Nombre	Suma Caja de polinomios Web Móvil	Prioridad	ALTA	Release	02	
Descripción	El software debe permitir al usuario realizar sumas entre polinomios.					
Estimación	1 semana	Dependencia	Historia de Usuarios N°. 0001, 0002, 0003, 0004, 0005	Iteración	2	
Actualización			Base			
Usuario	Dorian Fernández Diego Ocoró		Rol	Desarrollador		

2.2.1.6. Historia de usuario Resta Caja de polinomios Web Móvil

Tabla 12. Historia de usuario resta caja de polinomios web móvil

Historia de Usuario Nº. 0012					6/28/2015	
Nombre	Resta Caja de polinomios Web Móvil	Prioridad	ALTA	Release	02	
Descripción	El software deberá permitirle al usuario realizar restas entre polinomios.					
Estimación	1 semana	Dependencia	Historia de Usuarios Nº. 0001, 0002, 0003, 0004, 0005	Iteración	2	
Actualización			Base			
Usuario	Dorian Fernández Diego Ocoró	Rol			Desarrollador	

2.2.1.7. Historia de usuario División Caja de polinomios Web Móvil

Tabla 13. Historia de usuario división caja de polinomios web móvil

Historia de Usuario Nº. 0014					6/28/2015	
Nombre	División Caja de polinomios Web Móvil	Prioridad	MEDIA	Release	03	
Descripción	El software deberá permitirle al usuario realizar división entre polinomios.					
Estimación	1 semana	Dependencia	Historia de Usuarios Nº. 0001, 0002, 0003, 0004, 0005	Iteración	3	
Actualización			Base			
Usuario	Dorian Fernández Diego Ocoró	Rol			Desarrollador	

2.2.1.8. Historia de usuario multiplicación caja de polinomios web móvil

Tabla 14. Historia de usuario multiplicación caja de polinomios web móvil

Historia de Usuario N°. 0016				6/28/2015	
Nombre	Multiplicación Caja de polinomios Web Móvil	Prioridad	MEDIA	Release	03
Descripción	El software deberá permitirle al usuario realizar multiplicación entre polinomios.				
Estimación	1 semana	Dependencia	Historia de Usuarios N°. 0001, 0002, 0003, 0004, 0005	Iteración	3
Actualización			Base		
Usuario	Dorian Fernández Diego Ocoró	Rol		Desarrollador	

2.2.1.9. Historia de usuario factorización caja de polinomios web móvil

Tabla 15. Historia de usuario factorización caja de polinomios web móvil

Historia de Usuario N°. 0018				6/28/2015	
Nombre	Factorización Caja de polinomios Web Móvil	Prioridad	MEDIA	Release	03
Descripción	El software deberá permitirle al usuario realizar la factorización de un polinomio.				
Estimación	1 semana	Dependencia	Historia de Usuarios N°. 0001, 0002, 0003, 0004, 0005	Iteración	3
Actualización			Base		
Usuario	Dorian Fernández Diego Ocoró	Rol		Desarrollador	

2.2.2. Historias de usuarios a partir del cliente

2.2.2.1. Historia de usuario Mover fichas

Tabla 16. Historia de usuario mover fichas

Historia de Usuario N°. 0004					6/15/2015	
Nombre	Mover fichas	Prioridad	ALTA	Release	02	
Descripción	El usuario podrá mover diferentes fichas dentro del espacio de trabajo.					
Estimación	1 semana	Dependencia		Iteración	2	
Actualización			Base			
Usuario	Dorian Fernández Diego Ocoró	Rol	Desarrollador			

2.2.2.2. Historia de usuario Eliminar fichas

Tabla 17. Historia de usuario eliminar fichas

Historia de Usuario N°. 0005					6/15/2015	
Nombre	Eliminar fichas	Prioridad	BAJA	Release	02	
Descripción	El usuario podrá eliminar fichas dentro del espacio de trabajo asignado.					
Estimación	6 horas	Dependencia		Iteración	2	
Actualización			Base			
Usuario	Dorian Fernández Diego Ocoró	Rol	Desarrollador			

2.2.2.3. Historia de usuario Escoger operación

Tabla 18. Historia de usuario escoger operación

Historia de Usuario Nº. 0006				6/28/2015	
Nombre	Escoger Operación	Prioridad	MEDIA	Release	02
Descripción	El usuario podrá escoger que operación realizar (Suma, Resta, Multiplicación, División o Factorización).				
Estimación	3 horas	Dependencia		Iteración	2
Actualización			Base		
Usuario	Dorian Fernández Diego Ocoró	Rol	Desarrollador		

2.2.2.4. Historia de usuario Reiniciar operación

Tabla 19. Historia de usuario reiniciar operación

Historia de Usuario Nº. 0007				6/28/2015	
Nombre	Reiniciar operación	Prioridad	BAJA	Release	04
Descripción	El usuario podrá iniciar una nueva operación cada vez que finalice la presente.				
Estimación	1 hora	Dependencia		Iteración	4
Actualización			Base		
Usuario	Dorian Fernández Diego Ocoró	Rol	Desarrollador		

2.2.2.5. Historia de usuario Animaciones-guías

Tabla 20. Historia de usuario animaciones guías

Historia de Usuario N°. 0009					6/28/2015	
Nombre	Manual de Usuario	Prioridad	BAJA	Release	04	
Descripción	El usuario podrá acceder a los gifs animados en cada una de las operaciones algebraicas.					
Estimación	1 hora	Dependencia	Historia de Usuario N°. 0008	Iteración	4	
Actualización			Base			
Usuario	Dorian Fernández Diego Ocoró	Rol		Desarrollador		

2.2.2.6. Historia de usuario Suma usuario

Tabla 21. Historia de usuario suma usuario

Historia de Usuario N°. 0011					6/28/2015	
Nombre	Suma usuario	Prioridad	ALTA	Release	0 2	
Descripción	El usuario podrá realizar sumas entre polinomios de acuerdo a la metodología de la caja de polinomios.					
Estimación	1 hora	Dependencia	Historia de Usuario N°. 0010	Iteración	2	
Actualización			Base			
Usuario	Dorian Fernández Diego Ocoró	Rol		Desarrollador		

2.2.2.7. Historia de usuario Resta usuario

Tabla 22. Historia de usuario resta usuario

Historia de Usuario Nº. 0013					6/28/2015	
Nombre	Resta usuario		Prioridad	ALTA	Release	02
Descripción	El usuario podrá realizar restas entre polinomios de acuerdo a la metodología de la caja de polinomios.					
Estimación	1 hora	Dependencia	Historia de Usuario Nº. 0012	Iteración	2	
Actualización			Base			
Usuario	Dorian Fernández Diego Ocoró		Rol	Desarrollador		

2.2.2.8. Historia de usuario División usuario

Tabla 23. Historia de usuario división usuario

Historia de Usuario Nº. 0015					6/28/2015	
Nombre	División usuario		Prioridad	MEDIA	Release	03
Descripción	El usuario podrá realizar división entre polinomios de acuerdo a la metodología de la caja de polinomios.					
Estimación	1 hora	Dependencia	Historia de Usuario Nº. 0014	Iteración	3	
Actualización			Base			
Usuario	Dorian Fernández Diego Ocoró		Rol	Desarrollador		

2.2.2.9. Historia de usuario Multiplicación usuario

Tabla 24. Historia de usuario multiplicación usuario

Historia de Usuario N°. 0017					6/28/2015	
Nombre	Multiplicación usuario	Prioridad	MEDIA	Release	03	
Descripción	El usuario podrá realizar multiplicación entre polinomios de acuerdo a la metodología de la caja de polinomios.					
Estimación	1 hora	Dependencia	Historia de Usuario N°. 0016	Iteración	3	
Actualización			Base			
Usuario	Dorian Fernández Diego Ocoró	Rol			Desarrollador	

2.2.2.10. Historia de usuario Factorización usuario

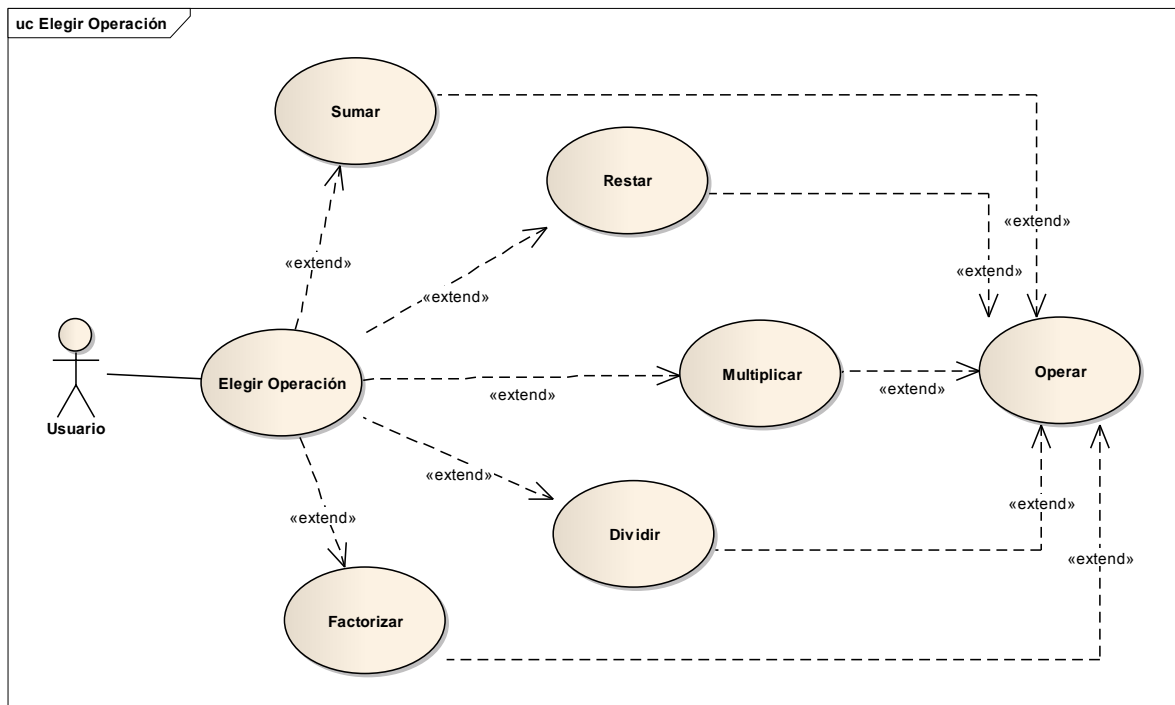
Tabla 25. Historia de usuario factorización usuario

Historia de Usuario N°. 0019					6/28/2015	
Nombre	Factorización usuario	Prioridad	MEDIA	Release	03	
Descripción	El usuario podrá realizar factorización de un polinomio de acuerdo a la metodología de la caja de polinomios.					
Estimación	1 hora	Dependencia	Historia de Usuario N°. 0018	Iteración	3	
Actualización			Base			
Usuario	Dorian Fernández Diego Ocoró	Rol			Desarrollador	

2.3. Diagramas casos de usos

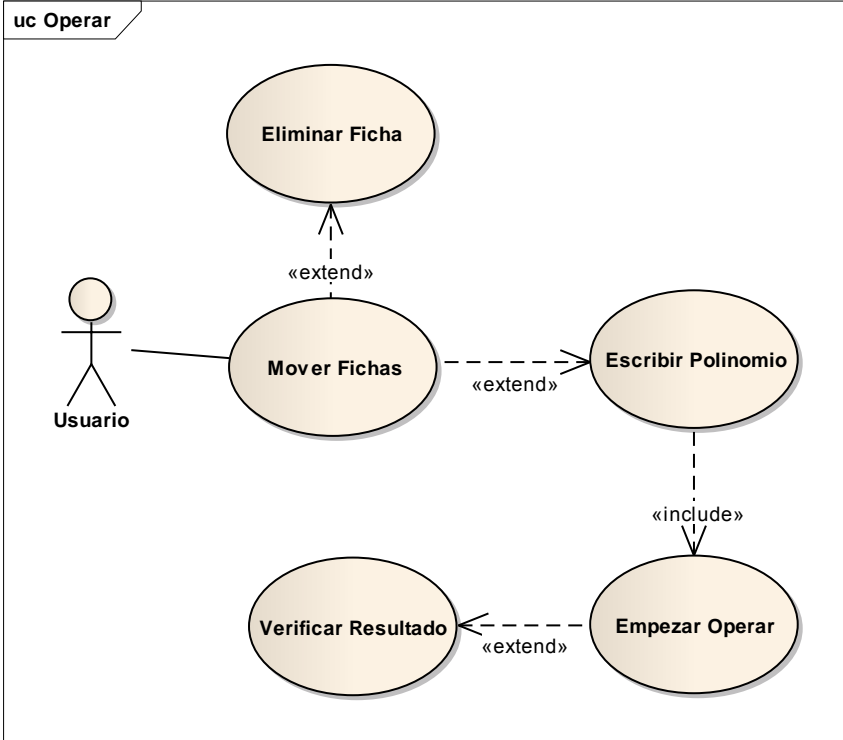
2.3.1. Caso de uso Iniciar operaciones

Figura 38. Caso de uso iniciar operación



2.3.2. Caso de uso Operar

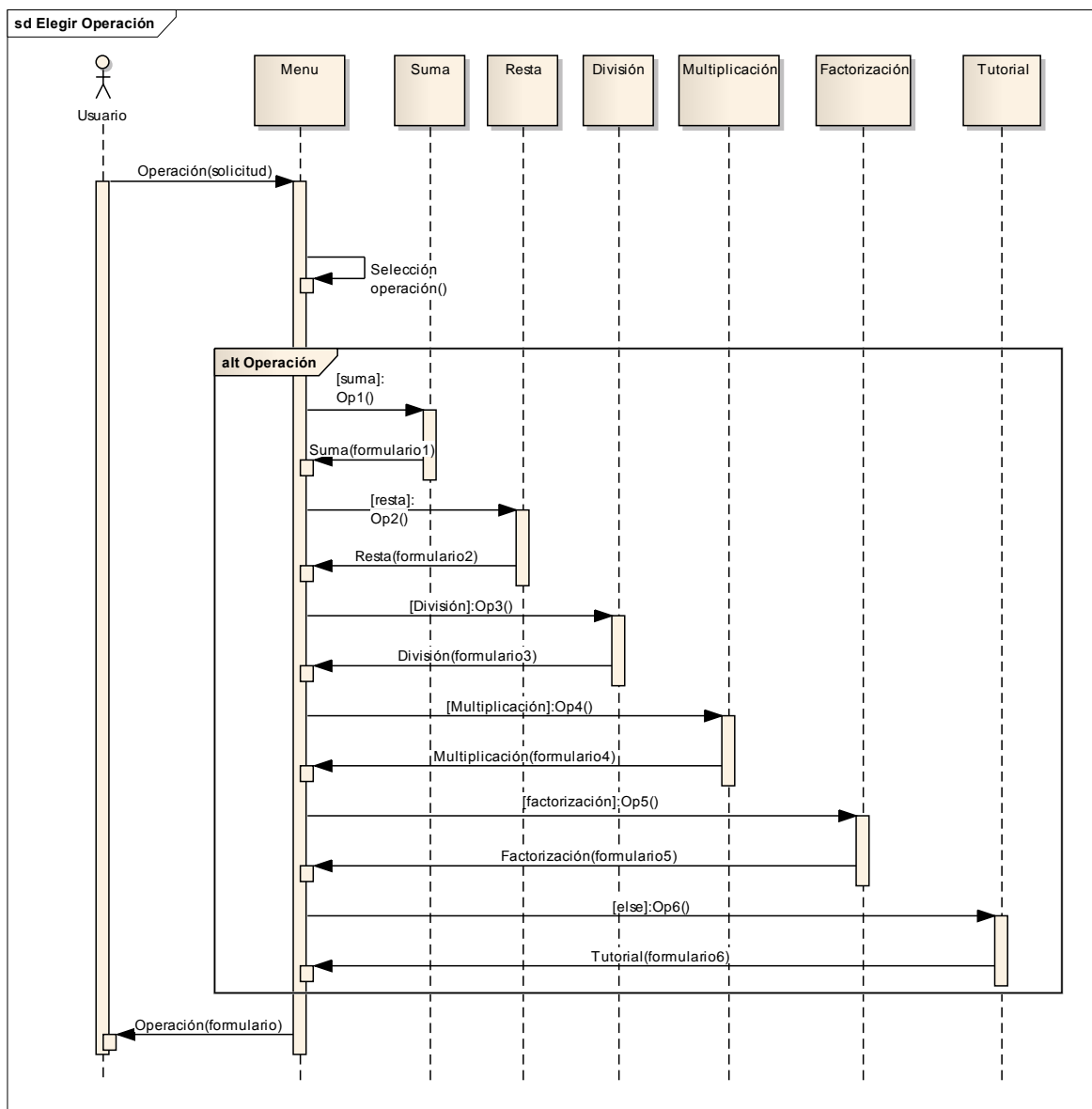
Figura 39. Caso de uso operar



2.4. Diagramas de secuencia

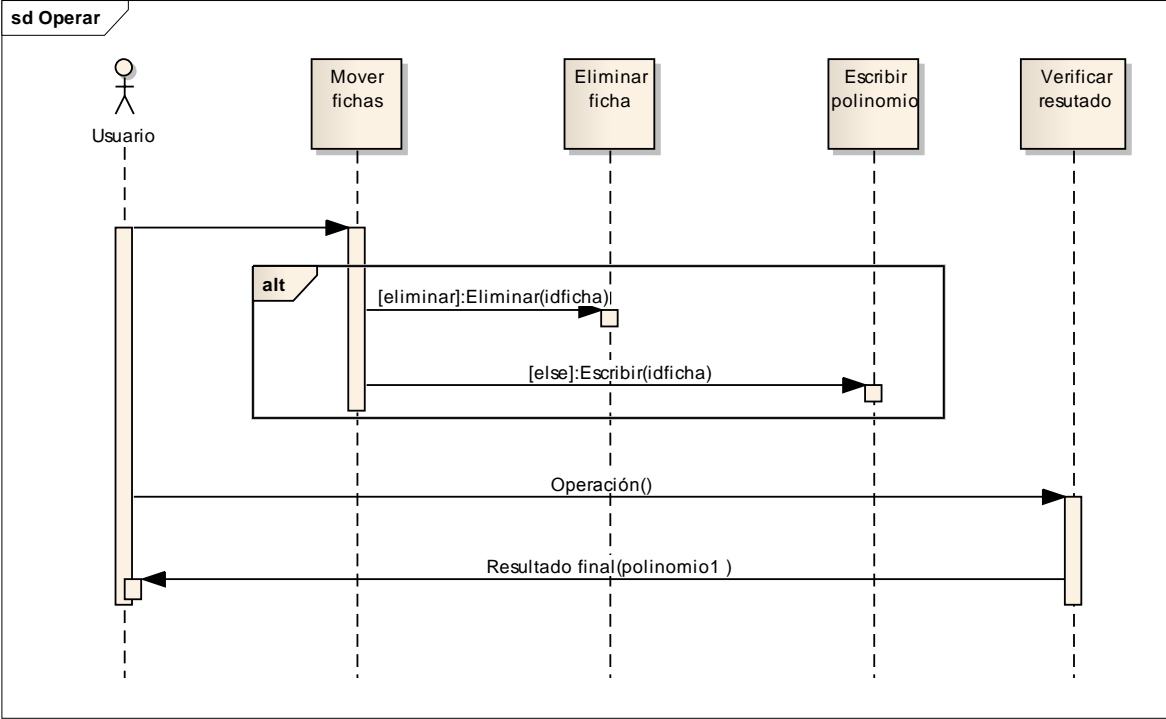
2.4.1. Diagrama de secuencia elegir operación

Figura 40. Diagrama de secuencia elegir operación



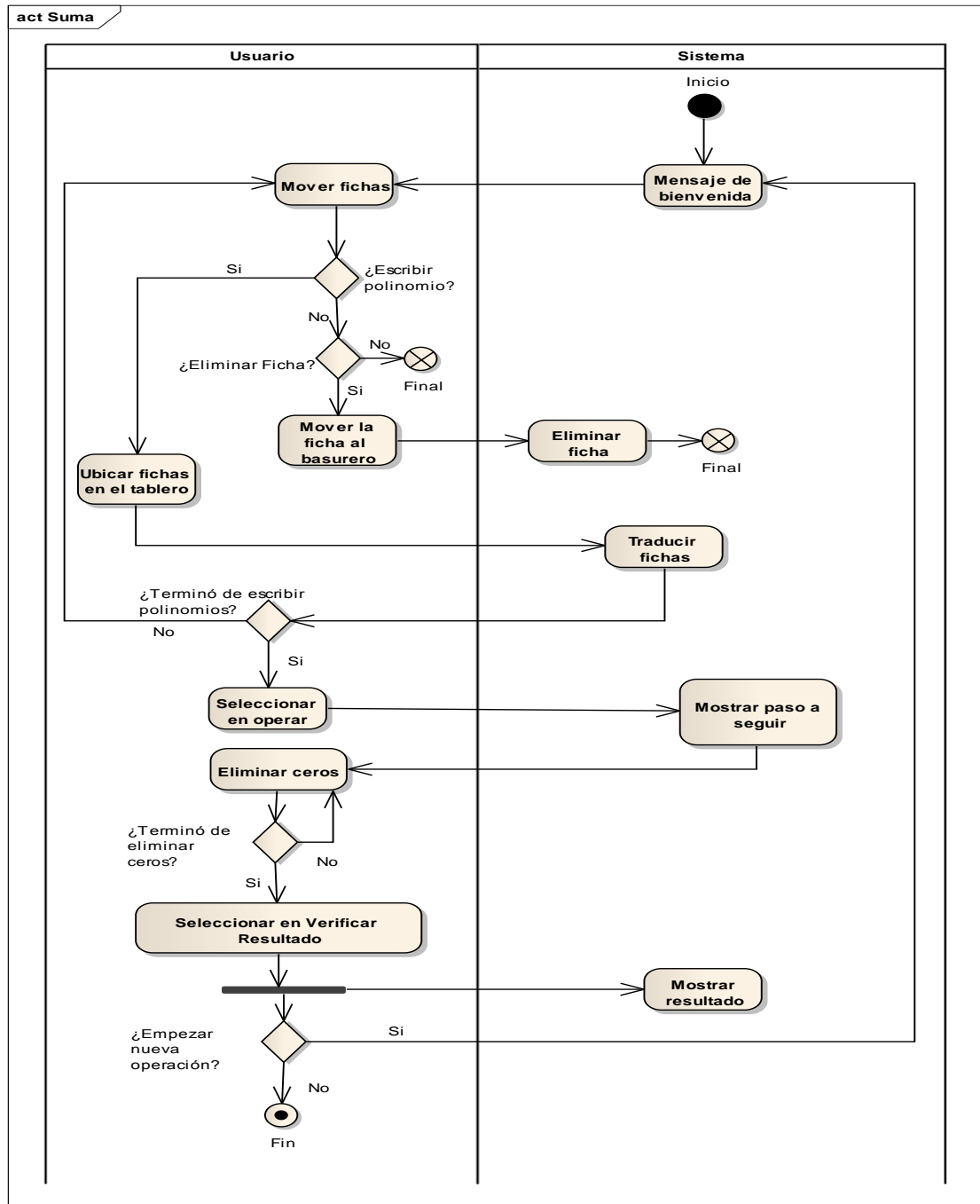
2.4.2 Diagrama de secuencia operar

Figura 41. Diagrama de secuencia operar



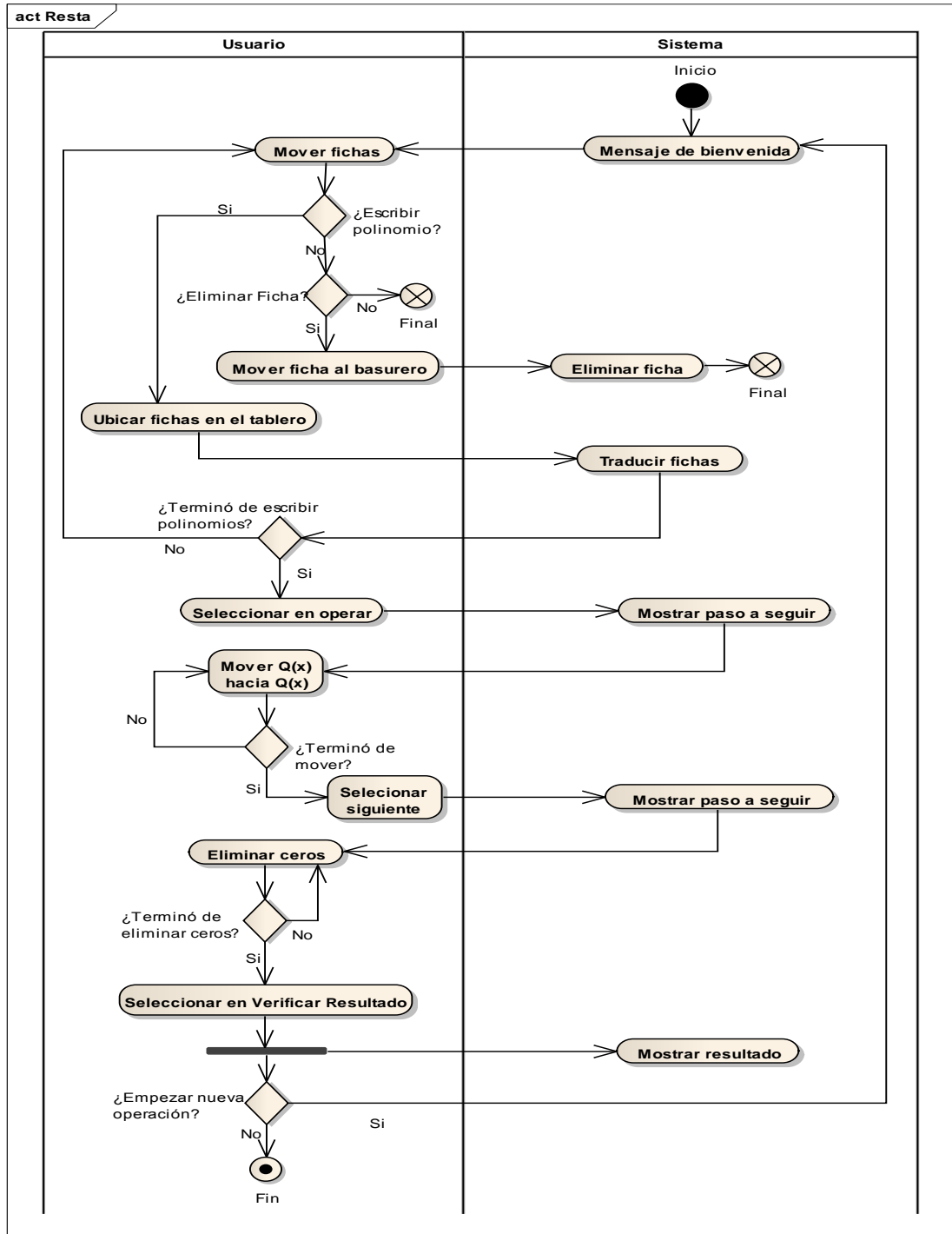
2.5. Diagramas de actividades
 2.5.1 Diagrama de actividad suma

Figura 42. Diagrama de actividad suma



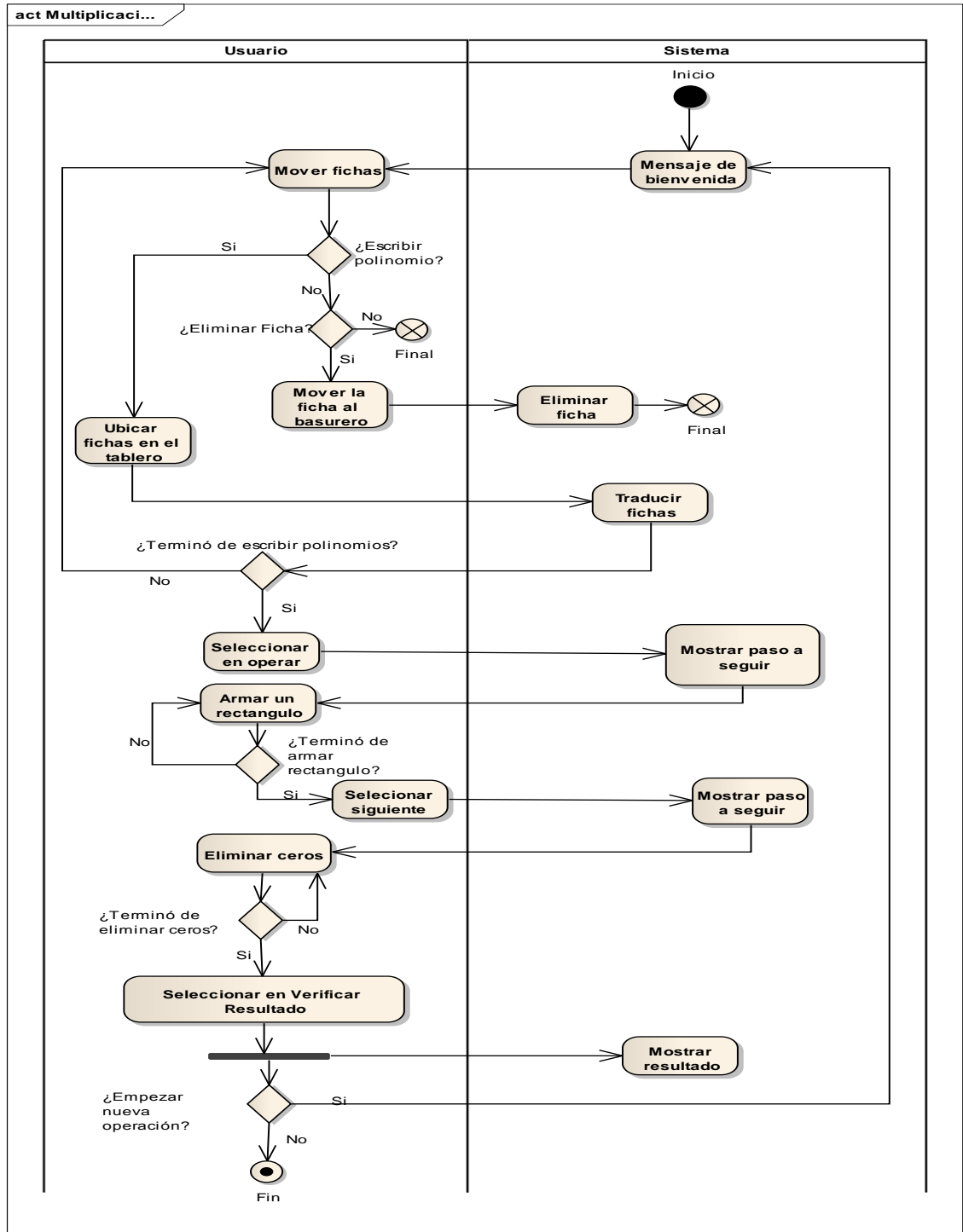
2.5.2 Diagrama de actividad resta

Figura 43. Diagrama de actividad resta



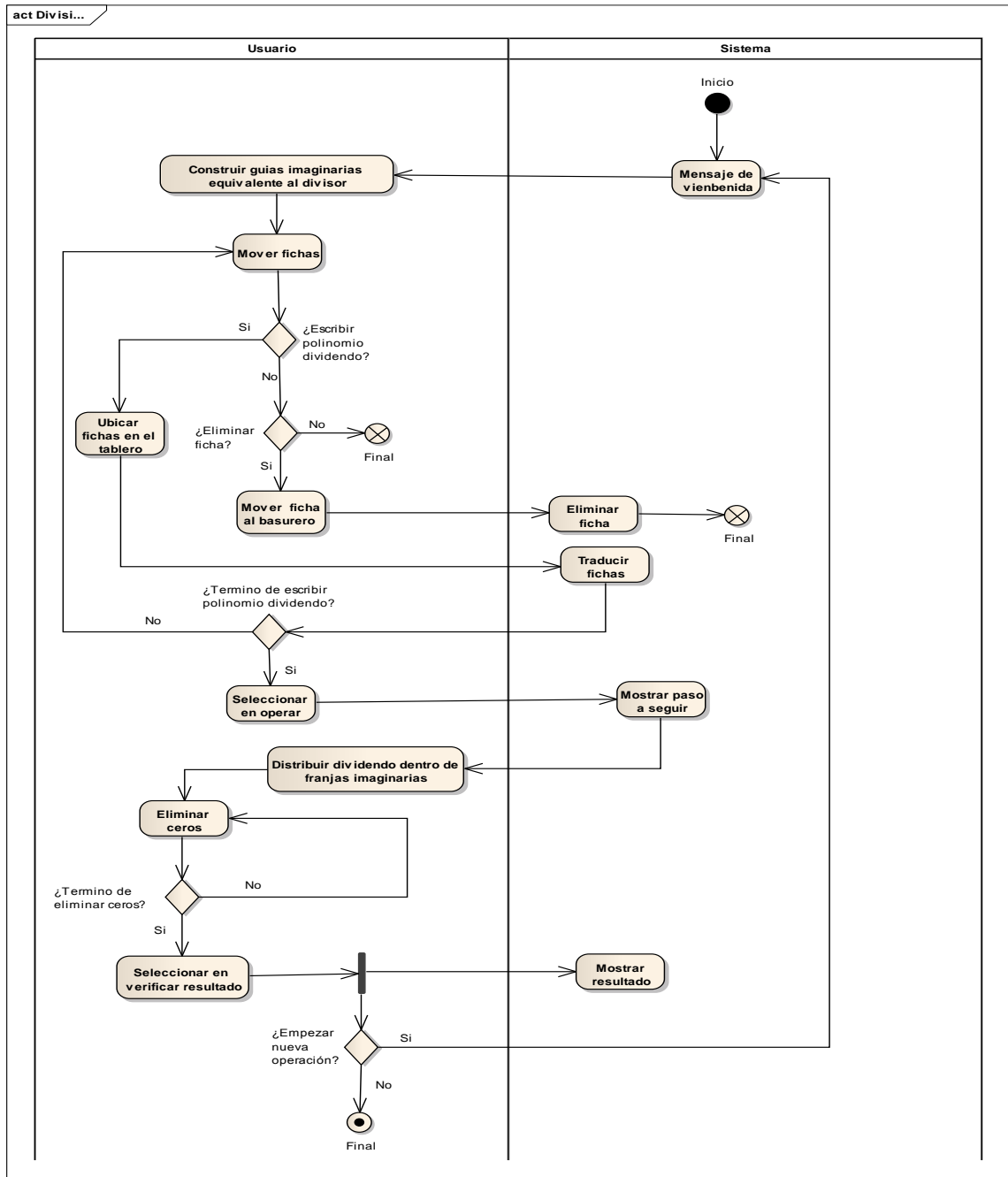
2.5.3 Diagrama de actividad multiplicación

Figura 44. Diagrama de actividad multiplicación



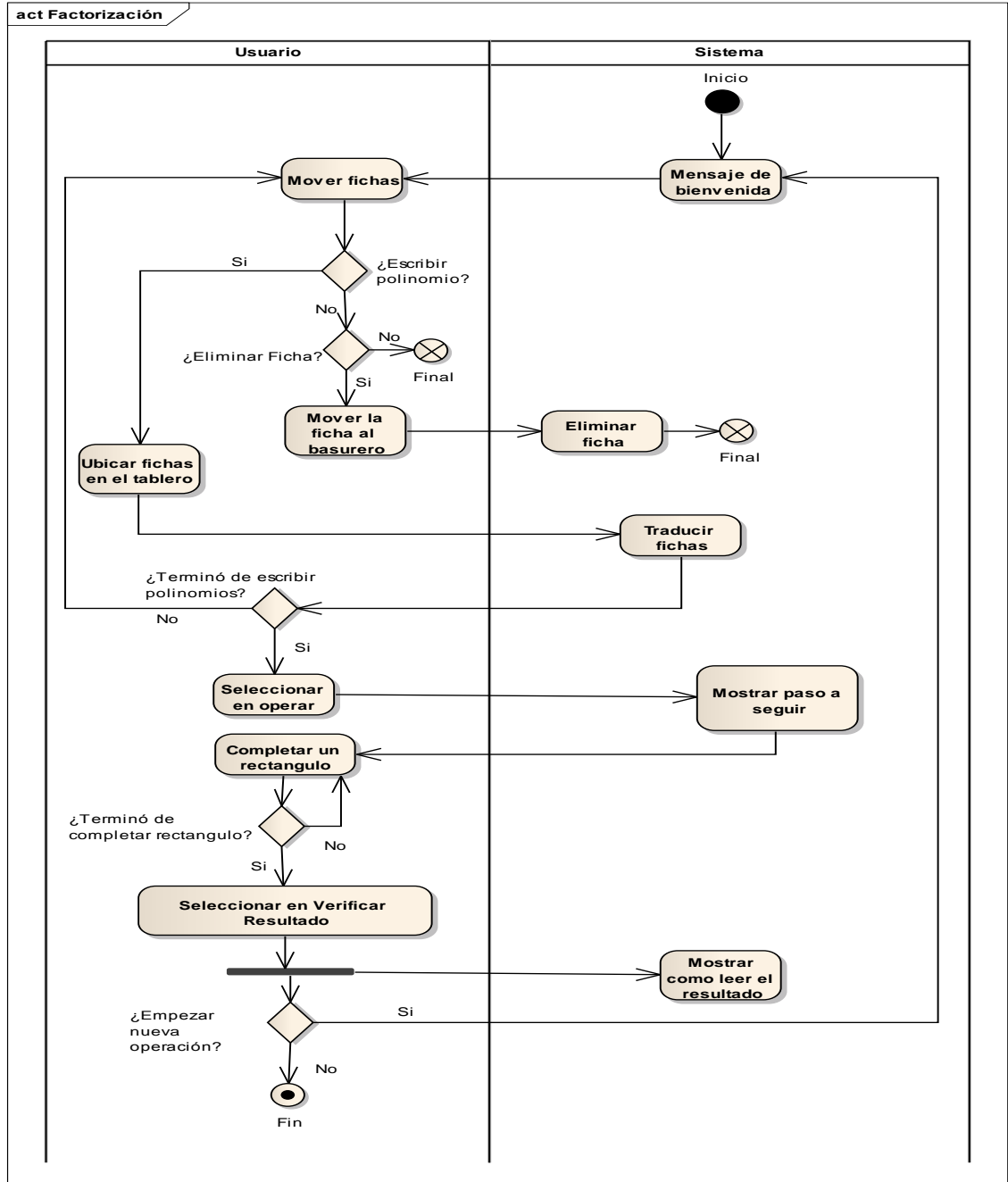
2.5.4 Diagrama de actividad división

Figura 45. Diagrama de actividad división



2.5.5 Diagrama de actividad factorización

Figura 46. Diagrama de actividad factorización



2.6 Código principal de la aplicación

2.6.1 Agregar ficha en la caja de polinomios. Esta función permite crear una ficha que sea arrastrada en la caja de polinomios, identificando los respectivos atributos de cada ficha.

```
function Generar_Ficha(lf, tp, idficha) {
    var img = document.createElement("img");

    switch (idficha) {
        case "uno":
            img.setAttribute("src", "images/num1.png");
            img.setAttribute("id", "uno");
            break;
        case "axuno":
            img.setAttribute("src", "images/var_x.png");
            img.setAttribute("id", "axuno");
            break;
        case "bxuno":
            img.setAttribute("src", "images/ver_x.png");
            img.setAttribute("id", "bxuno");
            break;
        case "axdos":
            img.setAttribute("src", "images/x^2.png");
            img.setAttribute("id", "axdos");
            break;
        case "bxdos":
            img.setAttribute("src", "images/lado1_x^2.png");
            img.setAttribute("id", "bxdos");
            break;
        case "axtres":
            img.setAttribute("src", "images/x^3.png");
            img.setAttribute("id", "axtres");
            break;
        case "bxtres":
            img.setAttribute("src", "images/ladox_x^3.png");
            img.setAttribute("id", "bxtres");
            break;
        case "cxdos":
            img.setAttribute("src", "images/ladox^2_1.png");
            img.setAttribute("id", "cxdos");
            break;
        case "cxtres":
            img.setAttribute("src", "images/lado1_x^3.png");
    }
}
```

```

        img.setAttribute("id", "cxtres");
        break;
    case "dxtres":
        img.setAttribute("src", "images/ladox^3_1.png");
        img.setAttribute("id", "dxtres");
        break;
    case "axcuatro":
        img.setAttribute("src", "images/ladox^3_x.png");
        img.setAttribute("id", "axcuatro");
        break;
    case "bxcuatro":
        img.setAttribute("src", "images/ladox_x^4.png");
        img.setAttribute("id", "bxcuatro");
        break;
    case "cxcuatro":
        img.setAttribute("src", "images/x^4.png");
        img.setAttribute("id", "cxcuatro");
        break;
    }

    img.setAttribute("class", "draged");
    img.style.left = lf + "px";
    img.style.top = tp + "px";
    document.body.appendChild(img);

    $(".draged").draggable();
    $(".draged").data("soltado", true);
    $(".draged").data("px", false);
    $(".draged").data("qx", false);
}

```

2.6.2 Establecer la propiedad *draggable* a las fichas de la caja de polinomios. Función que le agrega la propiedad de *draggable* a las fichas; esto es hacer que las fichas se puedan mover o arrastrar dentro de la caja de polinomios.

```

function dragFicha(){
    $(".drag").draggable({
        helper: "clone", cursor: "pointer",
        stop: function (ev, ui) {
            $("#" + this.id).data("soltado", false);
        }
    });
}

```

2.6.3 Establecer la propiedad droppable. Función que le agrega la propiedad de *droppable* a cada uno de los cuadrantes de la caja de polinomios; estos cuadrantes son los receptores de cada una de las fichas que se suelten y salgan de ellos y se encargan de llevar el número de cada una de las fichas del mismo tipo soltadas dentro de la caja de polinomios.

```
function dropCuadrantes(esResta_Suma,esMultiplicacion) {
    $("#cuadrante2").droppable({
        drop: function (ev, ui) {
            pos = $(ui.helper).offset();

            if (ui.draggable.attr('class') == "drag ui-draggable ui-draggable-handle")
                Generar_Ficha(pos.left, pos.top, ui.draggable.attr('id'));
            if (!ui.draggable.data("soltado")) {
                ui.draggable.data("soltado", true);

                switch (ui.draggable.attr('id')) {
                    case "uno":
                        unos1 = unos1 - 1;
                        break;
                    case "axuno":
                        xunos1 = xunos1 - 1;
                        break;
                    case "bxuno":
                        xunos1 = xunos1 - 1;
                        break;
                    case "axdos":
                        xdos1 = xdos1 - 1;
                        break;
                    case "bxdos":
                        xdos1 = xdos1 - 1;
                        break;
                    case "cxdos":
                        xdos1 = xdos1 - 1;
                        break;
                    case "axtres":
                        xtres1 = xtres1 - 1;
                        break;
                    case "bxtres":
                        xtres1 = xtres1 - 1;
                        break;
                    case "cxtres":
                        xtres1 = xtres1 - 1;
                }
            }
        }
    });
}
```

```

        break;
    case "dxtres":
        xtres1 = xtres1 - 1;
        break;
    case "axcuatro":
        xcuatro1 = xcuatro1 - 1;
        break;
    case "bxcuatro":
        xcuatro1 = xcuatro1 - 1;
        break;
    case "cxcuatro":
        xcuatro1 = xcuatro1 - 1;
        break;
    }
}
if (!esMultiplicacion) {
    if (esResta_Suma && !operando) {
        document.getElementById("Tpx").innerHTML = '<strong>P(X) =
</strong>' + getExpresion(1);
    }
    else {
        if (!esResta_Suma && !operando) {
            document.getElementById("Tpx").innerHTML = '<strong>P(X) =
</strong>' + getResultado();
        }
    }
}
},

out: function (event, ui) {
    if (ui.draggable.data("soltado")) {
        ui.draggable.data("soltado", false);
        switch (ui.draggable.attr('id')) {
            case "uno":
                unos1 = unos1 + 1;
                break;
            case "axuno":
                xunos1 = xunos1 + 1;
                break;
            case "bxuno":
                xunos1 = xunos1 + 1;
                break;
            case "axdos":
                xdos1 = xdos1 + 1;
                break;
        }
    }
}

```

```

        case "bxdos":
            xdos1 = xdos1 + 1;
            break;
        case "cxdos":
            xdos1 = xdos1 + 1;
            break;
        case "axtres":
            xtres1 = xtres1 + 1;
            break;
        case "bxtres":
            xtres1 = xtres1 + 1;
            break;
        case "cxtres":
            xtres1 = xtres1 + 1;
            break;
        case "dxtres":
            xtres1 = xtres1 + 1;
            break;
        case "axcuatro":
            xcuatro1 = xcuatro1 + 1;
            break;
        case "bxcuatro":
            xcuatro1 = xcuatro1 + 1;
            break;
        case "cxcuatro":
            xcuatro1 = xcuatro1 + 1;
            break;
    }
}
if (!esMultiplicacion) {
    if (esResta_Suma && !operando) {
        document.getElementById("Tpx").innerHTML = '<strong>P(X) =
</strong>' + getExpresion(1);
    }
    else {
        if (!esResta_Suma && !operando) {
            document.getElementById("Tpx").innerHTML = '<strong>P(X) =
</strong>' + getResultado();
        }
    }
}
});

```

```

$("#cuadrante3").droppable({
  drop: function (ev, ui) {
    pos = $(ui.helper).offset();

    if (ui.draggable.attr('class') == "drag ui-draggable ui-draggable-handle")
    Generar_Ficha(pos.left, pos.top, ui.draggable.attr('id'));
    if (!ui.draggable.data("soltado")) {
      ui.draggable.data("soltado", true);

      switch (ui.draggable.attr('id')) {
        case "uno":
          unos1 = unos1 + 1;
          break;
        case "axuno":
          xunos1 = xunos1 + 1;
          break;
        case "bxuno":
          xunos1 = xunos1 + 1;
          break;
        case "axdos":
          xdos1 = xdos1 + 1;
          break;
        case "bxdos":
          xdos1 = xdos1 + 1;
          break;
        case "cxdos":
          xdos1 = xdos1 + 1;
          break;
        case "axtres":
          xtres1 = xtres1 + 1;
          break;
        case "bxtres":
          xtres1 = xtres1 + 1;
          break;
        case "cxtres":
          xtres1 = xtres1 + 1;
          break;
        case "dxtres":
          xtres1 = xtres1 + 1;
          break;
        case "axcuatro":
          xcuatro1 = xcuatro1 + 1;
          break;
        case "bxcuatro":

```

```

        xcuatro1 = xcuatro1 + 1;
        break;
    case "cxcuatro":
        xcuatro1 = xcuatro1 + 1;
        break;
    }
}
if (!esMultiplicacion) {
    if (esResta_Suma && !operando) {
        document.getElementById("Tpx").innerHTML = '<strong>P(X) =
</strong>' + getExpresion(1);
    }
    else {
        if (!esResta_Suma && !operando) {
            document.getElementById("Tpx").innerHTML = '<strong>P(X) =
</strong>' + getResultado();
        }
    }
}
},

```

```

out: function (event, ui) {
    if (ui.draggable.data("soltado")) {
        ui.draggable.data("soltado", false);
        switch (ui.draggable.attr('id')) {
            case "uno":
                unos1 = unos1 - 1;
                break;
            case "axuno":
                xunos1 = xunos1 - 1;
                break;
            case "bxuno":
                xunos1 = xunos1 - 1;
                break;
            case "axdos":
                xdos1 = xdos1 - 1;
                break;
            case "bxdos":
                xdos1 = xdos1 - 1;
                break;
            case "cxdos":
                xdos1 = xdos1 - 1;
                break;
            case "axtres":
                xtres1 = xtres1 - 1;

```

```

        break;
    case "bxtres":
        xtres1 = xtres1 - 1;
        break;
    case "cxtres":
        xtres1 = xtres1 - 1;
        break;
    case "dxtres":
        xtres1 = xtres1 - 1;
        break;
    case "axcuatro":
        xcuatro1 = xcuatro1 - 1;
        break;
    case "bxcuatro":
        xcuatro1 = xcuatro1 - 1;
        break;
    case "cxcuatro":
        xcuatro1 = xcuatro1 - 1;
        break;
    }
}
if (!esMultiplicacion) {
    if (esResta_Suma && !operando) {
        document.getElementById("Tpx").innerHTML = '<strong>P(X) =
</strong>' + getExpresion(1);
    }
    else {
        if (!esResta_Suma && !operando) {
            document.getElementById("Tpx").innerHTML = '<strong>P(X) =
</strong>' + getResultado();
        }
    }
}
});

$("#cuadrante1").droppable({
    drop: function (ev, ui) {
        pos = $(ui.helper).offset();

        if (ui.draggable.attr('class') == "drag ui-draggable ui-draggable-handle")
Generar_Ficha(pos.left, pos.top, ui.draggable.attr('id'));
        if (!ui.draggable.data("soltado")) {
            ui.draggable.data("soltado", true);
        }
    }
});

```



```

switch (ui.draggable.attr('id')) {
  case "uno":
    unos2 = unos2 + 1;
    break;
  case "axuno":
    xunos2 = xunos2 + 1;
    break;
  case "bxuno":
    xunos2 = xunos2 + 1;
    break;
  case "axdos":
    xdos2 = xdos2 + 1;
    break;
  case "bxdos":
    xdos2 = xdos2 + 1;
    break;
  case "cxdos":
    xdos2 = xdos2 + 1;
    break;
  case "axtres":
    xtres2 = xtres2 + 1;
    break;
  case "bxtres":
    xtres2 = xtres2 + 1;
    break;
  case "cxtres":
    xtres2 = xtres2 + 1;
    break;
  case "dxtres":
    xtres2 = xtres2 + 1;
    break;
  case "axcuatro":
    xcuatro2 = xcuatro2 + 1;
    break;
  case "bxcuatro":
    xcuatro2 = xcuatro2 + 1;
    break;
  case "cxcuatro":
    xcuatro2 = xcuatro2 + 1;
    break;
}
}
if (!esMultiplicacion) {
  if (esResta_Suma && !operando) {

```

```

        document.getElementById("Tqx").innerHTML = '<strong>Q(X) =
</strong>' + getExpresion(2);
    }
    else {
        if (!esResta_Suma && !operando) {
            document.getElementById("Tpx").innerHTML = '<strong>P(X) =
</strong>' + getResultado();
        }
    }
}
},

```

```

out: function (event, ui) {
    if (ui.draggable.data("soltado")) {
        ui.draggable.data("soltado", false);
        switch (ui.draggable.attr('id')) {
            case "uno":
                unos2 = unos2 - 1;
                break;
            case "axuno":
                xunos2 = xunos2 - 1;
                break;
            case "bxuno":
                xunos2 = xunos2 - 1;
                break;
            case "axdos":
                xdos2 = xdos2 - 1;
                break;
            case "bxdos":
                xdos2 = xdos2 - 1;
                break;
            case "cxdos":
                xdos2 = xdos2 - 1;
                break;
            case "axtres":
                xtres2 = xtres2 - 1;
                break;
            case "bxtres":
                xtres2 = xtres2 - 1;
                break;
            case "cxtres":
                xtres2 = xtres2 - 1;
                break;
            case "dxtres":
                xtres2 = xtres2 - 1;

```

```

        break;
    case "axcuatro":
        xcuatro2 = xcuatro2 - 1;
        break;
    case "bxcuatro":
        xcuatro2 = xcuatro2 - 1;
        break;
    case "cxcuatro":
        xcuatro2 = xcuatro2 - 1;
        break;
    }
}
if (!esMultiplicacion) {
    if (esResta_Suma && !operando) {
        document.getElementById("Tqx").innerHTML = '<strong>Q(X) =
</strong>' + getExpresion(2);
    }
    else {
        if (!esResta_Suma && !operando) {
            document.getElementById("Tpx").innerHTML = '<strong>P(X) =
</strong>' + getResultado();
        }
    }
}
});

$("#cuadrante4").droppable({
    drop: function (ev, ui) {
        pos = $(ui.helper).offset();

        if (ui.draggable.attr('class') == "drag ui-draggable ui-draggable-handle")
Generar_Ficha(pos.left, pos.top, ui.draggable.attr('id'));
        if (!ui.draggable.data("soltado")) {
            ui.draggable.data("soltado", true);

            switch (ui.draggable.attr('id')) {
                case "uno":
                    unos2 = unos2 - 1;
                    break;
                case "axuno":
                    xunos2 = xunos2 - 1;
                    break;
                case "bxuno":
                    xunos2 = xunos2 - 1;

```

```

        break;
    case "axdos":
        xdos2 = xdos2 - 1;
        break;
    case "bxdos":
        xdos2 = xdos2 - 1;
        break;
    case "cxdos":
        xdos2 = xdos2 - 1;
        break;
    case "axtres":
        xtres2 = xtres2 - 1;
        break;
    case "bxtres":
        xtres2 = xtres2 - 1;
        break;
    case "cxtres":
        xtres2 = xtres2 - 1;
        break;
    case "dxtres":
        xtres2 = xtres2 - 1;
        break;
    case "axcuatro":
        xcuatro2 = xcuatro2 - 1;
        break;
    case "bxcuatro":
        xcuatro2 = xcuatro2 - 1;
        break;
    case "cxcuatro":
        xcuatro2 = xcuatro2 - 1;
        break;
    }
}
if (!esMultiplicacion) {
    if (esResta_Suma && !operando) {
        document.getElementById("Tqx").innerHTML = '<strong>Q(X) =
</strong>' + getExpresion(2);
    }
    else {
        if (!esResta_Suma && !operando) {
            document.getElementById("Tpx").innerHTML = '<strong>P(X) =
</strong>' + getResultado();
        }
    }
}
}

```

```

},
out: function (event, ui) {
  if (ui.draggable.data("soltado")) {
    ui.draggable.data("soltado", false);
    switch (ui.draggable.attr('id')) {
      case "uno":
        unos2 = unos2 + 1;
        break;
      case "axuno":
        xunos2 = xunos2 + 1;
        break;
      case "bxuno":
        xunos2 = xunos2 + 1;
        break;
      case "axdos":
        xdos2 = xdos2 + 1;
        break;
      case "bxdos":
        xdos2 = xdos2 + 1;
        break;
      case "cxdos":
        xdos2 = xdos2 + 1;
        break;
      case "axtres":
        xtres2 = xtres2 + 1;
        break;
      case "bxtres":
        xtres2 = xtres2 + 1;
        break;
      case "cxtres":
        xtres2 = xtres2 + 1;
        break;
      case "dxtres":
        xtres2 = xtres2 + 1;
        break;
      case "axcuatro":
        xcuatro2 = xcuatro2 + 1;
        break;
      case "bxcuatro":
        xcuatro2 = xcuatro2 + 1;
        break;
      case "cxcuatro":
        xcuatro2 = xcuatro2 + 1;
        break;
    }
  }
}

```

```

    }
  }
  if (!esMultiplicacion) {
    if (esResta_Suma && !operando) {
      document.getElementById("Tqx").innerHTML = '<strong>Q(X) =
</strong>' + getExpresion(2);
    }
    else {
      if (!esResta_Suma && !operando) {
        document.getElementById("Tpx").innerHTML = '<strong>P(X) =
</strong>' + getResultado();
      }
    }
  }
}
});
}

```

2.6.4 Traducción simbólica de polinomios. Función que arma una expresión algebraica la cual retorna un dato tipo String, utilizada para traducir la cantidad de fichas agregadas en la caja de polinomios a una expresión simbólica.

```

function getExpresion(n) {
  var x4, x3, x, x1, c;

  if (this['xcuatro' + n] == 1) x4 = "x<sup>4</sup>";

  if (this['xtres' + n] == 1 && this['xcuatro' + n] != 0) {
    x3 = "+x<sup>3</sup>";
  }
  else {
    x3 = "x<sup>3</sup>";
  }
  if (this['xdos' + n] == 1 && (this['xtres' + n] != 0 || this['xcuatro' + n] != 0)) {
    x2 = "+x<sup>2</sup>";
  }
  else {
    x2 = "x<sup>2</sup>";
  }
  if (this['xunos' + n] == 1 && (this['xdos' + n] != 0 || this['xtres' + n] != 0 ||
this['xcuatro' + n] != 0)) {
    x1 = "+x";
  }
  else {

```

```

    x1 = "x";
}

if (this['xcuatro' + n] < -1) x4 = this['xcuatro' + n] + "x<sup>4</sup>";
if (this['xtres' + n] < -1) x3 = this['xtres' + n] + "x<sup>3</sup>";
if (this['xdos' + n] < -1) x2 = this['xdos' + n] + "x<sup>2</sup>";
if (this['xunos' + n] < -1) x1 = this['xunos' + n] + "x";

if (this['xcuatro' + n] > 1) x4 = this['xcuatro' + n] + "x<sup>4</sup>";

if (this['xtres' + n] > 1 && this['xcuatro' + n] != 0) {
    x3 = "+" + this['xtres' + n] + "x<sup>3</sup>";
}
else {
    if (this['xtres' + n] != 1) x3 = this['xtres' + n] + "x<sup>3</sup>";
}

if (this['xdos' + n] > 1 && (this['xtres' + n] != 0 || this['xcuatro' + n] != 0)) {
    x2 = "+" + this['xdos' + n] + "x<sup>2</sup>";
}
else {
    if (this['xdos' + n] != 1) x2 = this['xdos' + n] + "x<sup>2</sup>";
}

if (this['xunos' + n] > 1 && (this['xdos' + n] != 0 || this['xtres' + n] != 0 ||
this['xcuatro' + n] != 0)) {
    x1 = "+" + this['xunos' + n] + "x";
}
else {
    if (this['xunos' + n] != 1) x1 = this['xunos' + n] + "x";
}

if (this['unos' + n] >= 1 && (this['xunos' + n] != 0 || this['xdos' + n] != 0 || this['xtres'
+ n] != 0 || this['xcuatro' + n] != 0)) {
    c = "+" + this['unos' + n];
}
else {
    c = this['unos' + n];
}

if (this['xcuatro' + n] == -1) x4 = "-x<sup>4</sup>";
if (this['xtres' + n] == -1) x3 = "-x<sup>3</sup>";
if (this['xdos' + n] == -1) x2 = "-x<sup>2</sup>";
if (this['xunos' + n] == -1) x1 = "-x";

```

```

    if (this['xcuatro' + n] == 0 && this['xtres' + n] == 0 && this['xdos' + n] == 0 &&
    this['xunos' + n] == 0 && this['unos' + n] == 0) {
        return "" + 0;
    }
    else {
        if (this['xcuatro' + n] == 0) x4 = "";
        if (this['xtres' + n] == 0) x3 = "";
        if (this['xdos' + n] == 0) x2 = "";
        if (this['xunos' + n] == 0) x1 = "";
        if (this['unos' + n] == 0) c = "";

        return x4 + x3 + x2 + x1 + c;
    }
}
}

```

3 INTERFACES Y FUNCIONAMIENTO

La interfaz de la App cuenta con 7 interfaces de las cuales 5 son para las diferentes operaciones.

A continuación, se presenta los elementos principales:

- a) Icono: representación de la caja de polinomios web móvil. Ver figura 47.

Figura 47. Icono app.



- b) Imagen de Inicio: es el logo de la caja de polinomio web móvil. Ver figura 48.

Figura 48. Imagen de inicio app



- c) Tablero: representación virtual del plano de la caja de polinomio, elemento principal de toda la App junto a las fichas respectivas de la caja de polinomios. Ver figura 49.

Figura 49. Tablero caja de polinomio



- d) Fichas caja de polinomios: 15 diferentes fichas las cuales representan los valores de $1, x, x^2, x^3, x^4$ en cuadrados y rectángulos, como lo indica la figura 50.

Figura 50. Fichas caja de polinomio



- e) Representación de un Basurero: esto con el fin de que las fichas que se arrastren en él sean eliminadas. Ver figura 51, icono para eliminar fichas.

Figura 51. Representación de basurero



- f) Botón acción a seguir: es un botón que va cambiando a medida que se va desarrollando la operación e inicia con el rotulo *Operar*. Ver figura 52, botón acción a seguir.

Figura 52. Botón acción a seguir



3.1 MENÚ PRINCIPAL

Interfaz por la cual se acceden a las diferentes operaciones de la caja de polinomios (suma, resta, multiplicación, división, factorización) como también al tutorial y un “acerca de” de la aplicación. ver figura 53 .

Figura 53. Menú principal

Caja de polinomios



3.2 OPERACIÓN SUMA

Interfaz en la que el usuario podrá llevar a cabo la suma de dos polinomios llamados $P(x)$ y $Q(x)$.

- Pantalla inicial de bienvenida a la operación suma mediante una ventana emergente y permite al usuario elegir si desea mirar un ejemplo mediante gifs animados.

Figura 54. Operación suma bienvenida



- Proceder a escribir los polinomios para esto se arrastra las fichas en la caja de polinomios el primer polinomio $P(x)$ se ubicará en la parte izquierda, esto es en los cuadrantes 2 y 3, respetando su respectivo signo, de igual manera el segundo polinomio $Q(x)$ en los cuadrantes de la derecha (1 y 4) para finalmente dar click en el botón “Operar”. La figura 55, ilustra la correcta ubicación de las fichas en el tablero tomando los polinomios a sumar los siguientes:

$$P(x) = 2x^2 - 3$$
$$Q(x) = -x^2 + 1$$

Figura 55. Escribir polinomios suma



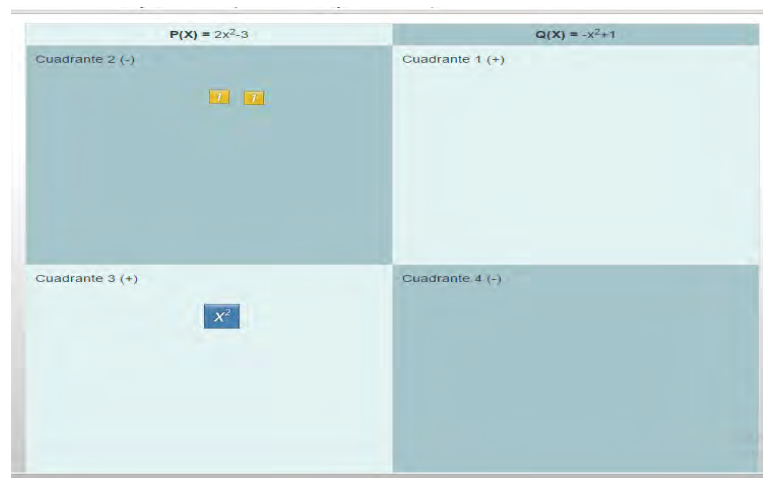
- Una vez ubicado correctamente los dos polinomios que se desea sumar y de dar click en el botón "Operar" la aplicación le guiará al siguiente paso dándole la opción al usuario de mirar un ejemplo a través de gifs animados, como lo indica la figura 56.

Figura 56. Elimina ceros suma



- Eliminar ceros, proceso que el usuario deberá realizar, eliminar un cero es eliminar fichas de igual valor pero que su signo sea contrario o también se puede ver como eliminar aquellas fichas de igual valor pero que esten en cuadrantes de diferente color y finalmente continuar con el siguiente paso dando click en “Verificar Resultado”. Para eliminar fichas se deberán arrastrar hasta el basurero. La figura 57, indica el tablero una vez se ha eliminado las fichas correspondientes a un cero.

Figura 57. Final eliminar ceros suma



- Verificar resultado, indicará al usuario las fichas que tiene en el tablero en una ventana emergente de forma simbólica, como lo muestra la figura 58.

Figura 58. Verificar resultado suma



3.3 OPERACIÓN RESTA

Interfaz en la que el usuario podrá llevar a cabo la resta de dos polinomios llamados $P(x)$ y $Q(x)$.

- **Pantalla inicial:** da la bienvenida a la operación resta y permite al usuario elegir ver un ejemplo del primer paso que se debe hacer en el procedimiento de resolución de un problema a través de gifs animados. Ver figura 59.

Figura 59. Operación resta bienvenida



- **Escribir polinomios:** para esto se arrastra las fichas en la caja de polinomios al igual que en la operación suma y al final dar click en el botón "Operar". En la figura 60, ilustra la correcta ubicación de las fichas en el tablero tomando como base los siguientes polinomios a restar:

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

$$Q(x) = x^2 + x - 1$$

Figura 60. Escribir polinomios resta



- Una vez ubicado correctamente los dos polinomios que se desea restar y de dar click en el botón “Operar” la aplicación le guiará al siguiente paso, permitiendo que el usuario elija la opción de ver un ejemplo sobre lo que debería hacer para el correcto funcionamiento de la Caja de polinomios. Ver figura 61.

Figura 61. Movimiento fichas Q(x) a P(x)



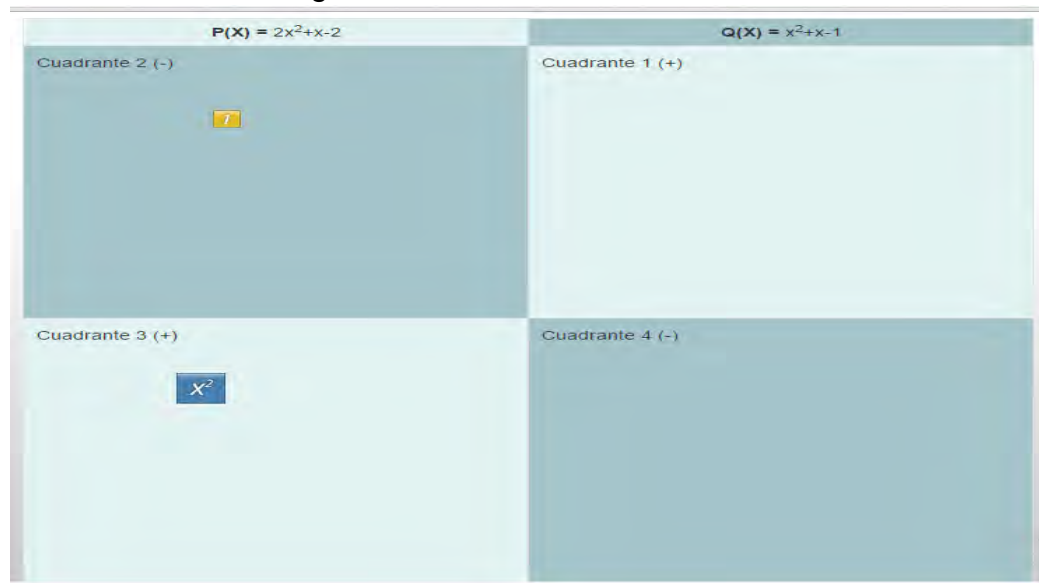
- Mover el polinomio $Q(x)$ hasta $P(x)$ esto es mover las fichas de $Q(x)$ a los cuadrantes de a lado como se muestra en la figura 62.

Figura 62. Termino movimiento fichas $Q(x)$ a $P(x)$



- **Eliminar ceros:** al igual que en la suma es eliminar fichas de igual valor pero con signo contrario, finalmente, el usuario debe continuar con el siguiente paso dando click en "Verificar Resultado". Para eliminar fichas se deberán arrastrar hasta el basurero. La figura 63, indica el tablero una vez se ha eliminado las fichas correspondientes a un cero.

Figura 63. Eliminar ceros resta



- Verificar resultado, indicará al usuario lo que tiene en el tablero de manera simbólica. Ver figura 64.

Figura 64. Verificar resultado resta



3.4 OPERACIÓN MULTIPLICACIÓN

Interfaz en la que el usuario podrá llevar a cabo la multiplicación de dos polinomios llamados P(x) y Q(x).

- **Pantalla inicial:** da la bienvenida a la operación multiplicación y permite que el usuario tenga la opción de elegir ver un ejemplo de cómo debe realizar el primer paso a través de gifs animados. Ver figura 65.

Figura 65. Operación multiplicación bienvenida



- Proceder a escribir los polinomios para esto se arrastra las fichas en la caja de polinomios; al final dar click en el botón “Operar”. En la figura 66, ilustra la correcta ubicación de las fichas en el tablero tomando como base los siguientes polinomios a multiplicar:

$$P(x) = x - 1$$
$$Q(x) = x + 1$$

Figura 66. Escribir polinomios multiplicación



- Al terminar de escribir los polinomios el usuario debe dar click en el botón Operar; la aplicación desplegará una ventana emergente en donde indicara el paso siguiente además de la posibilidad de ver un ejemplo a través de gifs animados. Ver figura 67.

Figura 67. Ventana emergente explicativa multiplicación



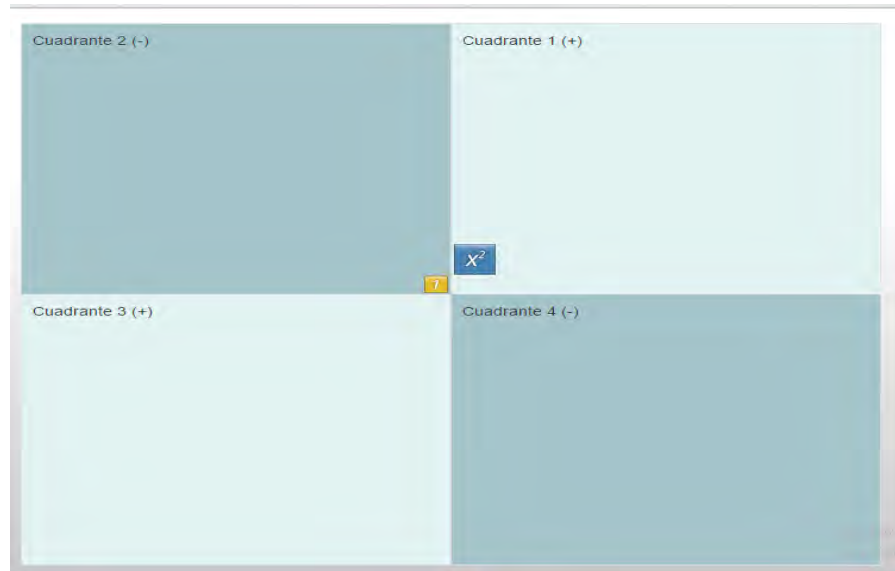
- Completar el rectángulo es el siguiente paso. La figura 68, indicará como queda completo el rectángulo.

Figura 68. Completar rectángulo multiplicación



- Eliminar ceros; eliminar fichas de igual valor con signos contrarios. La figura 69, muestra cómo queda tras eliminar las fichas que conforman ceros.

Figura 69. Eliminar ceros multiplicación



- Verificar resultado, indicará lo que se tiene en el tablero que es lo equivalente al resultado de la multiplicación si el proceso se ha llevado satisfactoriamente. Ver figura 70.

Figura 70. Verificar resultado multiplicación



3.5 OPERACIÓN DIVISIÓN

Interfaz en la que el usuario podrá llevar a cabo la división de dos polinomios llamados $P(x)$ y $Q(x)$.

- **Pantalla inicial:** da la bienvenida a la operación división indica cual sería el primer paso a seguir y permite al usuario ver un ejemplo de cómo se debe hacer este paso. Ver figura 71.

Figura 71. Operación división bienvenida

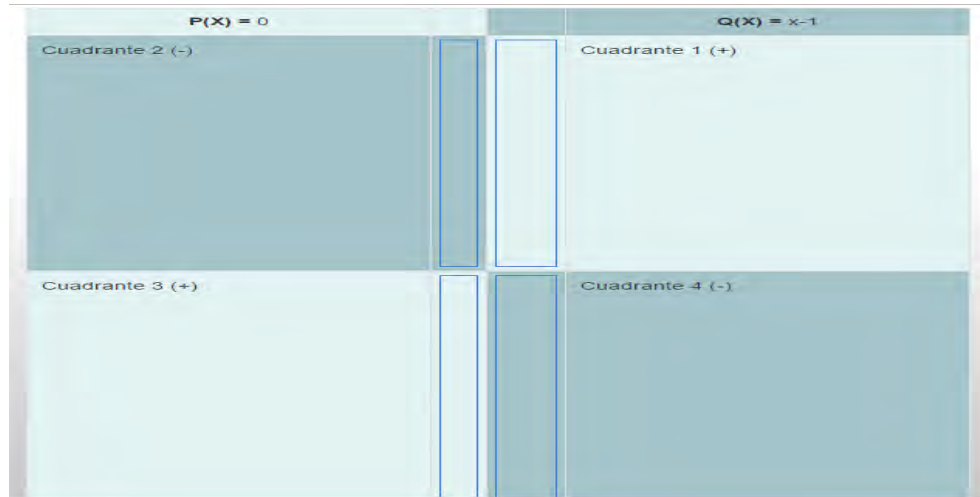


- Para escribir los polinomios a dividir se procede escribiendo el divisor $Q(x)$ utilizando los botones en la parte superior de la interfaz los cuales crearán unas franjas de anchura equivalente al divisor tal como se ilustra en la figura 72, utilizando el siguiente ejemplo:

$$P(x) = -x^2 + x + 1$$

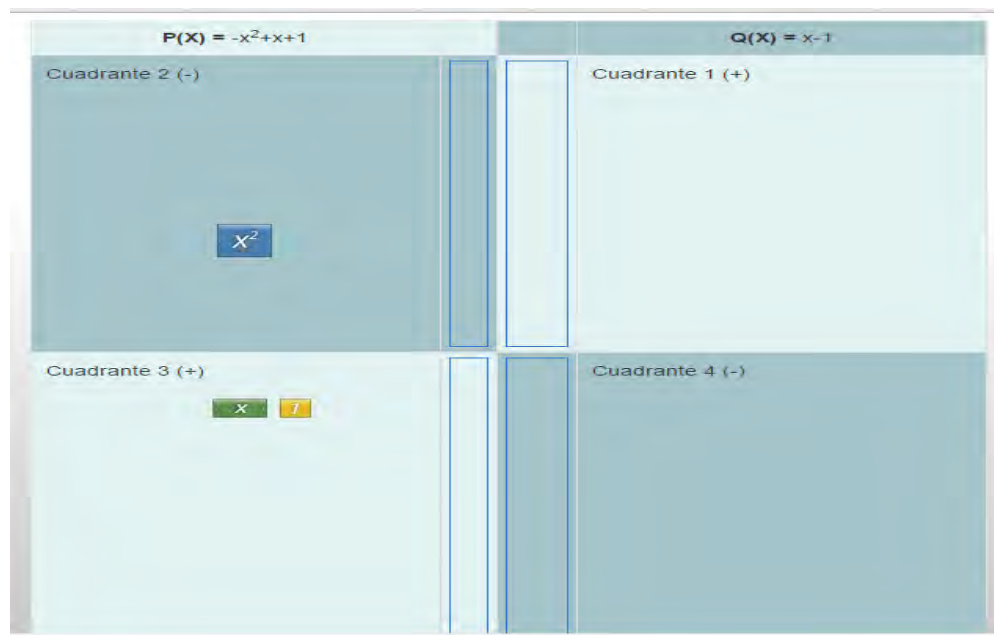
$$Q(x) = x - 1$$

Figura 72. Armar divisor



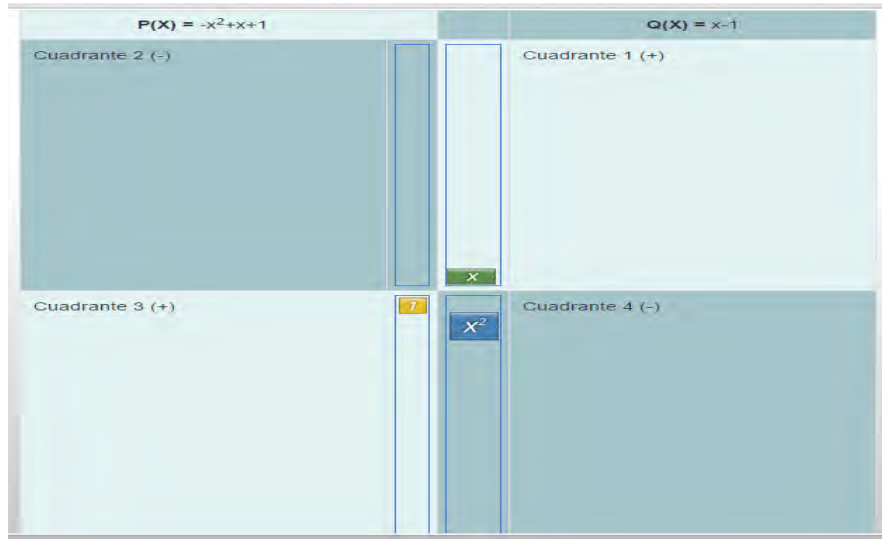
- Para escribir el dividendo se hará con las fichas de la caja de polinomio ubicándolas por fuera de las franjas creadas; al final dar click en el botón “Operar”. La figura 73, ilustra la correcta ubicación de las fichas en el tablero.

Figura 73. Polinomio dividendo



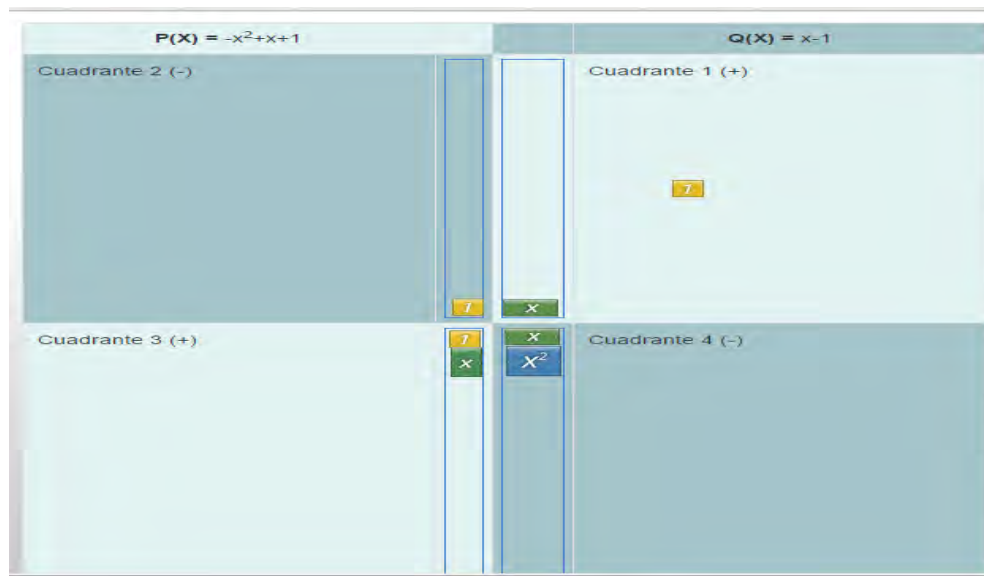
- Ubicar las fichas en las franjas imaginarias respetando el signo de la ficha y el tamaño de las franjas y dar click en el botón Operar. La figura 74, ilustra este procedimiento.

Figura 74. Ubicar fichas franjas imaginarias



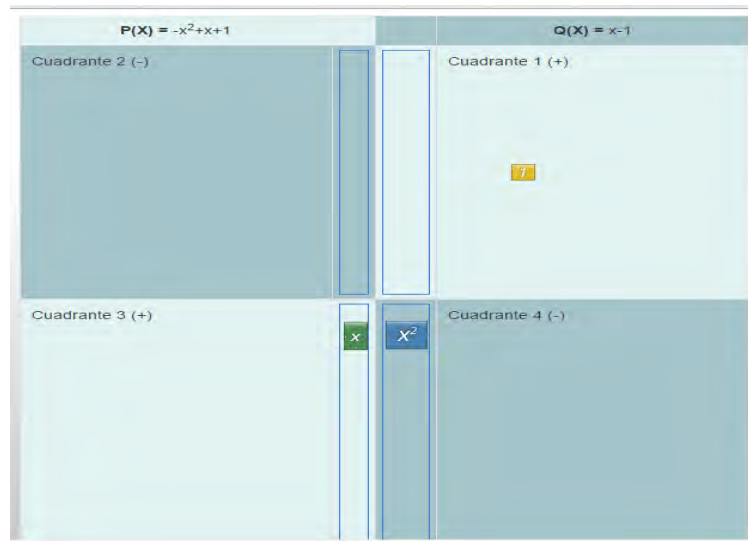
- Completar el rectángulo añadiendo ceros hasta que el residuo sea de grado menor que el divisor $Q(x)$. Ver figura 75.

Figura 75. Completar rectangulo división



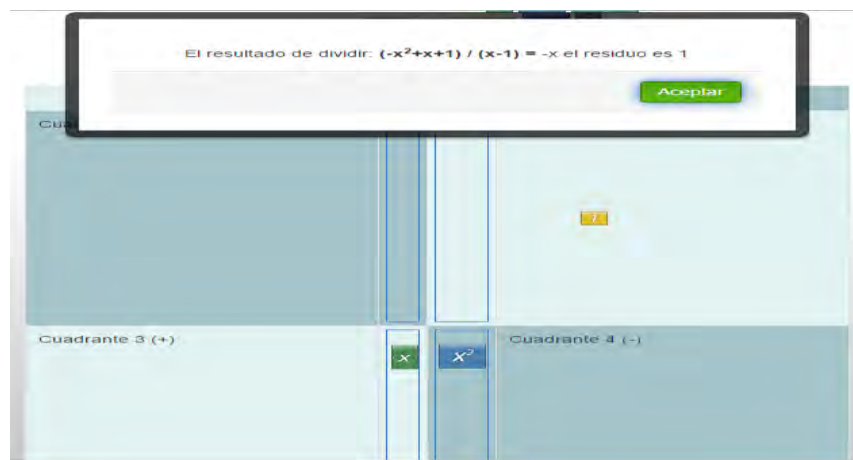
- Eliminar ceros, en la división esto es: eliminar filas que se contrarresten, es decir, que una fila tenga igual valores de fichas que otra pero con sus signos contrarios. Al finalizar dar click en “Verificar resultado” La figura 76, Indica cómo queda el tablero después de eliminar aquellas filas que conformen un cero.

Figura 76. Eliminar ceros división



- Verificar resultado, indicará el resultado de la división si el proceso se ha llevado satisfactoriamente. Indica los polinomios que se dividieron y su respectivo cociente y residuo. Ver figura 77.

Figura 77. Verificar resultado división



3.6 OPERACIÓN FACTORIZACIÓN

Interfaz en la que el usuario podrá llevar a cabo la factorización de un polinomio llamado $P(x)$.

- Pantalla inicial de bienvenida a la operación de factorización y permite que el usuario acceda a un ejemplo de cómo realizar el primer paso. Ver figura 78.

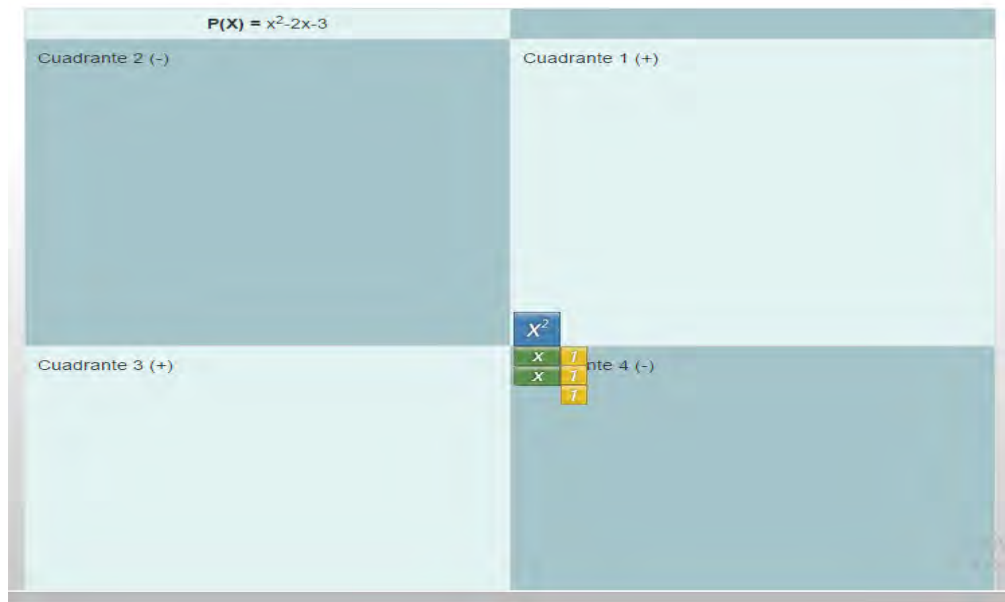
Figura 78. Operación factorización bienvenida



- Proceder a escribir el polinomio para esto se arrastra las fichas en la caja de polinomios, al final dar click en el botón "Operar". En la figura 79, ilustra la correcta ubicación de las fichas en el tablero tomando como base el siguiente polinomio a factorizar:

$$P(x) = x^2 - 2x - 3$$

Figura 79. Escribir polinomio factorización



- Al completar de escribir los polinomios se procede a completar el rectángulo añadiendo ceros. Al finalizar dar click en el botón “Verificar resultado”. La figura 80, indica el tablero una vez se ha completado el rectángulo.

Figura 80. Completar rectángulo factorización

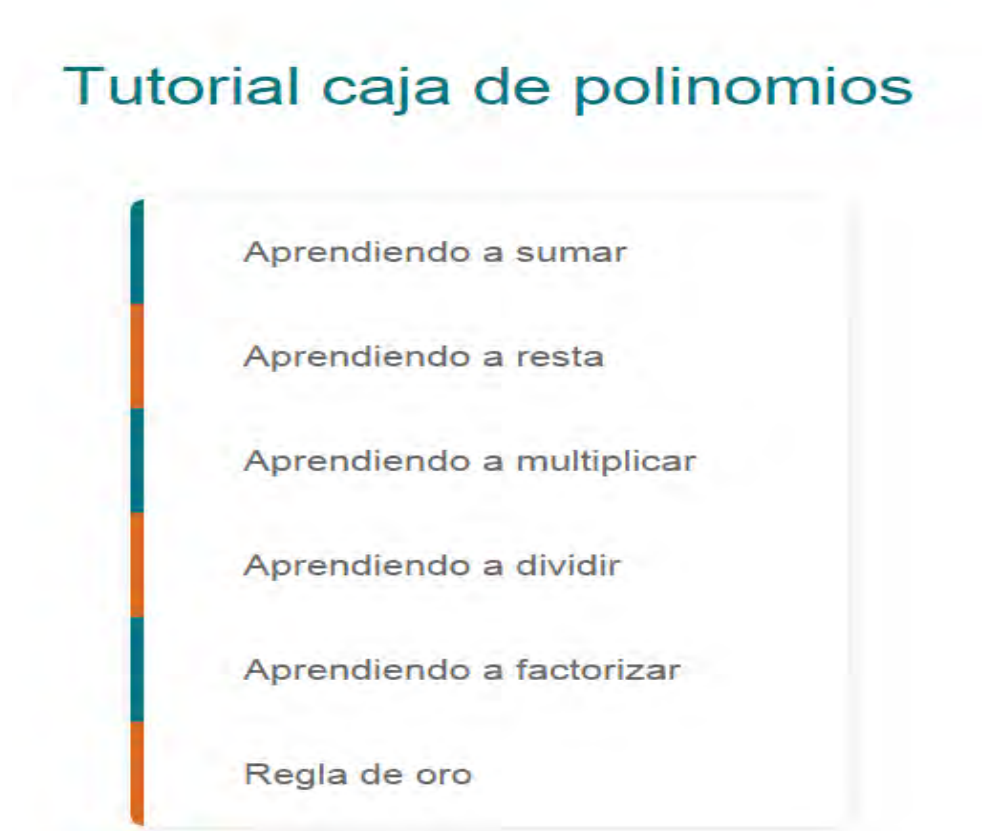


- Verificar el resultado, indicará como leer el resultado de factorizar un polinomio siguiendo la metodología de la caja de polinomios.

3.7 TUTORIAL

Interfaz que ofrece un tutorial al usuario de cómo utilizar la caja de polinomios para cada una de sus operaciones. Ver figura 81.

Figura 81. Interfaz tutorial



3.8 ACERCA DE

Interfaz que muestra información de la App: Nombre de la App, autores, logo, lugar donde se desarrolló, entre otros. Ver figura 82.

Figura 82. Acerca de



4 CONCLUSIONES

- En este trabajo se ha presentado la construcción de un modelo virtual de la caja de polinomios, llevándola a dispositivos móviles y plataforma web, basándose en una arquitectura de desarrollo *cross platform* mediante la utilización del *Framework Apache Cordova* y lenguaje JavaScript con la biblioteca *Jquery*.
- La implementación de ayudas ilustrativas con el uso de *gifs* animados en vez de videos permiten guiar al usuario dentro de cada una de las operaciones logrando un mejor entendimiento de los diferentes procesos que se llevan en la caja de polinomios web móvil, de esta forma se obtienen ventajas como velocidad de carga, consumo de menos espacio en memoria del dispositivo móvil, entre otras, lo que hace eficiente el funcionamiento de la aplicación.
- Se ha utilizado la metodología programación extrema o por sus siglas en ingles *XP* para el desarrollo de cada componente, esto fue en un gran porcentaje acertado ya que permitió obtener un flujo de trabajo constante, mejor código y ganar en calidad de software, al llevar acabo la programación en parejas que propone *XP*.
- El desarrollo de aplicaciones *cross platform* requiere de nuevas tecnologías poco utilizadas en la Universidad de Nariño y con grandes ventajas tales como costo, rapidez de desarrollo, mantenimiento de un único código para todas las plataformas, etc. Lo cual despierta un interés creciente en la comunidad educativa en este tipo de proyectos.
- Con la realización del presente trabajo se logró apoyar el aprendizaje de la operatividad con polinomios construyendo así una herramienta de fácil uso, didáctica y al alcance de los usuarios, basado en el funcionamiento de la caja de polinomio física del profesor Fernando Soto.
- Finalmente el sitio web está desplegado en <http://www.galeras.net/cajadepolinomios> en donde los usuarios tienen la posibilidad de descargar el apk para la aplicación móvil.

5 RECOMENDACIONES

Implementar algún proceso que permita en las operaciones de Multiplicación, División y Factorización el acople de fichas con otras, respetando la “Regla de Oro” de la caja de polinomios, lo cual consiste en que solo se pueden unir fichas aledañas a otras fichas siempre y cuando sus lados coincidan. Esto para facilitarle al usuario el proceso de armar polinomio, así como también dar un efecto vistoso a la aplicación.

Implementar en la multiplicación un proceso que ayude y facilite al usuario a identificar de manera más fácil la lectura de los polinomios $P(x)$ y $Q(x)$ A multiplicar.

Utilizar herramientas de desarrollo y creación de software para juegos que ayuden con el manejo de imágenes, *sprites*, entre otros, muy común mente utilizados para este tipo de aplicaciones.

Utilizar *Apache cordova* como Framework de desarrollo y creación de aplicaciones *cross platform*.

Usar la caja de polinomios web móvil con la guía de un docente que conozca la metodología de la caja de polinomios.

6 BIBLIOGRAFÍA

- Alvarez, M. A. (s.f.). *Manual de jQuery*. Obtenido de desarrolloweb.com:
<http://www.desarrolloweb.com>
- Goodman, D., & Michael, M. (2007). *JavaScript Bible*. United States of America:
Wiley Publishing.
- Guérin, B. A. (2013). *Asp.Net En C# Con Visual Studio 2013. Diseño y Desarrollo de Aplicaciones Web*. Eni ediciones.
- ICFES. (2013). COLOMBIA EN PISA 2012 Informe nacional de resultados Resumen ejecutivo. Bogotá: ISBN de la versión electrónica: 978-958-11-0627-1.
- ICFES. (2014). *Interactivo*. Recuperado el 14 de 02 de 2015, de <http://www.icfes.gov.co/resultados/saber-11-resultados/resultados-agregados-2014-2>
- Lequerica, J. R. (2015). *Desarrollo De Aplicaciones Para Android*. ANAYA Multimedia.
- Matthew, D. (2010). *HTML5*. Amsterdam, Boston, Heidelberg, London, new York: Oxford: Focal Press.
- Moseley , R. (2007). *Desarrollo de aplicaciones Web (Manual Avanzado)*. Anaya multimedia.
- Soto Agreda, Fernando; Naranjo, Claudia Stephania; Lozano, Javier Armando;. (2009). Aprendizaje del Álgebra en grupos con discapacidad auditiva utilizando la Caja de Polinomios . *SIGMA*, IX, 38-60.
- Vicente Carbonell, J. T., Miguel García , C. V., & Daniel Ferri , J. B. (2013). *El gran libro de Android Avanzado*. marcombo ediciones técnicas.