

UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMÁTICA



TESIS DE GRADO

**“MÉTODO CONECTIVO BAJO PRESIÓN V-BLOOM PARA EL APRENDIZAJE
DE PROGRAMACIÓN COMPETITIVA ORIENTADA A PARTICIPANTES DE LA
OLIMPIADA BOLIVIANA DE INFORMÁTICA”**

**PARA OPTAR AL TÍTULO DE LICENCIATURA EN INFORMÁTICA
MENCIÓN: INGENIERÍA DE SISTEMAS INFORMÁTICOS**

POSTULANTE: SAÚL QUISPE VALDEZ

TUTOR METODOLÓGICO: M.Sc. ALDO RAMIRO VALDEZ ALVARADO

ASESOR: Lic. BRÍGIDA CARVAJAL BLANCO

LA PAZ – BOLIVIA

2017



**UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMÁTICA**



LA CARRERA DE INFORMÁTICA DE LA FACULTAD DE CIENCIAS PURAS Y NATURALES PERTENECIENTE A LA UNIVERSIDAD MAYOR DE SAN ANDRÉS AUTORIZA EL USO DE LA INFORMACIÓN CONTENIDA EN ESTE DOCUMENTO SI LOS PROPÓSITOS SON ESTRICTAMENTE ACADÉMICOS.

LICENCIA DE USO

El usuario está autorizado a:

- a) visualizar el documento mediante el uso de un ordenador o dispositivo móvil.
- b) copiar, almacenar o imprimir si ha de ser de uso exclusivamente personal y privado.
- c) copiar textualmente parte(s) de su contenido mencionando la fuente y/o haciendo la referencia correspondiente respetando normas de redacción e investigación.

El usuario no puede publicar, distribuir o realizar emisión o exhibición alguna de este material, sin la autorización correspondiente.

TODOS LOS DERECHOS RESERVADOS. EL USO NO AUTORIZADO DE LOS CONTENIDOS PUBLICADOS EN ESTE SITIO DERIVARA EN EL INICIO DE ACCIONES LEGALES CONTEMPLADOS EN LA LEY DE DERECHOS DE AUTOR.

DEDICATORIA

A mis padres Francisco y Laura por el infinito apoyo.

A mi hermano Roanald por nunca perder su fe en mí.

A mi amada esposa Alison por su espíritu inquebrantable.

A mi hija Wendy eterna fuente de inspiración.

A mi primo hermano Junior siempre impulsándome un round más.

A mi eterno amigo Ericsson Alapati Paz, tu recuerdo en mí vivirá.

A todos mis legendarios estudiantes de la OBI 2011 - 2017

AGRADECIMIENTO

A Dios.

En primer lugar, a Dios por estar conmigo en cada paso de la vida, por fortalecer mi corazón e iluminar mi mente.

A mis padres.

Laura Benigna Valdez Ramos y Francisco Quispe Monzón quienes siempre me brindaron incondicionalmente todo su apoyo sin importar las circunstancias, motivándome a seguir siempre adelante.

A mi esposa.

Alison Yovana Hilari Guarachi por estar a mi lado cada segundo, minuto, cada día, cada año. Mi compañera que supo entender cada aspecto de mí, por dejarme solo cuando lo necesitaba y por hacerme reír cuando me sentía triste, te amo.

A mi hermano.

Ronald Franz Quispe Valdez por estás al pendiente de mi en todo momento y además de ser un gran ejemplo.

A mi hija.

Wendy Andrea Quispe Hilari, luz de mi vida, mi gran felicidad y mayor alegría.

A mi primo hermano.

Junior Gabriel Valdez Aliendre el guerrero, por su aliento y apoyo quien siempre estuvo en todas mis fortunas y peores desventuras.

A la universidad.

Un agradecimiento a mi segunda casa, la Universidad Mayor de San Andrés por acogerme en sus aulas y en especial a la carrera de Informática que me dio la oportunidad de formar parte de ella.

A mi tutor y asesora.

A mi tutor metodológico M.Sc. Aldo Ramiro Valdez Alvarado y asesora Lic. Brígida Carvajal Blanco por la magnífica guía, compartiendo su sabiduría con disposición y gran profesionalismo.

Gracias totales

ÍNDICE

CAPÍTULO I MARCO INTRODUCTORIO.....	1
1.1. INTRODUCCIÓN.....	1
1.2. ANTECEDENTES	3
1.3. EL PROBLEMA.....	5
1.3.1. PLANTEAMIENTO DEL PROBLEMA.....	5
1.3.2. FORMULACIÓN DEL PROBLEMA	7
1.4. OBJETIVOS	7
1.4.1. OBJETIVO GENERAL	7
1.4.2. OBJETIVOS ESPECÍFICOS	8
1.5. HIPÓTESIS	8
1.6. JUSTIFICACIÓN	8
1.6.1. JUSTIFICACIÓN SOCIAL	8
1.6.2. JUSTIFICACIÓN TEÓRICA Y CIENTÍFICA.	8
1.7. ALCANCES Y LÍMITES	9
1.7.1. ALCANCES	9
1.7.2. LÍMITES	9
CAPÍTULO II MARCO TEÓRICO.....	10
2.1. INTRODUCCIÓN.....	10
2.2. MÉTODO	12
2.2.1. NIVELES DE CONCEPTUALIZACIÓN DE UN MÉTODO	12
2.2.2. FACTORES QUE INFLUYEN SOBRE UN MÉTODO.....	14
2.3. ASPECTOS EDUCATIVOS RELEVANTES.....	15
2.3.1. EL CONO DEL APRENDIZAJE DE EDGAR DALE.....	15
2.3.2. LA PIRÁMIDE DE MILLER	16
2.3.3. EL DIAGRAMA V DE GOWIN	17
2.3.4. LA TAXONOMÍA DE BLOOM	18
2.3.5. UNA TEORÍA DE APRENDIZAJE PARA LA ERA DIGITAL, EL CONECTIVISMO	19
2.3.6. ENSEÑANZA Y APRENDIZAJE BAJO PRESIÓN	21
2.4. PENSAMIENTO COMPUTACIONAL	21
2.4.1. ENSEÑAR PENSAMIENTO COMPUTACIONAL “SIN PROGRAMAR” ...	23
2.4.2. PENSAMIENTO COMPUTACIONAL “A TRAVÉS DE LA PROGRAMACIÓN”	24
2.5. PROGRAMACIÓN.....	27
2.5.1. LA PROGRAMACIÓN COMO UN ARTE	28
2.5.2. MÉTODOS DE ANÁLISIS DE PROBLEMAS	29
2.5.3. LÓGICA DE PROGRAMACIÓN	36
2.5.4. ALGORITMOS	38
2.5.5. EL PARADIGMA ESTRUCTURADO.....	40

2.5.6. REPRESENTACIÓN DE LOS ALGORITMOS MEDIANTE DIAGRAMAS DE FLUJO	43
2.5.7. PRUEBAS DE ESCRITORIO PARA LA VERIFICACIÓN DE RESULTADOS	44
2.5.8. CODIFICACIÓN	46
2.6. PROGRAMACIÓN COMPETITIVA.....	47
2.6.1. COMPETENCIAS DE PROGRAMACIÓN.....	48
2.6.2. JUECES EN LÍNEA.....	51
2.6.3. SISTEMAS DE ENTRENAMIENTO PARA PROGRAMACIÓN COMPETITIVA	53
2.7. CURRÍCULOS EDUCATIVOS PARA LA ENSEÑANZA-APRENDIZAJE DE PROGRAMACIÓN.....	54
2.8. ALGORITMOS Y PROGRAMACIÓN EN LA EDUCACIÓN ESCOLAR	58
CAPÍTULO III MÉTODO CONECTIVO BAJO PRESIÓN V-BLOOM.....	60
3.1. EL MÉTODO CONECTIVO BAJO PRESIÓN V-BLOOM.....	60
3.2. CONCEPTUALIZACIÓN DEL MÉTODO	60
3.2.1. ENFOQUE	61
3.2.2. DISEÑO	61
3.2.3. PROCEDIMIENTOS	64
CAPITULO IV EVALUACIÓN DEL MÉTODO	71
4.1. INVESTIGACIÓN	71
4.1.1. ENFOQUE DE LA INVESTIGACIÓN.....	71
4.1.2. TIPO DE INVESTIGACIÓN.....	71
4.1.3. OPERACIONALIZACIÓN DE VARIABLES.....	72
4.1.4. POBLACIÓN	72
4.1.5. MUESTRA.....	72
4.1.6. TÉCNICAS PARA EL ANÁLISIS E INTERPRETACIÓN DE DATOS	73
4.2. VALIDACIÓN DEL MÉTODO	73
4.2.1. BASES DE LA PROPUESTA.....	73
4.2.2. DESARROLLO DE LA PROPUESTA.....	74
CAPITULO V CONCLUSIONES Y RECOMENDACIONES	89
5.1. CONCLUSIONES.....	89
5.2. RECOMENDACIONES	90
BIBLIOGRAFÍA	91

ÍNDICE DE TABLAS

Tabla 1.1 Trabajos de investigación desarrollados en Latinoamérica	4
Tabla 1.2 Trabajos de investigación desarrollados en Bolivia	5
Tabla 1.3 Fechas para la toma de pruebas de la OCEPB 2017	7
Tabla 2.1 Comparativa de las distintas teorías del aprendizaje	20
Tabla 2.2 Comparativa y definiciones de pensamiento computacional	22
Tabla 2.3 Habilidades que define el Pensamiento Computacional.	23
Tabla 2.4 Cuadro de competencias de programación	48
Tabla 2.5 Cuadro comparativo programación en niños y jóvenes	55
Tabla 3.1 Contenidos temáticos a abordar.....	62
Tabla 3.2 Plantilla para la gestión de contenidos	65
Tabla 3.3 Cuadro descriptivo de los niveles del método V-Bloom.....	70
Tabla 4.1 Tabla para la operacionalización de variables.....	72
Tabla 4.2 Cantidad de inscritos de la primera etapa.....	74
Tabla 4.3 Prueba de normalidad Kolmogorov-Smirnov etapa 2:.....	75
Tabla 4.4 Prueba de U de Mann-Whitney etapa 2.....	76
Tabla 4.5 Numero de aprobados post-test segunda etapa.....	78
Tabla 4.6 Prueba de normalidad Kolmogorov-Smirnov etapa 3:.....	79
Tabla 4.7 Prueba de U de Mann-Whitney etapa 3.....	79
Tabla 4.8 Numero de aprobados post-test segunda fase.....	82
Tabla 4.9 Significado de siglas del reporte del Juez Patito	83
Tabla 4.10 Resumen de resultados Juez Virtual Patito Informática UMSA	84
Tabla 4.11 Prueba de normalidad Kolmogorov-Smirnov la etapa 4:	85
Tabla 4.12 Prueba de U de Mann-Whitney etapa 4.....	86
Tabla 4.13 Numero de aprobados post-test segunda fase.....	88

ÍNDICE DE FIGURAS

Figura 2.1 Conceptualización del método	13
Figura 2.2 Factores que influyen sobre un método	15
Figura 2.3 Pirámide del aprendizaje	15
Figura 2.4 Guía para la evaluación de competencias en medicina	17
Figura 2.5 Esquema simplificado del Diagrama	18
Figura 2.6 Taxonomía de Bloom.....	19
Figura 2.7 Mapa Mental de la Lógica de Programación.	38
Figura 2.8 Estructuras de control para el paradigma estructurado	42
Figura 2.9 Diagrama de Flujo para el algoritmo área de un rectángulo.	44
Figura 2.10 Prueba de escritorio.....	46
Figura 2.11 Diagrama de Flujo para la solución de problemas de programación	50
Figura 3.1 Cronograma de actividades.	63
Figura 3.2 Descripción de niveles del método conectivo bajo presión V-Bloom	70
Figura 4.1 Diagrama de cajas etapa 2.....	77
Figura 4.2 Diagrama de cajas etapa 3.....	81
Figura 4.3 Premiación 3 etapa (departamental).....	82
Figura 4.4 Reporte de desempeño del mejor estudiante del grupo experimental.....	83
Figura 4.5 Diagrama de cajas etapa 4.....	87
Figura 4.6 Premiación 4 etapa (Nacional).....	88

RESUMEN

Esta investigación trabaja sobre el tema de tecnología informática en el ámbito educativo. Inicialmente se presenta un sustento teórico para luego diseñar el denominado método conectivo bajo presión V-Bloom, orientado a la enseñanza aprendizaje de programación competitiva para estudiantes de nivel escolar con edades comprendidas 11 y 15 años. En una segunda instancia se prueba la validez del mismo en 3 evaluaciones correspondientes a las 4 etapas de la 7ma Olimpiada Científica Estudiantil Plurinacional Boliviana 2017 en el área de Informática (Nivel 2). Finalmente, se proporciona una interpretación de los resultados obtenidos y se presentan las conclusiones del trabajo.

Palabras clave

Olimpiada Boliviana de Informática, OBI, Olimpiada Científica Estudiantil Plurinacional Boliviana, conectivismo, programación competitiva, método V-Bloom.

ABSTRACT

This research works on the subject of computer technology in the educational field. Initially a theoretical support is presented to design the so-called connective method under pressure V-Bloom, oriented to the teaching of competitive programming for school-age students aged 11 and 15 years. In a second instance, the validity of the same is proved in 3 evaluations corresponding to the 4 stages of the 7th Bolivian Plurinational Student Olympiad 2017 in the area of Information Technology (Level 2). Finally, an interpretation of the obtained results is provided and the conclusions of the work are presented.

Keywords

Bolivian Informatics Olympiad, OBI, Bolivian Plurinational Student Science Olympiad, connectivism, competitive programming, V-Bloom method.

CAPÍTULO I

MARCO INTRODUCTORIO

1.1. INTRODUCCIÓN

El uso de computadoras en la educación para desarrollar habilidades en los niños, como la abstracción y la resolución de problemas, tiene ya varias décadas. Uno de sus impulsores, Seymour Papert, discípulo directo de Jean Piaget, creó a fines de los años 60, en el Instituto Tecnológico de Massachusetts (MIT), el lenguaje de programación Logo¹, que, con impulso local, llegó a usarse en varias escuelas del país en la década de los 80. En 1972, Alan Kay presentó el diseño de la Dynabook, una precursora de las actuales tabletas, cuyo uso estaba focalizado a la educación. Pero luego, cuando comenzó la revolución de las computadoras personales, la enseñanza de la computación en las escuelas se orientó hacia el uso de aplicaciones como hojas de cálculo y procesadores de texto, tendencia que puede verse en la mayoría de las escuelas que hoy enseñan informática. Esta visión eminentemente hizo perder el foco original, que ofrecía un enorme potencial de programación en la escuela (Ceria, 2014).

En la actualidad, los Estados Unidos impulsa través de la asociación code.org y su evento la Hora del Código (Hour of Code), que congrega a más de 100 millones de estudiantes de todo el mundo, para impulsar las ciencias de la computación a través de la programación (code.org, 2017). Por su parte emprendedores tecnológicos encabezados por figuras emblemáticas como Steve Jobs y Mark Zuckerberg, han abierto paso a un ecosistema que privilegia estas habilidades. Así, cada vez más jóvenes terminado su estudio escolar ven en la informática un área de especialización (Garzón, 2015).

Bajo estas circunstancias la revolución de las llamadas Tecnologías de la Información y Comunicación (TIC), ha despertado nuevamente aquellas ideas fundamentales de aprovechar el desarrollo de habilidades que puede generar la programación de

¹ **Logo.** Fue un lenguaje de programación creado con la finalidad de que los niños pequeños aprendieran matemáticas de forma fácil y sencilla. Su protagonista era una pequeña tortuga que el niño/a podía manejar utilizando un vocabulario natural para dibujar todo tipo de figuras geométricas.

computadoras, ahora bajo el título de pensamiento computacional,² acuñado por la investigadora Jeannette Wing.

Tanto el Reino Unido como los Estados Unidos, los grandes precursores de la computación, están promoviendo iniciativas para incorporar las ciencias de la computación a la educación formal. En Estados Unidos se han planteado el objetivo de formar 10.000 docentes para enseñar ciencias de la computación en escuelas secundarias antes de 2016. Otros países como Finlandia, China, Nueva Zelanda, Estonia e Israel han seguido o siguen caminos similares. Considerando este nuevo escenario, varios gobiernos a nivel mundial como Bélgica, Irlanda, Malta, Polonia y República Checa entre otros, mencionan al pensamiento computacional como una competencia clave a adquirir mediante la integración de la programación como materia oficial de sus sistemas educativos (INTEF, 2015).

En síntesis, como señala Steve Jobs: "Todos en este país deberían aprender a programar computadoras, porque te enseña a pensar" (Garzón, 2015, pág. 6).

En Bolivia, el Ministerio de Educación, a través del Viceministerio de Ciencia y Tecnología, en coordinación con el Viceministerio de Educación Regular y las Universidades Bolivianas, organizan la Olimpiada Científica Estudiantil Plurinacional Boliviana, actividad que consecuentemente da espacio a la Olimpiada Boliviana de Informática (OBI), organizada gracias al Comité Nacional de la ACM-ICPC³ Bolivia, los cuales convocan a estudiantes de Unidades Educativas Públicas, Privadas y de Convenio a participar de la misma y seleccionar representantes para las competencias internacionales.

² **Pensamiento Computacional.** Se define como el proceso por el cual un individuo, a través de habilidades propias de la computación y del pensamiento crítico, pensamiento lateral y otros más logra hacer frente a problemas de distinta índole.

³ ACM-ICPC Competición Internacional Universitaria ACM de Programación (en inglés ACM *International Collegiate Programming Contest*, abreviado ACM-ICPC o simplemente ICPC) es una competición anual de programación y algorítmica entre universidades de todo el mundo patrocinada por IBM.

Esta Olimpiada forma parte del Sistema Internacional de Olimpiadas Científicas con su evento denominado Olimpiada Internacional de Informática (IOI⁴) (Educabolivia, 2016).

A razón de todo lo mencionado, el presente trabajo de investigación, expondrá las características de un método de enseñanza aprendizaje desarrollado para mejorar habilidades, aptitudes y destrezas, siendo aplicado a estudiantes de nivel de escolaridad primaria y secundaria, quienes participaron de la 7ma Olimpiada Científica Estudiantil Plurinacional Boliviana (7ma OCEPB), los cuales deberán consolidar en un tiempo determinado las competencias necesarias que se requieren, para afrontar diversos problemas dentro el mundo de la programación competitiva⁵.

1.2. ANTECEDENTES

A nivel internacional, se llevaron a cabo numerosas investigaciones relacionadas entorno al proceso de enseñanza y aprendizaje de la programación, tanto en bachillerato y a nivel universitario para despertar el interés y aumentar el grado de motivación en los alumnos hacia la materia en cuestión (Mercado, Andrade, & Reynoso, 2008); (Arévalo & Solano, 2013); (Brito & Sá-Soares, 2010); (Malliarakis, Satratzemi, & Xinogalos, 2014).

Por otro lado, en determinados países se optó en realizar estudios utilizando lenguajes de programación educativos en nivel bachillerato y/o en etapa escolar, esto con la finalidad de que los estudiantes tengan una agradable transición a un lenguaje de programación orientado a objetos como lo es C++ o Java, obteniendo resultados positivos en estudiantes que no habían tenido experiencia alguna en materias de programación, despertando así su interés en ellas (Kereki, 2008); (Malan & Leitner, 2007).

⁴ IOI Olimpiada Internacional de Informática es una más de las Olimpiadas Internacionales de Ciencias, que tiene como objetivo potenciar el aprendizaje de la informática en alumnos de secundaria y preparatoria (nivel medio y medio superior). Los problemas de esta competición están orientados a los algoritmos.

⁵ Programación Competitiva. Deporte mental en el cual los participantes resuelven un conjunto de problemas bien especificados a través de programas de computador.

Según la bibliografía, existen numerosos trabajos internacionales que abordan el tema de enseñanza aprendizaje citando diversos métodos. Sin embargo, en nuestro país, quedan pendientes investigaciones vinculadas con la temática expuesta anteriormente.

A continuación, se menciona algunos trabajos fundamentales descritos en la Tabla 1.1 y Tabla 1.2.

Tabla 1.1
Trabajos de investigación desarrollados en Latinoamérica

TÍTULO	AUTOR	AÑO
Modelo Tecno-Pedagógico basado en Ludificación y Programación Competitiva para el diseño de cursos de programación.	Andrés Felipe Pineda Corcho	2014
Universidad Nacional de Colombia, Departamento de Ciencias de la computación, Colombia		
Motivación en el aprendizaje de la programación a nivel bachillerato utilizando un lenguaje de programación educativo.	Lic. Neira Yashira García Ortega	2016
Universidad Autónoma de Aguascalientes, Carrera de Informática y Tecnologías Computacionales, México		
Codigoalfabetización y pensamiento computacional en educación primaria y secundaria: validación de un instrumento y evaluación de programas.	MSc. Marcos Román González	2016
Escuela Internacional de Doctorado EIDUNED, Programa de doctorado en educación, España		

Nota. Fuente: Elaboración propia

Tabla 1.2
Trabajos de investigación desarrollados en Bolivia

TÍTULO	AUTOR	AÑO
Tutor Inteligente para la Olimpiada Científica Estudiantil Plurinacional Boliviana en el área de informática bajo un modelo de aprendizaje.	Yvan Mayta Quispe	2013
Universidad Mayor de San Andrés, Carrera de Informática, Bolivia		
Tutor Inteligente para desarrollar la lógica de programación en estudiantes de secundaria.	Kristian Rudy Ortuño Villanueva	2014
Universidad Mayor de San Andrés, Carrera de Informática, Bolivia		

Nota. Fuente: Elaboración propia 2017

En los casos de investigación vigentes desarrollados en Bolivia a la fecha, se puso hincapié al desarrollo de tutores inteligentes.

1.3. EL PROBLEMA

1.3.1. PLANTEAMIENTO DEL PROBLEMA

El proceso de enseñanza y aprendizaje de la programación es uno de los grandes desafíos de la computación (Barker, Coope, McGettrick, Thatcher, & Topi, 2015). Es así que muchos educadores consideran que la enseñanza de la programación representa un paradigma⁶ nuevo de introducción de la computadora en la escuela, lo cual lleva a pensar nuevas formas de introducir las tecnologías en las aulas, más allá del uso de las TIC. En este contexto, el trabajo conjunto entre las Ciencias de la Computación y Ciencias de la Educación, están en un momento de sinergia (Wolovick & Martínez, 2016).

En Bolivia, la OBI trabaja bajo los argumentos anteriormente citados, aplica sobre sus Niveles 2 y 3, la modalidad de programación competitiva. Planteando como objetivos: contribuir al mejoramiento de la enseñanza y del aprendizaje, además de identificar y

⁶ **Paradigma.** Un paradigma define un conjunto de reglas, patrones y estilos de programación que son usados por un grupo de lenguajes de programación

preparar a jóvenes talentosos para que representen a sus departamentos según esto según la convocatoria de la 7ma OCEPB.

Para entrar más a detalle, son los profesores de las distintas unidades educativas quienes se encargan de seleccionar a sus mejores estudiantes los cuales deben cursar los niveles 1ro de secundaria y 6to secundaria, posteriormente se da paso a la inscripción de los mismos como parte de la 1^{ra} etapa (Inscripción). Luego de algún tiempo los inscritos proceden a ser evaluados en una 2^{da} etapa (Distrital), rinden una prueba virtual no presencial que consta de 3 problemas con una calificación máxima de 300 puntos, donde la evaluación es realizada por un juez virtual bajo la corrección y aprobación de los distintos encargados departamentales, en esta instancia son seleccionados aquellos estudiantes que alcanzan una calificación mínima de 1 punto o código relativamente efectivo. Después de un determinado lapso de tiempo y con el grupo selecto del proceso anterior, se da paso a la 3^{ra} etapa (Departamental) que a diferencia de la etapa anterior, la evaluación virtual es presencial, se presentan 3 problemas con una calificación máxima de 300 puntos, donde la evaluación es realizada por un juez virtual bajo la corrección y aprobación de los distintos encargados departamentales, el criterio de selección exige bajo norma elegir las 7 mejores calificaciones por departamento, acreditando a las mejores calificaciones con las medallas de oro, plata y bronce de connotación departamental. Finalmente, y en una fecha determinada, se procede con la denominada 4^{ta} etapa (Nacional) con una selección de 7 estudiantes por cada uno de los 9 departamentos de Bolivia, esta prueba final presencial bajo evaluación de un juez virtual y jueces físicos seleccionados por el comité nacional de OBI, consta de 4 preguntas, con calificación máxima de 400 puntos, en este punto las calificaciones más altas permiten a los estudiantes ser acreedores a las medallas de Oro Plata o Bronce de connotación nacional. Otros aspectos a considerar son las fechas de evaluación (Tabla 1.3) y los contenidos (ver Tabla 3.1) establecidos por parte de la OBI.

Queda asumir que todos los profesores participantes emplearán diversos métodos para este fin, con una única y complicada finalidad, conseguir que se adquieran los conocimientos,

habilidades y destrezas fundamentales para obtener un buen desempeño en las distintas etapas de selección, además de tomar en cuenta un factor muy importante, el tiempo de capacitación.

Tabla 1.3
Fechas para la toma de pruebas de la OCEPB 2017

Etapas	Ámbito	Modalidad	Inscripción	Prueba
Primera	Unidad Educativa	Presencial	Automática	Previa valoración interna en cada Unidad Educativa hasta el 30 de junio
Segunda	Distritos Educativos	Presencial	Automática	30 de julio.
Tercera	Departamental	Presencial	Automática	27 de agosto
Cuarta	Nacional	Presencial	Automática	14 de octubre

Nota. Fuente: Convocatoria de la 7ma OCEPB

1.3.2. FORMULACIÓN DEL PROBLEMA

Con todo lo expuesto anteriormente se puede considerar una problemática y plantear la siguiente pregunta de investigación, la cual se buscó responder durante el desarrollo de este trabajo:

¿Cómo se puede enseñar Programación Competitiva a participantes de la Olimpiada Boliviana de Informática con edades comprendidas entre 11 y 15 años?

1.4. OBJETIVOS

1.4.1. OBJETIVO GENERAL

- Plantear un método que mejore los resultados obtenidos en las distintas etapas de la Olimpiada Boliviana de Informática 2017.

1.4.2. OBJETIVOS ESPECÍFICOS

- Diseñar y aplicar un experimento que pruebe la validez del método, interpretando sus resultados.
- Desarrollar una plantilla para un plan de clase que se ajuste al método y sus destinos niveles.
- Cuantificar la evolución previa a la última fase de todos de los estudiantes bajo el método V-Bloom.
- Señalar los recursos basados en el conectivismo que emplea el método V-Bloom.

1.5. HIPÓTESIS

El Método conectivo bajo presión V-Bloom influye positivamente en las calificaciones obtenidas en cada una de las fases de selección de la Olimpiada Boliviana de Informática 2017.

1.6. JUSTIFICACIÓN

1.6.1. JUSTIFICACIÓN SOCIAL

Las condiciones en las cuales se desarrolla la OBI, en cierto punto son desequilibrantes, debido a la ausencia de recursos metódicos para afrontar dicha competencia. El presente estudio representa un recurso para la enseñanza y aprendizaje de la programación competitiva pensado para estudiantes de los niveles primaria y secundaria.

1.6.2. JUSTIFICACIÓN TEÓRICA Y CIENTÍFICA.

La presente investigación desarrolla un estudio significativo en el campo de la informática, puesto que la enseñanza de programación en las escuelas continúa siendo un importante tema de estudio que genera diversos espacios de debate en el mundo.

1.7. ALCANCES Y LÍMITES

1.7.1. ALCANCES

La investigación presenta los siguientes alcances:

- La investigación está ajustada a las condiciones temporales y de contenido establecidas por la OBI.
- La experimentación se desarrolló en torno a estudiantes de etapa escolar con edades comprendidas entre los 11 y 15 años de edad.
- El tiempo de aplicación del método demoró 144 horas en aula.

1.7.2. LÍMITES

- El estudio contempló la aplicación del método exclusivamente en el área de informática, no habiéndose realizado experimentación en otras áreas.
- La investigación se limita a desarrollarse como un aspecto teórico para posteriormente convertirse en un sistema tutor.

CAPÍTULO II MARCO TEÓRICO

2.1. INTRODUCCIÓN

El mundo está viviendo un momento donde la educación está altamente comprometida con las competencias digitales⁷ de sus ciudadanos, es así que las necesidades propias de una sociedad post-industrial⁸, se convierten en un derecho que debe protegerse y fomentarse con el fin de desarrollar aquellas capacidades cognitivas y sociales necesarias para interactuar, de una manera crítica, dentro un contexto digital cambiante. La visión compartida por los contextos político-ideológico, económico-empresarial, científico-técnico y socioeducativo, está relacionada con la necesidad de una reforma profunda de los sistemas educativos, que se consideran están estancados en un modelo de sociedad industrial que no cumple con las demandas educativas, laborales y económicas de la actual sociedad. Es así que las soluciones para un sistema educativo que no satisface las necesidades de una sociedad digital, pasa por incorporar nuevos métodos en el proceso de enseñanza-aprendizaje. (Berrocoso, Sánchez, & Arroyo, 2015).

De esta forma la investigación permite ofrecer evidencias sobre los procesos de enseñanza-aprendizaje y, desde la realidad de cada contexto socio-educativo, ofrece conocimientos para desarrollar propuestas curriculares que generen aprendizajes profundos (comprensión vs. memorización). Las innovadoras competencias de programación en la escuela, forman parte de una nueva ecología del aprendizaje que tiene como particularidad romper los límites espacio-temporales del proceso educativo gracias a las tecnologías digitales. Simultáneamente, se alteran los tradicionales roles educador-educando, puesto que, en el contexto de una enseñanza y aprendizaje en red, no se define de manera inamovible quién

⁷ **Competencias digitales.** (en inglés, e-skills) son un conjunto de conocimientos, capacidades, destrezas y habilidades, en conjunción con valores y actitudes, para la utilización estratégica de la información, y para alcanzar objetivos de conocimiento tácito y explícito, en contextos y con herramientas propias de las tecnologías digitales.

⁸ **Post-industrial.** Es un concepto propuesto por varios teóricos de la economía y la sociología para describir el estado de un sistema político, económico y social que corresponde a un estadio de desarrollo posterior al proceso de industrialización clásico de la Revolución Industrial

enseña y quién aprende. Desde este enfoque se entiende que los sistemas educativos no necesitan más contenidos sino, esencialmente, entornos flexibles que permitan desarrollar las capacidades de autoaprendizaje, creatividad, autonomía, iniciativa y expresión multilingüaje (Záhore, Hašková, & Munk, 2014).

Bajo estas circunstancias en los últimos años, resurgió un movimiento educativo a nivel internacional relacionado con la introducción del pensamiento computacional, la programación informática y la robótica en las escuelas.

Según Berrocoso, Sánchez y Arroyo (2015), indican que historia de la Tecnología Educativa se describe cómo los primeros pasos de la informática en las aulas, en la década de los ochenta del siglo XX, estuvieron ligados al aprendizaje de la geometría mediante el uso de un lenguaje de programación denominado Logo y su famosa tortuga, dentro de un proyecto educativo conformado por las ideas de Dewey, Piaget o Vygotsky, y materializado por Seymour Papert inventor del lenguaje de programación Logo. Era, por consiguiente, un entorno favorable a la enseñanza y aprendizaje de lenguajes de programación; pero, la aparición de sistemas operativos visuales e intuitivos desvalorizó en la actividad docente el uso de lenguajes de programación frente al aprendizaje, como usuario, de diferentes herramientas informáticas. Con el auge de los movimientos DIY (*Do It Yourself*, Hágalo usted mismo) y DIT (*Do It Together*, Háganlo todos), en este punto se consolidan capacidades donde las personas crean sus propios objetos, con frecuencia usando tecnología impulsando el denominado pensamiento computacional para realizar programación, además de conseguir el trabajo en equipo, solidaridad, colaboración, respeto y aceptación.

Como una respuesta a esta demanda varios gobiernos, empresas y organizaciones sin ánimo de lucro o instituciones educativas a nivel mundial, han desarrollado proyectos y adoptada disposición relacionada con el fomento de la codificación en la educación.

2.2. MÉTODO

Desde el ángulo de la didáctica general, la palabra método, encierra el concepto de una dirección hacia el logro de un propósito, un camino a recorrer, aunque es claro que ha de entenderse que no se trata de un camino cualquiera, sino del mejor, del más razonable, del que más garantice la consecución de la finalidad propuesta. El método implica un proceso de ordenamiento, la dirección del pensamiento y de la acción para lograr algo previamente determinado. Significa entonces, que un buen método será aquel que garantice un máximo aprovechamiento o rendimiento de la enseñanza aprendizaje en menor tiempo. Un método integra un conjunto de principios, una descripción de la praxis, actividades y normalmente el sistema de evaluación. La elección del método o métodos de enseñanza que se utilizarán, dependerá gran parte de la información o habilidad que se está enseñando, en los métodos educativos se trabaja las habilidades cognitivas, las cuales serán afectadas por el contenido de aprendizaje y el nivel de los estudiantes. (Martínez, 2004).

2.2.1. NIVELES DE CONCEPTUALIZACIÓN DE UN MÉTODO

El concepto de método en la enseñanza es la noción de un conjunto sistemático de prácticas de enseñanza basadas en una teoría particular del aprendizaje. Hay tres niveles de conceptualización y organización identificados por (Richards & Rodgers, 1963), los cuales son: enfoque, diseño y procedimientos, conceptos que describen cualquier método completo (ver Figura 2.1).

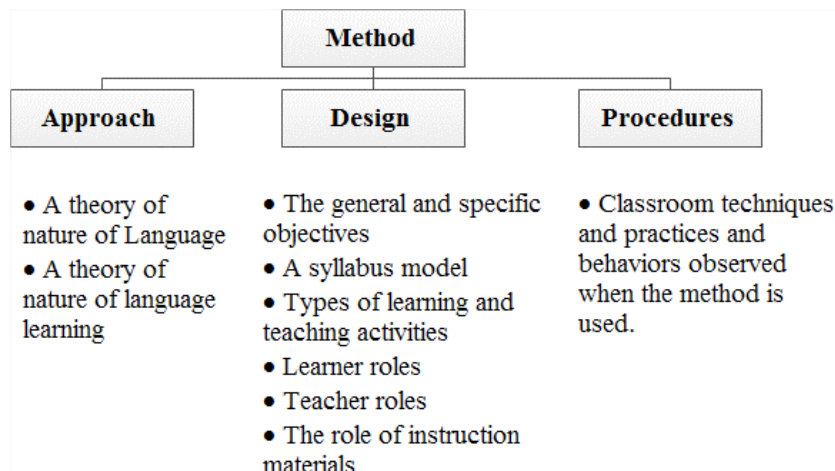


Figura 2.1 Conceptualización del método. Fuente: (Richards & Rodgers, 1963)

2.2.1.1. ENFOQUE (APPROACH)

El enfoque se refiere a las teorías sobre la naturaleza del aprendizaje el cual sirve como fuente de prácticas y principios en la enseñanza.

2.2.1.2. DISEÑO (DESIGN)

Para que el enfoque conduzca a un método, es necesario desarrollar un diseño sistemático de instrucción en el cual se considera:

- Cuáles son los objetivos del método.
- Cómo se selecciona y organiza el plan de estudios que se incorporan en el método.
- Los tipos de tareas de aprendizaje y actividades de enseñanza que el método defiende.
- Los roles de los estudiantes.
- Los roles de los profesores.
- El papel de los materiales de instrucción.

2.2.1.3. PROCEDIMIENTOS (PROCEDURES)

El último nivel de conceptualización y organización dentro de un método es a lo que nos referiremos como procedimiento. Esto abarca las técnicas, prácticas y comportamientos actuales de momento a momento que operan en la enseñanza de acuerdo a lo planteado. Es el nivel en cual se descubre cómo un método hace cuenta de su enfoque y diseño en el

comportamiento de aula. A nivel de diseño, se apreció que el método recomienda el uso de ciertos tipos de actividades de enseñanza como consecuencia de sus suposiciones teóricas sobre la programación.

A nivel de procedimiento, se hace hincapié en las tareas y actividades que se integran en las lecciones y se utilizan como base para la enseñanza y el aprendizaje.

Hay tres dimensiones para un método en el nivel de procedimiento:

- a. El uso de actividades de enseñanza (simulacros, diálogos, actividades de brechas de información, etc.) para presentar las temáticas y los recursos en términos de tiempo, espacio y equipo utilizados por el estudiante y profesor.
- b. Las formas en que las actividades de enseñanza particulares se utilizan para practicar.
- c. Los procedimientos y técnicas utilizados para dar retroalimentación de los contenidos, asignaciones y estrategias utilizadas por el profesor y los estudiantes cuando se utiliza el método.

2.2.2. FACTORES QUE INFLUYEN SOBRE UN MÉTODO

También se debe considerar los factores que influyen en método (ver Figura 2.2).

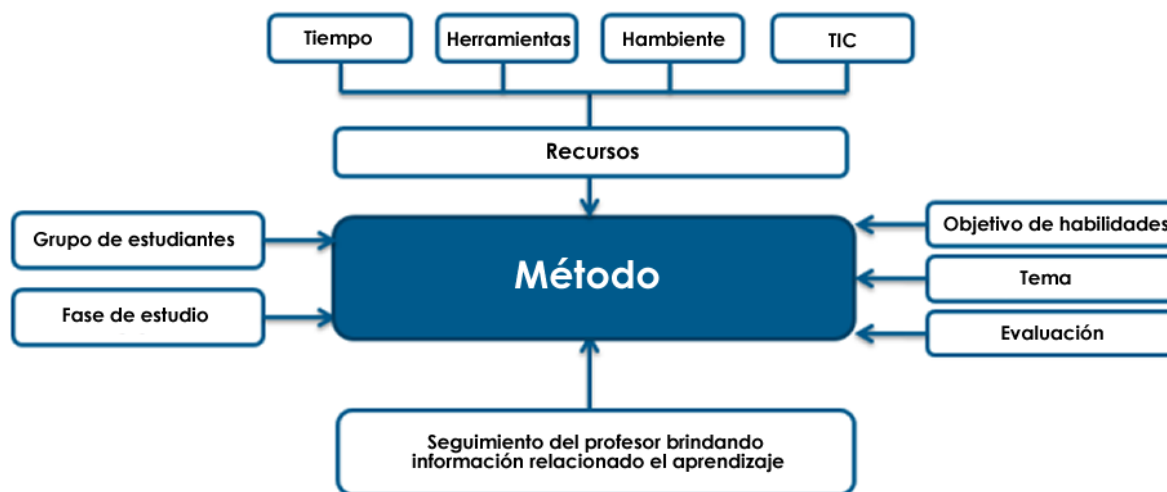


Figura 2.2 Factores que influyen sobre un método. Fuente: (Edward & Bentley, 2011)

2.3. ASPECTOS EDUCATIVOS RELEVANTES

2.3.1. EL CONO DEL APRENDIZAJE DE EDGAR DALE

En 1966, Edgar Dale, pedagogo con doctorado, desarrolló un modelo que buscaba explicar con claridad cuáles son los métodos que tienden a funcionar o fracasar en el aprendizaje, con el fin de presentar nuevas alternativas a lo que se denominó la pirámide del aprendizaje (ver Figura 2.3) (Hackbarth, 1996).



Figura 2.3 Pirámide del aprendizaje. Fuente: (Dale, 1966)

2.3.2. LA PIRÁMIDE DE MILLER

George A. Miller, ofrece una alternativa para orientar a los educadores en la realización para la evaluación de las competencias. La pirámide (ver Figura 2.4) consta de cuatro niveles: el primero, el más sencillo de lograr, es conocer el tema, estudiarlo para obtener los conocimientos pertinentes.

En el segundo nivel, la persona es capaz de describir cómo lo haría. Hasta aquí se trabaja en el nivel cognitivo.

Los otros dos niveles superiores de la pirámide permanecen a lo conductual. Por ello el tercer nivel consiste en demostrar en una simulación cómo lo haría; aquí ya sabe hacer, pero no en la realidad escolar, sino en una simulación de clase.

Y, por último, el cuarto nivel, trabaja sobre las acciones en la realidad lo que antes conoció y lo realizó en una simulación, es decir, ahora actúa en la realidad (Miller, 1990).

Hay diversas definiciones de lo que se entiende por competencia, en la presente investigación se considera el aporte de Ángel Pérez Gómez (2008) en la que predomina la concepción de un sistema de acción complejo, multidimensional y reflexivo, y la define como: “la capacidad de responder a demandas complejas y llevar a cabo tareas diversas de forma adecuada. Supone una combinación de habilidades prácticas, conocimientos, motivación, valores, actitudes, emociones y otros componentes sociales y de comportamiento que se movilizan conjuntamente para lograr una acción eficaz” (Pérez, 2008, pág. 59).



Figura 2.4 Guía para la evaluación de competencias en medicina. Fuente: (Miller,1990)

2.3.3. EL DIAGRAMA V DE GOWIN

Bob Gowin, psicólogo con premio a la excelencia a la enseñanza, diseñó un recurso para ayudar a los estudiantes y profesores a captar el significado de los materiales que se van a aprender (González, 2008, pág. 78). Es un método que permite entender la estructura del conocimiento y el modo en que éste se produce. Gowin propone el diagrama V (ver Figura 2.5) como una herramienta que puede ser empleada para analizar críticamente un trabajo de investigación, así como para “extraer o desempaquetar” el conocimiento de tal forma que pueda ser empleado con fines instruccionales.

El diagrama V, deriva del método de las cinco preguntas:

1. ¿Cuál es la pregunta determinante?
2. ¿Cuáles son los conceptos clave?
3. ¿Cuáles son los métodos de investigación que se utilizan?
4. ¿Cuáles son las principales afirmaciones de conocimiento?
5. ¿Cuáles son los juicios de valor?

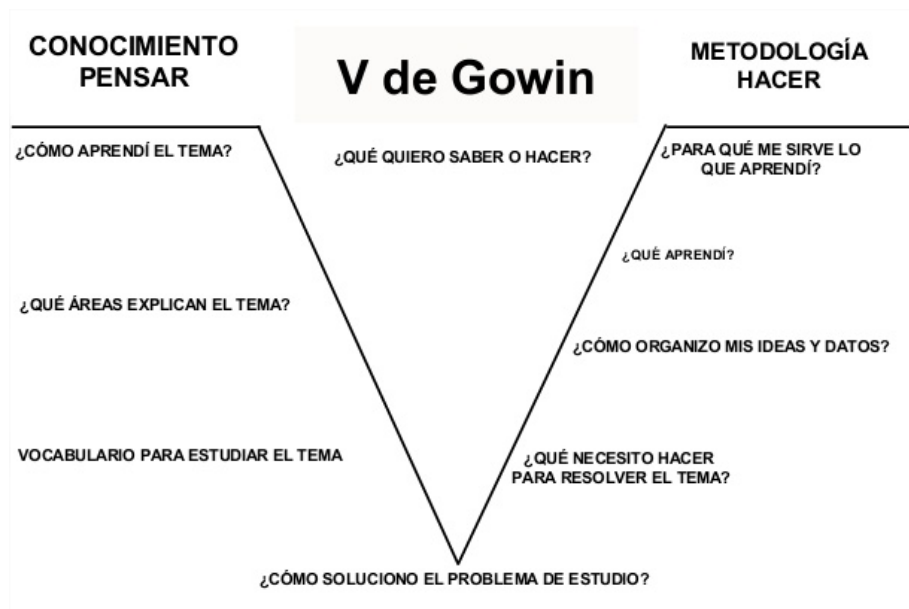


Figura 2.5 Esquema simplificado del Diagrama V. Fuente: (Novak & Gowin, 1988)

2.3.4. LA TAXONOMÍA DE BLOOM

Benjamín Bloom, Doctor en Educación de la Universidad de Chicago, formuló una Taxonomía de Dominios del Aprendizaje en 1956, desde entonces conocida como Taxonomía de Bloom, que puede entenderse como “Los Objetivos del Proceso de Aprendizaje” (Anderson & Krathwohl, 2001). Esto quiere decir que después de realizar un proceso de aprendizaje, el estudiante debe haber adquirido nuevas habilidades y conocimientos. En los años 90, antiguos estudiantes de Bloom, Lorin Anderson y David R. Krathwohl, revisaron la Taxonomía de su maestro y la publicaron en 2001 (Prensky, 2001). Uno de los aspectos clave de esta revisión fue el cambio de los sustantivos de la propuesta original a verbos, para significar las acciones correspondientes a cada categoría. Otro aspecto fue considerar la síntesis con un criterio más amplio y relacionarla con crear (considerando que toda síntesis es en sí misma una creación); además, se modificó la secuencia en que se presentan las distintas categorías. A continuación, se presentan las categorías en orden ascendente, de inferior a superior (Figura 2.6).

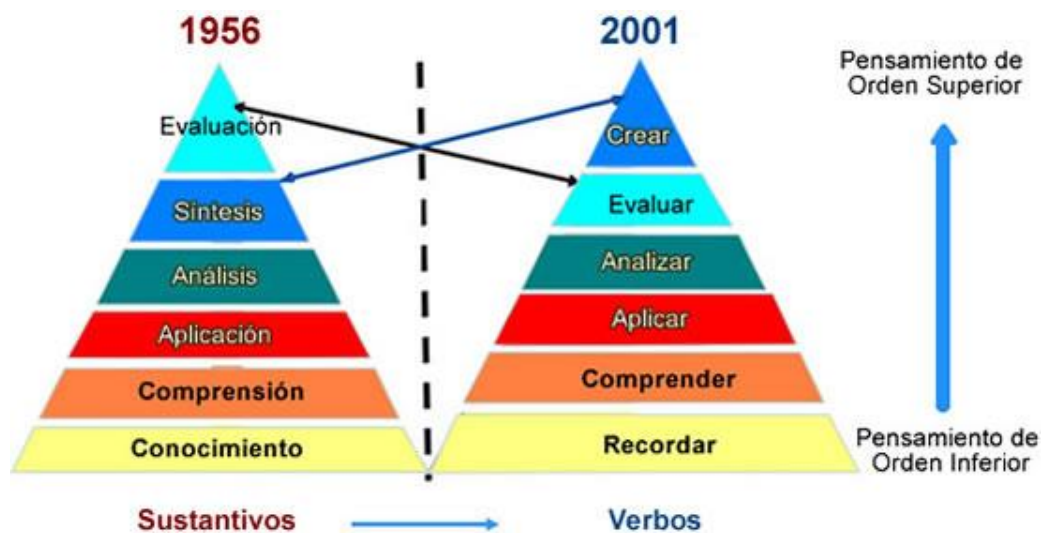


Figura 2.6 Taxonomía de Bloom. Fuente: (Bloom,2001)

2.3.5. UNA TEORÍA DE APRENDIZAJE PARA LA ERA DIGITAL, EL CONECTIVISMO

El conductismo, el cognitivismo y el constructivismo son las tres grandes teorías de aprendizaje utilizadas más a menudo en la creación de ambientes instruccionales. Estas teorías, sin embargo, fueron desarrolladas en una época en la que el aprendizaje no había sido impactado por la tecnología. En los últimos veinte años, la tecnología ha reorganizado la forma en la que vivimos, nos comunicamos y aprendemos. Las necesidades de aprendizaje y las teorías que describen los principios y procesos de aprendizaje, deben reflejar los ambientes sociales profundos.

Hace tan solo cuarenta años, los aprendices, luego de completar la educación formal requerida, ingresaban a una carrera que normalmente duraría toda su vida. El desarrollo de la información era lento. La vida del conocimiento era medida en décadas. Hoy, estos principios fundamentales han sido alterados. El conocimiento crece exponencialmente. En muchos campos la vida del conocimiento se mide ahora en meses y años (Gonzalez, 2004).

Un principio central de la mayoría de las teorías de aprendizaje es que el aprendizaje ocurre dentro de una persona. Incluso los enfoques del constructivismo social, los cuales sostienen

que el aprendizaje es un proceso social, promueven el protagonismo del individuo (y su presencia física, es decir, basado en el cerebro) en el aprendizaje. Estas teorías no hacen referencia al aprendizaje que ocurre por fuera de las personas, aprendizaje que es almacenado y manipulado por la tecnología). Más adelante se puede apreciar la tabla comparativa de las teorías del aprendizaje (Tabla 2.1).

El conectivismo fue presentado el año 2004 por George Siemens, como una teoría que supera “las tres grandes teorías” sobre el aprendizaje, superando respecto a las anteriores sus limitaciones a la hora de interpretar los efectos, las ventajas y que las supera también en la concepción de la naturaleza con que se produce el conocimiento en entornos tecnológicos, procesos de la información y de la comunicación, integrando los principios explorados por las teorías de caos, redes, complejidad y auto-organización. (Recio, Díaz, Saucedo, & Jiménez, 2017).

Tabla 2.1
Comparativa de las distintas teorías del aprendizaje

Propiedad/Teoría	Conductismo	Cognitivismo	Constructivismo	Conectivismo
¿Cómo se produce el aprendizaje?	Según el comportamiento por estímulos asociados.	Estructurado computacional.	Social, significado creado por cada estudiante (personal).	Distribuido dentro de una red social, mejorado tecnológicamente, reconociendo e interpretando patrones
Factores que influyen	Naturaleza de recompensa, castigo, estímulo.	Esquema existente, experiencias anteriores.	Compromiso, participación, sociales y culturales.	Diversidad de la red. la fuerza de los vínculos.
Rol de la memoria	La memoria es el resultado de repetidas experiencias, donde la recompensa y el castigo son influyentes.	Codificación, almacenamiento y recuperación.	Conocimiento previo remezclado al contexto actual.	Patrones de adaptación, representativos del estado actual que existe en las redes.
¿Cómo ocurre la transferencia?	Estímulo, respuesta.	Duplicación de las construcciones de conocimiento del conocedor.	Socialización	Conectando a (agregando) redes.
Otra forma de conocerlo	Aprendizaje basado en tareas.	Razonamiento, objetivos claros, la resolución de problemas.	Social, vago ('mal definido').	Aprendizaje complejo, diversas fuentes de conocimiento.

Nota: Fuente: Elaboración propia 2017

2.3.6. ENSEÑANZA Y APRENDIZAJE BAJO PRESIÓN

En la educación superior se vienen ofreciendo o se están comenzando a ofrecer cursos de informática enseñados en una escala de tiempo comprimida, a menudo donde los estudiantes toman solo un curso a la vez. El ritmo de estas clases puede diferir, pero a menudo se mueve a una velocidad donde el día que corresponde a una clase acelerada es equivalente a una semana de una clase semestral típica. Este formato tiene varias ventajas: más flexibilidad para el trabajo colaborativo, una mejor visibilidad de cómo los estudiantes pasan su tiempo, asumiendo que si un estudiante tiene dificultades no es debido al trabajo en una clase diferente y menos a la necesidad de realizar tareas múltiples entre diferentes alumnos. También tiene muchos desafíos (Burge & Brinkman, 2017).

Según Bunge y Brinkman (2017), desde la perspectiva del estudiante, existe la necesidad de estar al tanto de las cosas y no quedarse atrás, tener que ponerse al día si hay ausencias por enfermedad o actividades extracurriculares, y mantenerse enfocado en un tema mientras trabaja bajo presión durante un tiempo determinado (que a menudo resulta en una tendencia a apresurarse en las tareas para cumplir con los plazos). Para conseguir este fin se debe definir roles para los participantes, las evaluaciones del curso, disponer una planificación minuciosa para mantener una carga de trabajo razonable para los estudiantes y el profesor, además de seleccionar los enfoques pedagógicos necesarios.

2.4. PENSAMIENTO COMPUTACIONAL

Según (Wing, 2006), “El pensamiento computacional implica la resolución de problemas, diseño de sistemas, y la comprensión de la conducta humana, haciendo uso de los conceptos fundamentales de la informática. Pensamiento computacional incluye una serie de herramientas mentales que relejan la amplitud del campo de la informática”.

Años más tarde se revisó la definición indicando: “pensamiento computacional es el proceso de pensamiento involucrado en la formulación de los problemas y sus soluciones

para que estas soluciones estén representadas de forma que se puedan llevar a cabo con eficacia por un agente de procesamiento de información.” (Wing, 2008).

De esta forma la CSTA⁹, ISTE¹⁰ y Google presentaron los siguientes pasos definidos a seguir para desarrollar el pensamiento computacional, expuestos en la Tabla 2.2.

Tabla 2.2
Comparativa y definiciones de pensamiento computacional

CSTA / ISTE	Google
(1) Usar ordenadores para resolver problemas	(1) Descomposición
(2) Organizar datos lógicamente	(2) Patrones y tendencias
(3) Abstracciones, modelos y simulaciones	(3) Generalizar patrones
(4) Pensamiento algorítmico	(4) Desarrollar algoritmos
(5) Buscar soluciones eficientes y eficaces	
(6) Generalizar las soluciones	

Nota. Fuente: Elaboración propia 2017

A partir de las definiciones anteriores y con el objetivo de poder evaluar el conjunto de habilidades que definen el pensamiento computacional, se puso a disposición el siguiente conjunto de habilidades: análisis, abstracción y secuenciación. En la

Tabla 2.3 se detallan estas habilidades.

⁹ **CSTA** (Asociación de Maestros en Ciencias de la Computación) es una asociación profesional que apoya y fomenta la educación en el campo de la informática y áreas relacionadas.

¹⁰ **ISTE** (La Sociedad India para la Educación Técnica) Trabaja en las áreas de educación y tecnológica. Su objetivo principal, ayudar y contribuir en la producción para desarrollo de ingenieros y técnicos profesionales de la más alta calidad.

Tabla 2.3
Habilidades que define el Pensamiento Computacional.

Habilidades	
(1) Análisis	Habilidad para encontrar patrones y tendencias.
(2) Abstracción	Uso de la abstracción para la representación de los datos.
(3) Secuenciación	Habilidad para dividir y estructurar ordenadamente los pasos en los que se divide un problema.

Nota. Fuente: Elaboración propia 2017

En este apartado se presentan algunas experiencias de algunos expertos en cuanto a la enseñanza del pensamiento computacional en alumnos de secundaria. La primera experiencia sigue el punto de vista de enseñar Pensamiento Computacional “sin programar”. En las siguientes se sigue el punto de vista donde se enseña el pensamiento computacional “a través de la programación”.

2.4.1. ENSEÑAR PENSAMIENTO COMPUTACIONAL “SIN PROGRAMAR”

Mark Joseph Guzdía afirma que: “Enseñar el pensamiento computacional a todo el mundo puede requerir enfoques diferentes a los que usamos cuando asumimos que nuestros estudiantes quieren convertirse en profesionales de la informática” (Guzdía, 2008).

Para James Lu, la programación no ha de ser el primer paso, pues existe una serie de conceptos y aprendizajes previos que deben ser abordados por los estudiantes. Según James y Fletcher, debemos separar la enseñanza del pensamiento computacional de la programación en sí. Por otro lado, afirma que la complejidad de los lenguajes de programación, son de nivel superior a las habilidades que aporta el pensamiento computacional (James & Fletcher, 2009).

Por lo tanto, para enseñar estas habilidades en otras disciplinas distintas a la informática “tenemos que poner el énfasis en la comprensión y ser capaz de realizar de forma manual procesos computacionales, y no en sus manifestaciones en los lenguajes de programación

específicos” de tal manera que la programación se convierte en un medio y no un fin (James & Fletcher, 2009).

Según los investigador James & Fletcher, el pensamiento computacional pretende que los estudiantes se familiaricen con los conceptos algorítmicos como flujo básico de control y desarrollen habilidades para la abstracción¹¹ y la presentación de información, y para la evaluación de las propiedades de los procesos, mediante el aprendizaje de un vocabulario y lenguaje de pensamiento computacional.

Las actividades son divertidas y atractivas, por lo general, las explicaciones son bastante breves (el profesor expone los materiales y unas pocas reglas), son los estudiantes los que alcanzan los objetivos. Para el desarrollo de las actividades no se necesita ningún equipo especializado, los materiales necesarios son de bajo costo. Las actividades se planean como módulos independientes y están preparadas de manera que si se cometen algunos errores se pueda alcanzar igualmente los objetivos.

Son diversas las organizaciones que apoyan este tipo de enseñanza, entre ellas: ACM K-12¹² y los comités curriculares; CSTA, CS4ALL¹³, NCWIT (Centro Nacional de EE.UU para Mujeres y Tecnología de la Información), TECS (Enriquecimiento docente en informática), Google y Microsoft.

2.4.2. PENSAMIENTO COMPUTACIONAL “A TRAVÉS DE LA PROGRAMACIÓN”

En el segundo punto de vista, la enseñanza del pensamiento computacional, se basa en el uso de entornos y herramientas. En esta instancia llegamos a ver la aplicación de estrategias metodológicas basada en el uso del programa *Scratch*¹⁴ + Aprendizaje Basado en

¹¹ **Abstracción.** Es una operación intelectual que separa las cualidades de un objeto para considerarlas aisladamente o para analizar al objeto en su pura esencia.

¹² **K-12.** Es la designación utilizada en algunos sistemas educativos para la escolarización primaria y secundaria.

¹³ **CS4All** Es un innovador programa educativo de informática que brinda equidad, empoderamiento y oportunidades que maximizan el potencial innato de cada estudiante.

¹⁴ **Scratch.** Es un lenguaje de programación visual desarrollado por el MIT Media Lab.1 Scratch es utilizado por estudiantes, académicos, profesores y padres para crear fácilmente animaciones, juegos e interacciones etc.

Problemas (ABP¹⁵), para el desarrollo de competencias del pensamiento computacional. (Rios, 2015).

Gracias a su entorno Scratch permite una fácil incorporación de usuarios que no estén familiarizados en programación pero que al mismo tiempo permiten desarrollos en el pensamiento de elevada complejidad.

Estas herramientas cumplen con los requisitos citados por Seymour Papert. Los lenguajes de programación han de tener un “suelo bajo” (fácil empezar a programar) y un “techo alto” (oportunidades de crear proyectos mucho más complejos con el tiempo) (Gillespie, 2004).

Hace una década, Wing publicó el artículo fundacional de la disciplina, en el cual define que: el “pensamiento computacional implica la resolución de problemas, el diseño de sistemas, y la comprensión de la conducta humana, haciendo uso de los conceptos fundamentales de la informática” (Wing, 2006, pág. 33). Así, la esencia del pensamiento computacional sería pensar como un científico de la computación cuando uno enfrenta a un problema.

En dicho artículo, se afirma que el pensamiento computacional “representa un conjunto de habilidades y actitudes, aplicable universalmente; que toda persona (no sólo los científicos de la computación) debería estar ansiosa por adquirir y usar” (Wing, 2006, pág. 33).

Así mismo, Wing define al pensamiento computacional como una habilidad básica y fundamental para cualquier sujeto inmerso en la actual realidad digital, llegando incluso a afirmar que el pensamiento computacional debería añadirse a la “lectura, escritura, aritmética” como paquete básico en el desarrollo de las habilidades analítico-instrumentales de cualquier niño.

¹⁵ **Aprendizaje Basado en Problemas.** Es un método de enseñanza-aprendizaje centrado en el estudiante en el que éste adquiere conocimientos, habilidades y actitudes a través de situaciones de la vida real.²

A continuación, se sintetiza qué es, y qué no es, pensamiento computacional en esta aproximación pionera de (Wing, 2006):

- **Conceptualizar, no programar.** Pensar como un científico de la computación, es decir, pensar computacionalmente, significa más que ser capaz de programar un ordenador. Requiere pensar en múltiples niveles de abstracción.
- **Habilidad básica, no puramente mecánica.** Una habilidad básica es aquella que cualquier persona debe dominar para funcionar en la sociedad actual.
- **Una manera en que los humanos piensan, no las computadoras.** El pensamiento computacional es una manera en que los seres humanos solucionan problemas; no es tratar de que los humanos piensen como las computadoras. Las computadoras son previsibles y aburridas; los humanos, inteligentes e imaginativos. Somos los humanos los que hacemos a las computadoras interesantes. Los humanos, equipados con la potencia de las computadoras, usamos nuestra inteligencia para abordar y resolver problemas que no seríamos capaces de atacar en la era previa a la computación.
- **Se complementa y se combina con el pensamiento matemático e ingenieril.** Las ciencias de la computación y, por tanto, el pensamiento computacional, descansan sobre el pensamiento matemático dado que, como todas las ciencias, sus bases formales surgen de las matemáticas. Además, las ciencias de la computación descansan inherentemente en el pensamiento ingenieril dado que trata de construir sistemas informáticos que interactúan con el mundo real-físico.
- **Ideas, no artefactos-objetos.** El pensamiento computacional no tiene sólo que ver con los objetos de hardware y software que se produce, y que están presentes físicamente a diario en nuestras vidas; tiene que ver sobre todo con los conceptos computacionales que se usa para aproximarnos a los problemas y solucionarlos, para gestionar nuestra vida, y comunicarnos e interactuar con otras personas de manera efectiva.

- **Para cualquiera, en cualquier parte.** El pensamiento computacional será una realidad cuando sea tan inherente a las tareas y esfuerzos humanos, que desaparezca como término y filosofía explícitos.

Esta primera definición genérica, casi metafísica, del pensamiento computacional, viene siendo revisada y especificada en intentos sucesivos a lo largo de los últimos años, sin llegar aún a un acuerdo generalizado sobre la misma.

Por ello, Wing revisa su definición para 2008, y aclara que “el pensamiento computacional incluye los procesos de pensamiento implicados en la formulación de problemas y de sus soluciones, de tal modo que éstos estén representados de una manera que pueda ser abordada efectivamente por un agente-procesador de información” (Wing, 2008, pág. 3718) como un ordenador. Siguiendo con este grupo de definiciones, de corte genérico, cuatro años más se declara que el pensamiento computacional es “el proceso de pensamiento involucrado en la formulación de problemas de tal manera que sus soluciones puedan ser representadas como pasos computacionales discretos y algoritmos” (Aho, 2012, pág. 832). En ese mismo año, la *Royal Society*¹⁶ también aporta una definición que trata de capturar la esencia del pensamiento computacional:

“El pensamiento computacional es el proceso de reconocimiento de los aspectos computables en el mundo que nos rodea, y de aplicar las herramientas y técnicas de las Ciencias de la Computación para comprender y razonar sobre sistemas y procesos, tanto naturales como artificiales” (Society, 2012, pág. 29).

2.5. PROGRAMACIÓN.

Mucho del comportamiento y pensamiento humano se caracteriza por secuencias lógicas. Por ejemplo, en una escala más amplia, las matemáticas nunca se podrían haber desarrollado sin secuencias lógicas de pasos para resolver problemas y demostrar teoremas.

¹⁶ Royal **Society**, es una de las sociedades científicas más antiguas del mundo con más de 350 años de existencia y sede en Londres.

La producción en masa nunca habría funcionado sin operaciones que tienen lugar en cierto orden. La civilización se basa en el orden de las cosas y acciones.

Se crea orden, de manera consciente e inconsciente, en un proceso al que se denomina programación.

De esta forma cuando se cita el término programación de computadoras, se hace referencia al efecto de programar, es decir, de organizar una secuencia de pasos ordenados a seguir para hacer cierta tarea. La computadora permite hacer las tareas con más eficiencia, rapidez y exactitud de cómo se podrían hacer a mano, si acaso se pudieran hacer a mano. A fin de usar esta poderosa herramienta, se debe especificar lo que se desea hacer y el orden mediante el uso de un lenguaje de programación, esto es posible por medio de la programación que luego puede ser interpretado a lenguaje máquina y ejecutado por el computador (Dale, Dale, & Weems).

2.5.1. LA PROGRAMACIÓN COMO UN ARTE

En la programación se utiliza técnicas, teorías, reglas, principios y, por otra parte, una buena dosis de inspiración sublime prácticamente artística, asumiendo que el código está escrito en algún lenguaje, apegado, en mayor o menor medida, a un paradigma regido por sus propias reglas; que a su vez fue escrito por personas con distintos talentos.

Importantes autores siguen considerando que la programación aún no está en condiciones de recibir el título de ciencia. Entre estos autores se considera a Budd (2002) quien señala: “Es cierto que programar un computador es todavía una de las tareas más difíciles jamás emprendidas por la humanidad; llegar a ser un experto en programación requiere talento, creatividad, inteligencia, lógica, habilidad para construir y usar abstracciones, y experiencia, aun cuando se disponga de las mejores herramientas” (Budd, 2002).

Por otra parte, (Glenn, 1995) comenta: “El proceso de descubrimiento de algoritmos y de resolución de problemas tienen una íntima relación. Así, la capacidad para resolver

problemas sigue siendo más una aptitud artística que debe desarrollarse, que una ciencia precisa por aprender.” Finalmente, (Knuth, 1997), quizás uno de los programadores más notables, nombró a su libro el Arte de la Programación subrayando que a pesar de todos los formalismos que se han creado, el diseño de algoritmos y su codificación sigue siendo un arte.

2.5.2. MÉTODOS DE ANÁLISIS DE PROBLEMAS

Todo programa, sin importar su naturaleza, en esencia lo que busca es resolver un problema de la vida real automatizando una serie de actividades. Esta automatización está codificada en un lenguaje de programación, una sintaxis para expresar un algoritmo de forma que una computadora pueda ejecutarlo. Diseñar el algoritmo y codificarlo es de hecho encontrar la solución al problema, Brooks (1986) lo explica de esta forma:

“Descubrir un algoritmo para resolver un problema es equivalente, en lo esencial a descubrir una solución para ese problema” (Brooks, 1986, pág. 45).

Todo el esfuerzo e inteligencia requeridos para encontrar la solución a un problema se encuentran de alguna forma, encapsulados dentro del algoritmo. En tal sentido Glenn afirma lo siguiente: “Una vez descubierto un algoritmo para efectuar una tarea, la realización de esta ya no requiere entender los principios en los que se basa dicho algoritmo, pues el proceso se reduce a seguir ciertas instrucciones. En cierto sentido, la inteligencia requerida para llevar a cabo la tarea está codificada en el algoritmo. Es gracias a esta capacidad para capturar y comunicar inteligencia mediante algoritmos que se pueden construir máquinas cuyo comportamiento simula inteligencia. En consecuencia, el nivel de inteligencia que exhiba una máquina estará limitado por la inteligencia que se pueda comunicarle por medio de un algoritmo” (Brooks, 1986).

Es la inteligencia humana la que permite encapsular pedazos de inteligencia dentro de programas a través de los algoritmos. Sin embargo, este proceso no es de ninguna manera trivial. Encontrar soluciones a problemas es en sí mismo un problema. Lo ha definido de la

siguiente forma: “El mayor obstáculo que enfrentan los novatos al aprender a programar es lo que se podría llamar: ‘El problema de enfrentar una pantalla en blanco’, lo cual consiste en cómo ir de un vago planteamiento de un problema a un programa funcional que lo resuelva” (Decker & Hirshfield, 1992).

Diversos factores intervienen en este proceso de ir de un vago planteamiento a una solución computacional. Entre ellos se encuentran la capacidad de abstracción y el análisis., Joyanes (1987) ha agregado la independencia del algoritmo al señalar lo siguiente: “El proceso de realización de diferentes pasos hasta encontrar la solución de un problema es un proceso abstracto. Diseñar o concebir un problema en términos abstractos consiste en no tener en cuenta la máquina que lo va a resolver, así como el lenguaje de programación que se va a utilizar. Esto lleva consigo la obtención de un conjunto de acciones que se han de realizar para obtener la solución” (Joyanes, 1987).

Finalmente, no se puede olvidar el factor humano por sí mismo en materia de talento e inspiración. A Edison se le atribuye la siguiente frase: “Mi éxito depende del 98% de sudoración por un 2% de inspiración”. En el área computacional Glen (1995) ha escrito sobre lo que llama irregularidades en el proceso de creación de algoritmos lo siguiente: “Otra irregularidad es la misteriosa inspiración que puede llegar a un potencial solucionador de problemas, quien, después de trabajar con un problema sin éxito aparente, puede ver de repente la solución mientras realiza otra tarea (Glenn, 1995). Helmholtz identificó este fenómeno ya desde 1896, y el matemático Henri Poincaré lo analizó en una conferencia ante la sociedad psicológica de París. Hoy día, al período entre el trabajo consciente con un problema y la llegada de la inspiración repentina se le llama periodo de incubación, y su comprensión es una meta de varias investigaciones actuales.

Hay autores que abordan la solución de problemas a través de métodos para elaborar programas (lo cual en estricto sentido es válido si se visualiza el programa como la solución al problema) mientras que otros se van a enfoques más globales al abordar una metodología genérica de solución de problemas.

En tal sentido, la resolución de problemas no es una necesidad monopolizada por la computación. Muchas otras ciencias han requerido de esta habilidad desde muchos años antes al nacimiento de la primera computadora. Sin embargo, el método de resolución de problemas más citado es el del matemático Polya¹⁷. “Las fases definidas a grandes rasgos por el matemático G. Polya a finales de los años 40's siguen siendo los principios fundamentales en que se basan los intentos actuales de enseñar a resolver problemas (Glenn, 1995). En realidad, los trabajos de Polya se apoyaron en las investigaciones de H. Von Hemholtz a fines del siglo antepasado, J. Dewey en la década de los 30's y otros que estudiaron el proceso de resolución de problemas en situaciones generales. Los científicos de la computación suelen citar a Polya porque sus trabajos abordaron la resolución de problemas en el contexto de las matemáticas, pariente cercano de la computación. Las fases de Polya para resolver problemas son:

Fase 1. Entender el problema.

Fase 2. Idear un plan para resolver el problema.

Fase 3. Llevar a cabo el plan.

Fase 4. Evaluar la solución en cuanto a su exactitud y a su potencial como herramienta para resolver otros problemas.

Glenn rescribe estas fases, pero ahora con ojos computacionales llegando a lo siguiente:

Fase 1. Entender el Problema.

Fase 2. Pensar como un procedimiento algorítmico podría resolver el problema.

Fase 3. Formular el algoritmo y representarlo en forma de programa.

Fase 4. Evaluar el programa en cuanto a su exactitud y potencial como herramienta para resolver otros problemas.

Un autor indica: “La técnica consiste en dar una secuencia de sucesos para resolver el problema. Primero comprender el problema. Segundo, idear algún método para obtener una

¹⁷ **George Pólya:** Nació en Hungría en 1887 se lo considera el padre de las estrategias para la solución de problemas.

solución. Escribir la solución en términos generales. Detallarla en procesos sucesivos.” (Gosling, 1985).

Es importante no olvidar que no son fases, por lo tanto, no tienen que ser seguidas en orden ni tampoco debe haberse concluido una de ellas totalmente antes de iniciar la siguiente.

Muchas veces ni siquiera se puede tener un entendimiento del problema si no se cuenta con la solución (lo cual resulta paradójico). Es válido hacer procesos cíclicos comprobando hipótesis para ir logrando la comprensión del problema, por lo tanto, no es necesario terminar una etapa antes de iniciar la siguiente (de hacerlo así quizás nunca se encuentre la solución ya que quizás nunca se pueda entender completamente el problema si no se avanza sobre él).

El planteamiento fundamental consiste en dar el primer paso tratando de hacer algún análisis sobre la información que se tiene. Lo que sigue es conservar el impulso del primer paso. Para ello tanto Polya y sus seguidores han identificado al menos tres estrategias:

1. Resolver el problema al revés. Esto es identificar la salida, luego los procedimientos para obtenerlos, las restricciones y las entradas.
2. Buscar problemas relacionados. Investigar sobre problemas similares que ya hayan sido resueltos, analizar la solución y tratar de copiarla adaptándola al contexto vigente.
3. Refinamiento por pasos. Dividir el problema en partes (o módulos) y seguir dividiéndolo hasta sus componentes esenciales. Esto puede hacerse con el enfoque top-down o el bottom-up (incluso en el mundo orientado a objetos).

A pesar de que ésta última estrategia de refinamiento y desglose descendente tiene ya varios años de haberse publicado, diversos autores, ya entrados en los 90's, señalaban su vigencia y utilidad. Entre ellos Alcalde y García (1992) escribían sobre los refinamientos: “Los programas se diseñan de lo general a lo particular por medio de sucesivos refinamientos o descomposiciones que nos van acercando a las instrucciones finales del

programa”. En el mismo libro anota en otra ocasión “La programación estructurada y modular insiste en la descomposición en módulos que dividan el problema en sus funciones esenciales (descomposición funcional) y que posteriormente, a través de varios ciclos de refinamiento se obtengan los algoritmos específicos. El proceso de refinamiento cíclico sigue vigente.” (Alcalde & Garcia, 1992). Por su parte, (Perry, 1999), habla sobre el diseño descendente “El diseño descendente es el proceso de descomposición de un problema en detalles más específicos, hasta completar todos los detalles.”.

Entre las metodologías propuestas que se concentran en la solución de problemas a través de las etapas de elaboración de programas están las de Gosling, Joyanes, Decker, Alcalde, Glenn, Levine y Perry. Entre ellas hay una extrema similitud y retoman ideas de Polya sin hacer mención explícita de su modelo.

Según Gosling (1985): “El proceso de diseñar un programa consta de varias etapas. La primera de ellas consiste en la comprensión del problema que se va a resolver mediante un programa. La segunda en encontrar un método de resolución del problema. A continuación, escribimos una solución no en la forma de un diagrama de flujo, sino en la forma de una serie de pasos bien definidos. Estos pasos se establecen en términos amplios en las primeras etapas del desarrollo del programa y luego se van refinando hasta que se obtiene un programa completamente lógico.”.

Así mismo, Joyanes (1987), explica seis etapas para elaborar un programa de la siguiente forma: “El desarrollo de un programa abarca diferentes etapas, de las cuales la escritura puede ser la menos significativa. La primera es la definición del problema; la segunda es el diseño del algoritmo (algoritmo normalizado); la tercera es la codificación; la cuarta es la depuración; la quinta es la documentación; y la sexta es el mantenimiento.” (Joyanes, 1987) Siendo el diseño del algoritmo la parte central del proceso describe en las etapas para su desarrollo explicando: “El algoritmo de resolución consta de dos etapas: 1. Diseño del modelo de resolución del problema; y 2. Algoritmo de resolución del problema.”.

Finalmente, señala que para poder realizar el diseño de un programa funcionalmente exitoso se deben seguir las siguientes reglas:

1. Ir de lo general a lo particular descendiendo en la estructura del programa hasta su nivel de detalle.
2. De la definición inicial del problema se pasa a un esquema de algoritmo descrito en pseudo código.
3. Independencia inicial del lenguaje.
4. Diseño por niveles, dejando los detalles para niveles posteriores.
5. Finalizar con una recomposición del algoritmo completo.

Levine (1989) propone un proceso para pasar de la descripción de un problema a un programa definiendo los siguientes pasos:

- Se propone una solución global al problema en términos de una descripción en un lenguaje llamado pseudo código. Esto será el primer acercamiento a la solución. Este pseudo código, describirá de manera aproximada, el procedimiento para resolver el problema.
- Se toma el módulo recién generado y se comienza a refinar progresivamente, tratando de traducir cada una de sus pseudo instrucciones en órdenes inteligibles para la computadora verificando, a la vez, que esté correcto.
- Se ejecuta el paso anterior sobre cada uno de los módulos obtenidos, hasta que no quede nada escrito en pseudo código y todo haya sido traducido a lenguaje de computadora.

Más adelante, el mismo autor, propone las siguientes fases para la creación de un programa:

1. Entender el problema.
2. Hacer el análisis del mismo.
3. Programar el modelo de solución propuesto.

4. Codificarlo.
5. Cargarlo a la computadora para su ejecución y ajuste.
6. Darle mantenimiento durante su tiempo de vida.

Decker y Hirshfield (1992) propone su metodología de elaboración de programas con los siguientes pasos:

1. Diseñar y especificar el programa. Describiendo el problema cuidadosamente, especificando las entradas y salidas, así como la interfaz del usuario.
2. Desarrollar el programa. Refinar sucesivamente el esquema original.
3. Analizar el diseño del programa y escribir el código (utilizando técnicas de verificación dónde sea apropiado).
4. Probar el programa y corregir cualquier error detectado lo más detalladamente posible.

Mantener, extender y modificar el programa. (Decker & Hirshfield, 1992).

Alcalde y García (1992) presentaron una metodología simplificada en tres fases:

1. Fase de Análisis: consiste en el examen y descripción detallada de la especificación del problema.
2. Fase de Programación: consiste en el diseño de la solución del problema planteado en forma de algoritmo.
3. Fase de codificación: transcripción del algoritmo a un lenguaje de programación. (Alcalde & Garcia, 1992),

Glenn (1995) describe de la siguiente forma las actividades necesarias para la creación de un programa: “La creación de un programa consta de dos actividades: Descubrir el algoritmo básico y representar ese algoritmo en forma de programa. ...el descubrimiento de algoritmos suele ser el paso más difícil en el proceso de creación de software; después de

todo, descubrir un algoritmo es equivalente a encontrar un método para resolver el problema cuya solución el algoritmo va a calcular.” (Glenn, 1995),

Perry (1999), propuso una metodología de diseño, acompañada de una serie de pasos para hacer el diseño descendente, que se describe a continuación:

Metodología de diseño:

1. Defina la salida.
2. Desarrolle la lógica para lograr esta salida.
3. Escriba el programa.

Pasos para el diseño descendente:

1. Determinar la meta general.
2. Dividir esa meta en dos o tres partes más detalladas.
3. Posponer los detalles mientras sea posible. Repetir los pasos 1 y 2 hasta que ya no pueda dividir más el problema. (Perry, 1999).

Es importante señalar que los autores citados tienen la particularidad de que sus ideas están enfocadas en los microprocesos de elaboración de programas y definición de algoritmos no proponen sus metodologías para enfrentar el análisis y diseño de sistemas ni como técnicas de ingeniería de software. Para ello (Senn, 1987), (Kendall & Kendall, 1997), (McMillan & González, 1977), (Pressman, 2010), (Sommerville, 2011) y otros han expuesto metodologías formales.

2.5.3. LÓGICA DE PROGRAMACIÓN

La Lógica de Programación es una habilidad mental, que requiere para su correcto desarrollo de otras habilidades mentales como la capacidad de abstracción, análisis, síntesis, inferencia, diseño, etc.; todo esto acompañado de fuertes dosis de creatividad e

ingenio. Sin embargo, y a pesar de su complejidad, es posible desarrollarla a través de reglas, estructuras y práctica.

Habilidad viene del término latino *habilis* que significa manejable, el diccionario define el término como la capacidad, disposición, gracia o destreza para hacer algo; en este caso programar. A su vez programar consiste en dividir cualquier tarea compleja en una sucesión de ordenes simples que estén al alcance de la computadora Finalmente, Lógica proviene del término griego *logos* (λόγος) que significa tratado o conocimiento (Pékelis, 1977).

Partiendo de la información expuesta se propone considerar a la Lógica de Programación como una habilidad mental utilizada para implementar soluciones computacionales (programas) orientados a ejecutar tareas complejas funcionalmente verdaderas (que hagan lo que se espera de ellos - Conclusiones válidas) a través de la combinación de diferentes estructuras de control y estructuras de datos (premisas) combinadas utilizando una serie de reglas formales (heurísticas o “Reglas de la Lógica”) (Ruiz, López, & Brito, 2016).

Según Ruiz, López, & Brito (2016) las habilidades a diferencia de los conocimientos, se adquieren con la experiencia vivencial de su praxis. Este es el principal problema de varios enfoques educativos, ya que pretenden que la persona “aprenda” las definiciones y conceptos de los elementos que forman la Lógica de Programación y los observen en la ejecución de unos cuantos ejercicios. Esta propuesta contempla que, además de realizar las descripciones teórico-conceptuales, se acompañe de un modelo de desarrollo de habilidades, apoyados en ejercicios, que demuestre tanto la operación de las estructuras de control como de las estructuras de datos básicas a través de la aplicación de las ya comentadas Reglas de la Lógica y otras herramientas históricamente validadas. En pocas palabras consiste en privilegiar el aprendizaje basado en la experimentación y vivencia.

La Lógica de Programación es una habilidad mental y se desarrolla a través de experiencias (prácticas) que deben ser planeadas en temática y complejidad para que estén vinculadas con su realidad y el resto de sus materias de estudio.

A continuación, se esquematiza en la (Figura 2.7) un mapa mental de la lógica de programación.

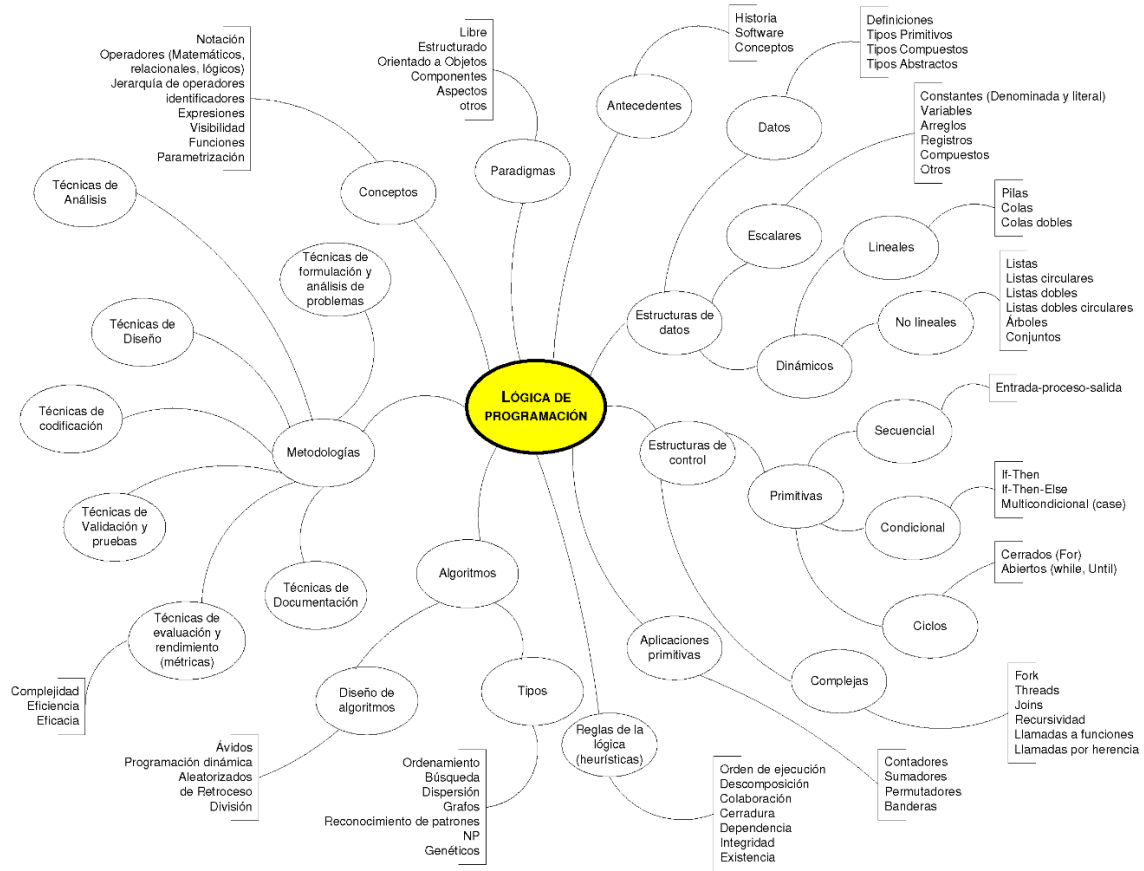


Figura 2.7 Mapa Mental de la Lógica de Programación. Fuente: (Vargas,2006)

2.5.4. ALGORITMOS

Un algoritmo es una “sucesión finita de reglas elementales, regidas por una prescripción precisa y uniforme, que permite efectuar paso a paso, en un encadenamiento estricto y riguroso, ciertas operaciones de tipo ejecutable, con vistas a la resolución de los problemas pertenecientes a una misma clase” (Ifrah, 1998).

Asimismo, José Gairin, expresa que un algoritmo es: “una prescripción, una orden o sistema secuenciado de órdenes que encadena una serie de operaciones elementales que llevan desde los datos iniciales al resultado” (Gairin, 2002, pág. 83).

Unda (2004) señala que un algoritmo no es más que un conjunto de pasos para lograr el objetivo deseado (Unda, 2004). El mismo autor agrega que dichos pasos son lógicos y ordenados para solucionar un determinado problema. Del mismo modo, (Guerrero, 2002) comparte tales afirmaciones, pero aporta que los problemas pueden ser de índole matemático, comercial, entre otros.

En el mismo orden ideas, (Beekman, 1995) hace un aporte en la definición de algoritmo y al respecto señala que es un conjunto de procedimientos paso a paso que, una vez completos, resuelven un problema o llevan a cabo una tarea.

Por su parte, Rojas (2004) en su módulo instruccional de algoritmos, plantea la definición de algoritmos, como “una serie de pasos o instrucciones que se ejecutan en orden lógico para procesar las entradas (variables y/o constantes) y obtener la salida” (Rojas, 2004, pág. 5).

Asimismo, indica algunos elementos que hay que tomar en cuenta para la construcción de un algoritmo, estos son:

- Análisis de un problema: desglosar el mismo, identificando en su enunciado las salidas(s), entrada(s) para posteriormente plantear su proceso.
- Entradas: datos, información que aporta el problema, pueden ser variables y constantes.
- Proceso: acciones, actividades que se deben cumplir para transformar los datos de entrada para obtener los resultados. Esas acciones pueden ser fórmulas.
- Salidas: son los resultados que pide el problema, todo aquello que se debe calcular. Resolución de incógnitas.

- Variables: identificador o palabra que va a recibir o que potencialmente puede recibir valores diferentes, bien sea en la entrada, proceso o salida.
- Constantes: son valores fijos que se identifican en el enunciado del problema, no cambian en el proceso.
- Instrucción: orden o sentencia que implica una acción.

El autor indica que cuando se agregan condicionales en un algoritmo para obtener resultados, se le conoce como algoritmo con estructura alternativa. También es conocido como algoritmo con estructura de control de selección. Al respecto, Beekman (1995) manifiesta que una estructura de control de selección o decisión se usa para tomar decisiones lógicas, eso significa que se eligen entre acciones alternativas dependiendo de ciertas condiciones. Por lo general, tiene la forma “Si (cierta condición es verdadera) entonces (hacer algo), en caso contrario (hacer otra cosa)” (Beekman, 1995).

La enseñanza de algoritmos es nueva para la mayoría de los estudiantes que ingresan a la educación superior, por tal razón (Roa, 2001) reconoce la necesidad de realizar diferentes actividades previas con materiales de distinto nivel de concreción. El autor considera adecuado iniciar el aprendizaje de los algoritmos con la manipulación de objetos concretos que permitan su asimilación hasta pasar a niveles más complejos.

2.5.5. EL PARADIGMA ESTRUCTURADO

Los paradigmas a lo largo de la historia, dieron lugar al nacimiento de distintas filosofías para modelar el mundo real. Los verdaderos e importantes cambios están más del lado filosófico: cómo transitar del dominio del problema al dominio de la solución.

Se ha utilizado como conjunto de reglas y principios que rigen una forma de pensar y de actuar. Los paradigmas no son conceptos propiamente del mundo de la computación, más bien éste los ha adoptado para poder expresar sus propias formas, por supuesto, de pensamiento y acción. Siendo un paradigma un aspecto de corte filosófico más que tecnológico no tiene fronteras fijas de nacimiento y muerte, de hecho, a lo largo del tiempo

se ha venido solapando un paradigma sobre otro cambiando paulatinamente, más en un proceso evolutivo que revolucionario. (Rodríguez, 2011).

El paradigma estructurado cuenta con su propio conjunto de reglas y principios gestados a partir de las teorías de programación, basado en el teorema de la estructura de Böhm y Jacopini de 1966 y la publicación de los trabajos de Dijkstra y Hoare, ratificados después por Harla D. Mills (Alcalde & Garcia, 1992), seguidos posteriormente por Yourdon y Constantine (Yourdon & Constantine, 1979) y otros autores.

Alcalde (1992), señala que la Programación Estructurada utiliza todas las estructuras de datos, tanto simples como complejas, que se han desarrollado hasta el momento y que propone tres conceptos fundamentales: Una metodología formal de diseño basada en la descomposición funcional descendente (top-down). Un conjunto de recursos abstractos basados en la modularidad y las características de la misma (cohesión y acoplamiento) y un conjunto de estructuras básicas de programación. (Alcalde & Garcia, 1992)

El proceso de descomposición funcional se basa en el principio básico de la división del problema en subtareas y las subtareas en subsubtarear y así sucesivamente hasta identificar tareas concretas y específicas. Este proceso repetitivo de descomposición funcional se le llama refinamientos sucesivos o simplemente refinamiento. El modelado resultante de la descomposición funcional es un diagrama muy similar al de un organigrama. Como el proceso se realiza de tareas abstractas o también llamadas de alto nivel, hasta tareas concretas o también llamadas de bajo nivel, se dice que el proceso se hace de arriba hacia abajo top-down en inglés aunque también existe la posibilidad de hacerlo de abajo hacia arriba bottom-up, si resulta conveniente para el proyecto.

El concepto de abstracción supone, en cada descomposición que todas las partes identificadas están resueltas, dejando su realización para el siguiente refinamiento y considerando que todas ellas pueden llegar a estar definidas en instrucciones y estructuras disponibles en los lenguajes de programación.

Los módulos resultantes del proceso de refinamiento tienen dos características: cohesión y acoplamiento. La cohesión se refiere al grado de independencia de un módulo para con los otros; es decir, se busca que un módulo pueda realizar una tarea por sí mismo sin necesidad de utilizar rutinas de otros módulos. En la medida en que el módulo puede resolver la tarea por sí mismo se dice si tiene alta o baja cohesión. La alta cohesión es algo normalmente deseable. El acoplamiento es la medida de interconexión que tiene un módulo en particular con otros. Mientras menos comunicación exista entre un módulo con otro se dice que hay bajo acoplamiento. El bajo acoplamiento es algo normalmente deseable.

El conjunto de estructuras básicas se divide en estructuras de datos y en estructuras de control. Las de datos clasifican a todas aquellas que hasta el momento han sido creadas. Las de control se basan únicamente en tres tipos de estructuras primitivas:

Secuencia, repetición y condición (ver Figura 2.8), descartando totalmente las de salto condicional y salto in condicional del paradigma anterior (aunque algunos lenguajes por compatibilidad hacia atrás conservaron mecanismos de salto).

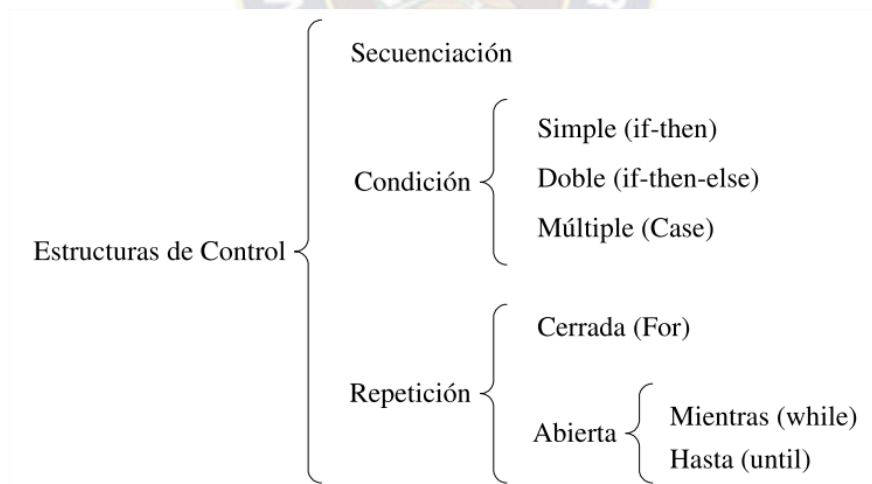


Figura 2.8 Estructuras de control para el paradigma estructurado

Las estructuras de repetición se dividen en dos categorías: cerradas y abiertas. Las de repetición cerradas son aquellas que señalan de antemano el número de repeticiones que

deben hacerse, por ejemplo: repite cinco veces este código; repite n veces este código teniendo n un valor conocido antes de ejecutar la instrucción. La implementación típica en un lenguaje es la instrucción *for*. Las de repeticiones abiertas son aquellas en las que se desconoce el número de iteraciones. Hay dos tipos: las repeticiones mientras se cumple una condición y las repeticiones hasta que se cumpla una condición. Las implementaciones típicas son *until* y *while*.

Las estructuras condicionales se dividen en tres tipos: simples, dobles y múltiples. Las simples son las que ejecutan una sección de código en caso de que una condición booleana se cumpla. Las dobles son aquellas que ejecutan una sección de código al cumplirse la condición booleana y otra sección diferente al no cumplirse. Las múltiples son aquellas que ejecutan diferentes secciones de código según el valor de una expresión. Las implementaciones típicas en los lenguajes son: *if-then*, *if-then-else* y *case*.

Considerando la secuencia, tres estructuras de repetición y tres de condición se forman siete estructuras de control básicas con las que se puede construir cualquier algoritmo.

2.5.6. REPRESENTACIÓN DE LOS ALGORITMOS MEDIANTE DIAGRAMAS DE FLUJO

Los Diagramas de Flujo (DF) corresponden a representaciones gráficas de acciones pre-establecidas, las cuales son situadas de manera secuencial con el fin de proveer transformaciones que permitan la obtención de resultados, dicha representación gráfica reduce algunas de las problemáticas presentadas por el lenguaje natural, como es el caso de la ambigüedad, el uso de expresiones no universales y la dificultad para la expresión escrita. Sin embargo, existen dificultades frente a la representación de estructuras de transformación complejas, llevando a que la construcción de algoritmos empleando lenguaje gráfico sea más perjudicial que benéfica si no se conocen las representaciones o convenciones para cada una de las acciones a realizar, sin contar que hasta este punto, tanto el lenguaje natural como el visual no son traducibles directamente a construcciones que permitan implantarse en sistemas informáticos sin requerir grandes cantidades de esfuerzo

en su transformación. La Figura 2.9 muestra una estructura algorítmica construida a partir de representaciones gráficas para el ejemplo del cálculo de área de un rectángulo, presentado con anterioridad (Toledo, 2002); (Joyanes, 2003).

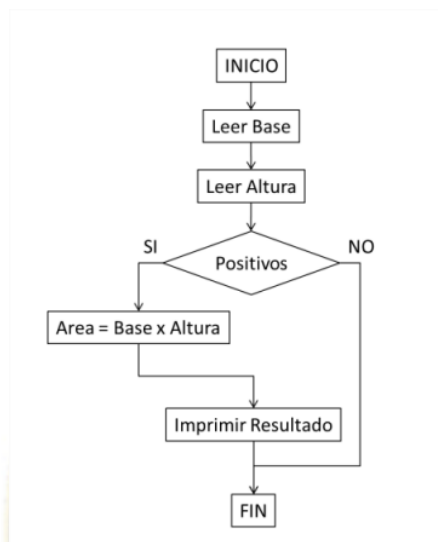


Figura 2.9 Representación de una descripción en Diagrama de Flujo para el algoritmo que permite calcular el área de un rectángulo. Fuente: (Toledo, 2002)

2.5.7. PRUEBAS DE ESCRITORIO PARA LA VERIFICACIÓN DE RESULTADOS

Hasta este punto se ha tomado como premisa la existencia de una definición del problema, de la cual se identifican sus aspectos más relevantes, facilitando la estructuración de un conjunto de pasos que conllevarían a una solución adecuada y su representación algorítmica. El algoritmo es el citado resultado de este proceso de construcción, y conforme lo describe (Harel, 2004) en los algoritmos se presenta un comportamiento constante en la medida que sus entradas de datos son constantes y legales, es decir, que mientras no exista variación en las características de las entradas asignadas a los problemas algorítmicos, y cumplan con las características esperadas para alimentar el algoritmo, sus resultados son invariantes independientemente del número de ocasiones en que sean ejecutados. Puede entonces decirse que un problema ha sido resuelto en la medida que se ha construido el algoritmo que brinde una solución genérica y adecuada al problema durante cada ejecución

del proceso, bajo la condición que se brinden entradas pertenecientes a un conjunto considerado permitido o legal.

Sin embargo, el comportamiento resultante del algoritmo puede no ser el esperado, bien sea por que la definición del problema sea ambigua o errónea, porque la estructuración de sus pasos no brinda una solución adecuada al problema, porque las estructuras de control que le conforman tienen algún tipo de desvío en su secuencia normal o por que los datos que le han sido ingresados pueden considerarse incorrectos o ilegales. De cualquier modo, puede entenderse que es interés y responsabilidad del problema algorítmico brindar tratamiento a cada una de estas vicisitudes con el fin de conservar su estructura y evitar resultados no contemplados o incorrectos, de acuerdo a lo manifestado por autores como (Bond, 2007) y (Meinke, 2004), entre otros.

Pero en este nivel surgen inquietudes diversas, por ejemplo: ¿Cómo es posible que un algoritmo no haga correctamente aquello para lo cual ha sido construido? Y si el algoritmo constituye una construcción mental orientada a la solución de un problema ¿Cómo esperar que tenga un comportamiento diferente al definido en su construcción y que sus resultados no sean los esperados? Si bien los algoritmos reflejan la manera en que su creador identifica las características de un problema, es él quien construye alrededor de la solución una estructura mental que le soporte, pero en pocas ocasiones, dicha estructura mental toma en consideración las diferentes variables que pueden ocasionar comportamientos no controlados o arrojar resultados no válidos.

Podría decirse que la estructuración del algoritmo se orienta a que haga lo que tiene que hacer, pero en ningún momento a validar que no haga lo que no tiene que hacer. Es en este sentido que la verificación de un algoritmo debe ser concebida como un proceso a realizarse en diferentes etapas, permitiendo generar la trazabilidad de la solución desde el momento de su concepción hasta llegar a su aplicación en la vida diaria mediante su implementación en un lenguaje de programación.

La primera etapa en la verificación de un algoritmo corresponde a su “seguimiento mental” en el momento mismo de su concepción, haciendo un recorrido a través de su colección de pasos e identificando posibles inconsistencias en su estructura, de manera que permita evidenciar su idoneidad en la solución al problema planteado. En la (Figura 2.10) puede observarse que este tipo de verificación, si bien es simple en su aplicación, se hace compleja en la medida que el algoritmo se hace más denso, ya que el manejo de diversas características y sus transformaciones pueden estar más allá de la capacidad mental de quien lo construye. Así, el incremento en la complejidad limita el seguimiento mental, surgiendo la verificación a través de un registro escrito o “prueba de escritorio”.

n	i	f
0	1	1
		Salida

Figura 2.10 Prueba de escritorio. Fuente: Elaboración propia 2017

2.5.8. CODIFICACIÓN

La teoría de la programación será considerada como el conjunto de principios que rigen un paradigma de programación y definen las características semánticas de cierto conjunto de lenguajes.

Algunos autores realizan una diferenciación entre el concepto de programar y el de codificar señalando lo siguiente: “Por programar se entiende un proceso mental complejo, dividido en varias etapas. La finalidad de la programación, así entendida, es comprender con claridad el problema que va a resolverse o simularse por medio de la computadora, y

entender también con detalle cuál será el procedimiento mediante el cual la máquina llegará a la solución deseada” (Levine, 1989).

“La codificación constituye una etapa necesariamente posterior a la programación, y consiste en describir, en el lenguaje de programación adecuado, la solución ya encontrada, o sugerida, por medio de la programación. Es decir, primero se programa la solución de un problema y después hay que traducirla a la computadora.” (Levine, 1989).

Para efectos de este trabajo se utilizará un concepto de programación amplio basado en el planteamiento de Levine que incluye el proceso de identificación del problema, el diseño algorítmico de la solución, la codificación en algún lenguaje y las pruebas correspondiente.

La teoría de la programación aspira a dar certidumbre sobre las formas más adecuadas de enfrentar un problema, modelarlo y encontrar su solución algorítmica para poder codificarlo en algún lenguaje y ejecutar la solución en una computadora.

2.6. PROGRAMACIÓN COMPETITIVA

La programación competitiva es una metodología que, como su nombre lo indica, tiene el fin de crear ambientes que permitan a los estudiantes competir sanamente entre sí, mediante la resolución de ejercicios de programación bajo los estándares de la ACM-ICPC (Revilla, Manzoor, & Liu, 2008).

Según (Halim & Halim, 2010) el principal objetivo de esta práctica es la de educar a los estudiantes para que sean excelentes ingenieros de software que utilicen el estado del arte de las estructuras de datos y algoritmos, de modo que puedan encontrar las mejores soluciones para cada uno de los problemas que surgen actualmente en el mundo.

Actualmente hay varias competencias que se llevan a cabo a nivel mundial haciendo uso de esta práctica con una gran variedad de ejercicios. Sin embargo, el objetivo de cada competencia puede variar entre cada uno, desde probar el conocimiento de los

participantes, a mejorar el aprendizaje, promover la informática y encontrar competidores talentosos (Hakulinen, 2011).

2.6.1. COMPETENCIAS DE PROGRAMACIÓN

La mayoría de las competencias de programación son anuales, pero algunas son continuas y permiten participar en cualquier momento de forma online. En muchas de estas competencias, una vez finalizadas se puede acceder a las soluciones de otros programadores, lo cual contribuye al proceso educativo y permite mejorar las habilidades del programador.

Existen dos tipos de competencias académicas muy conocidas a nivel mundial, la primera es el IOI o (International Olympiad in Informatics), es una olimpiada a nivel internacional de programación para estudiantes de bachillerato. Esta competencia es a nivel individual y se lleva a cabo desde 1989 con el objetivo es promover las ciencias de la computación entre los jóvenes y descubrir y estimular a estudiantes talentosos. La segunda competencia corresponde a la ICPC (International Collegiate Programming Contest), una competencia para estudiantes universitarios, en los cuales se debe conformar grupos de 3 personas con el fin de resolver una serie de ejercicios que varían de dificultad haciendo uso de un solo computador (Verhoeff, 2009).

Existen muchas otras competencias que son llevadas a cabo por internet, en donde cientos de participantes a nivel mundial pueden participar, en la Tabla 2.4 se presentan algunas de las más conocidas.

Tabla 2.4
Cuadro de competencias de programación

Competencia	Descripción
Top Coder	Es la plataforma con el mayor número de usuarios registrados a nivel mundial (alrededor de 634.000), permite participar semanalmente en competencias de programación y diseño de software con estudiantes de todas las partes del mundo. Su objetivo principal es el de ayudar a los

	estudiantes a mejorar sus habilidades, y ganar experiencia en el desarrollo de software para proveer a la industria de los más talentosos programadores que resuelvan los problemas de las tecnologías emergentes.
Google Code Jam	Es una competencia anual realizada por Google, en donde el objetivo principal es resolver problemas de programación en una cantidad de tiempo determinado. Su objetivo principal es identificar estudiantes talentosos que sean potenciales empleados de Google.
Facebook HackerCup	Es una competencia anual realizada por la red social, en donde se asignan un número determinados de ejercicios para ser resueltos en 24 horas. Consta de 3 etapas, hasta que los primeros 25 participantes se presentan en la final en las oficinas de Facebook.
Imagine Cup	Es una competencia anual realizada por Microsoft donde se asigna un problema relacionado con alguna problemática a nivel mundial, cada equipo deberá implementar una solución innovadora usando tecnologías Microsoft.

Nota. Fuente: Elaboración propia 2017

Aunque no existe un estándar acerca de los tipos de problemas y dinámicas de la programación competitiva, la gran mayoría de las competencias siguen la dinámica mostrada en la (Figura 2.11).

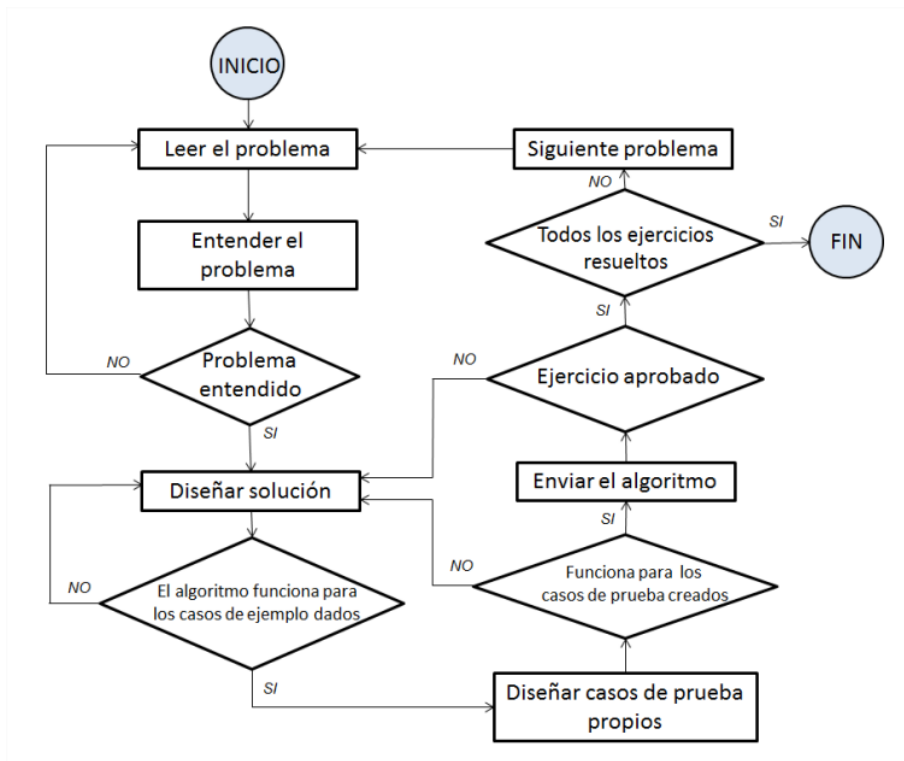


Figura 2.11 Representación de una descripción en Diagrama de Flujo para resolver problemas de programación. Fuente: (Pienda, 2014)

Ya respecto a la naturaleza de los problemas como tal y sobre todo de su utilización en aulas de clase, la misma puede ser muy variada; (Ragonis, 2012) por ejemplo, propone doce tipos diferentes de preguntas que podrían ser usadas por los docentes para llevar a cabo en los cursos relacionadas con las ciencias de la computación. A partir de dichos tipos (Hakulinen, 2011) resume las tres metodologías que son las más utilizadas en las competencias a nivel mundial las cuales son:

- **Problemas sin uso de programación:** Este tipo de competencias hacen uso solo de lápiz y papel, en este tipo de competencias se busca evaluar los conocimientos informáticos de los estudiantes relacionados con la programación, según (Burton, 2010) este tipo de competencias pueden atraer a más estudiantes de otras carreras que no necesariamente están relacionados con la programación pero que están interesados en la temática especialmente estudiantes de matemáticas y física.

Generalmente este tipo de competencias están basados en selección múltiple o seguir un algoritmo en donde su salida es un número entero (Kubica & Radoszewski, 2010).

- **Problemas con uso de programación:** La gran mayoría de las competencias en informática conllevan a programar. En este tipo de problemas los competidores generalmente deben realizar un código que resuelva un cierto problema, normalmente los problemas deben ‘pasar’ o superar unos casos de pruebas, con este tipo de problemas es posible brindar a los participantes una retroalimentación inmediata y permitirles el envío de más soluciones hasta que esta sea correcta (Hakulinen, 2011).
- **Comprensión de código:** Otro tipo de competencia es el uso de problemas que requieran de habilidades de lectura de códigos. En la vida real, los programadores deben muy a menudo tener que leer código fuente creado por otras personas. En este caso el participante debe entender el código fuente con el fin de analizarlo y mejorarlo. Generalmente a los competidores se les asigna un código con el fin de encontrar el error, construir un contraejemplo, estimar la complejidad, o implementar una versión más eficiente del mismo algoritmo.

2.6.2. JUECES EN LÍNEA

En la actualidad las Tecnologías de la Información y las Comunicaciones (TIC) han hecho impacto en todas las áreas del saber. Con el auge de las competencias de programación, se ha hecho necesario aumentar la preparación de los competidores para obtener mejores resultados. En este proceso de preparación han surgido los jueces en línea, que no son más que una herramienta que permiten a los competidores probar sus habilidades e intercambiar experiencias con otros competidores de cualquier región del orbe a través de la red de redes.

Los jueces en línea por lo general son aplicaciones Web que permite evaluar programas de computación que intentan solucionar tareas propuestas. Estos sistemas son capaces de compilar y ejecutar el código fuente de la solución propuesta, y validarlo con los juegos de

datos definidos para la tarea que se intenta resolver. A continuación, se describen algunos de los jueces en líneas internacionales y nacionales de mayor popularidad.

2.6.2.1. JUEZ EN LÍNEA DE LA UNIVERSIDAD DE VALLADOLID (UVA)

UVA¹⁸ es uno de los jueces en línea más antiguos y prestigiosos del mundo; surge en el año 1995, pero comienza a cobrar fuerzas en la comunidad internacional en el año 1997. Se emplea como herramienta de entrenamiento para concursos de programación y posee uno de los mejores archivos de problemas del mundo.

Dispone de varios módulos que brindan una mayor funcionalidad al juez en línea. El módulo de estadísticas permite acceder a la información detallada de los usuarios, problemas y concursos. El archivo 24 horas permite a los usuarios probar sus soluciones en tiempo real durante todo el día. Hay que destacar que UVA a pesar de ser un juez en línea está integrado con la plataforma EDUJUDGE¹⁹ lo cual lo vincula al proceso de enseñanza-aprendizaje de la programación competitiva.

2.6.2.1. SPHERE ONLINE JUDGE (SPOJ)

El SPOJ²⁰ (Bolton, 2007) es un juez en línea desarrollado por el proyecto Sphere Research Labs. Se caracteriza por tener una interfaz intuitiva, sencilla y ligera, propiciando a usuarios nuevos e inexpertos la fácil comprensión del propósito de la herramienta. Permite hacer diferenciación de problemas en cuanto a dificultad y propósito. Tiene soporte para más de 40 lenguajes de programación. Posee tiempos de evaluación muy rápidos, a pesar de las pocas prestaciones del servidor sobre el cual está desplegado. Uno de los elementos más destacables del SPOJ es el empleo de MOSS (Similitud masiva de software en línea), como herramienta de detección de plagios, y la posibilidad brindada a determinados usuarios de organizar concursos bajo sus propias reglas y problemas. El sitio carece de referencias a

¹⁸ UVA <https://uva.onlinejudge.org>

¹⁹ EDUJUDGE es un sistema que surge, por una parte, de la necesidad para mejorar las competencias matemáticas, científicas y tecnológicas a nivel europeo, como muestran las políticas europeas en materia de Educación y Formación y, por otra parte, del éxito de los concursos de programación a nivel Europeo e internacional en la educación secundaria y superior.

²⁰ SPOJ <http://www.spoj.com>

documentación o tutoriales destinados a la preparación de los usuarios en los temas relacionados con la programación competitiva.

2.6.3. SISTEMAS DE ENTRENAMIENTO PARA PROGRAMACIÓN COMPETITIVA

Aunque los jueces en línea han sido muy empleados como medios de entrenamiento y preparación, no cumplen todos los requisitos necesarios para esta función, pues de un principio los competidores deben tener un conocimiento básico de programación competitiva para poder practicar. Además, los jueces en línea no permiten que un entrenador asista a un estudiante de forma virtual, y pueda darle un seguimiento a la evolución de su entrenamiento. También hay que destacar que no todos los competidores deben tener el mismo entrenamiento pues las características y niveles cognitivos son diferentes en cada competidor.

A continuación, se hace una breve introducción a los sistemas de entrenamientos que son empleados en la preparación de los competidores con el objetivo de obtener mejores resultados en competencias de programación.

2.6.3.1. CODEFORCES

Codeforces²¹ es uno de los jueces en línea más populares, surgido en el año 2010 cambiando la dinámica de los jueces en línea existentes hasta el momento. Se caracteriza por ser uno de los pocos jueces en línea que aplica la Web 2.0 (Arnal, 2007) y constituir un sistema predominantemente para la realización de competencias de programación en varios estilos existentes. Para incrementar la preparación de los concursantes Codeforces permite clasificar los problemas por áreas temáticas. También permite el acceso a los códigos fuentes de las soluciones una vez finalizados los concursos.

²¹ CODEFORCES <http://codeforces.com>

2.6.3.2. USA COMPUTING OLYMPIAD (USACO)

USACO ²²según (Park, 2013) es una plataforma dedicada a la preparación, información y apoyo a los competidores de IOI. Está compuesta por varias subplataformas, entre las cuales destacan: USACO Contest y USACO Training. La primera permite realizar competencias de programación al estilo IOI, incluyendo el mismo sistema de puntuación. Por otra parte, USACO Training está destinada a la preparación de los competidores, con una serie de problemas que se organizan por secciones y capítulos. Los usuarios deben ir venciendo los problemas de cada una de las secciones para poder pasar a la próxima sección de entrenamiento. El entrenamiento es igual para todos los competidores sin diferenciar niveles de preparación.

2.7. CURRÍCULOS EDUCATIVOS PARA LA ENSEÑANZA-APRENDIZAJE DE PROGRAMACIÓN.

En este epígrafe se realiza un breve análisis comparativo sobre la situación del ‘coding’ en los currículos de los sistemas educativos de la Unión Europea. Para ello, se considera el reciente informe de la Comisión Europea (octubre de 2015), denominado “Computing our future. Computer programming and coding: Priorities, School curricula and initiatives across Europe” (European Schoolnet, 2015). En la exposición de motivos de dicho informe, se dice:

“Las competencias y habilidades digitales son una de las principales condiciones para que la transformación digital de Europa sea un éxito, así como para su crecimiento y el bienestar de sus ciudadanos y sociedades (...) El reto para el sector educativo es elevar el nivel de dichas habilidades digitales en la futura fuerza de trabajo; pero, aún más importante, empoderar a la gente joven con competencias que les permitan dominar y crear sus propias tecnologías digitales, y prosperar en la sociedad actual. Creemos que la enseñanza-aprendizaje del ‘coding’, tanto en contextos formales como no formales, jugará un papel fundamental en este proceso (...) Muchos

²² USACO <http://train.usaco.org/usacogate>

educadores, así como padres, economistas y políticos en Europa y el resto del mundo, comienzan a pensar que los estudiantes necesitan unas mínimas habilidades de computación y programación. Una razón es la escasez de profesionales empleables con habilidades digitales de orden superior. Para el año 2020 se calcula que Europa podría sufrir un déficit de alrededor de 800.000 profesionales capacitados en computación y programación. Otra razón importante es que el ‘coding’ ayuda a entender la sociedad digitalizada actual y fomenta habilidades fundamentales del siglo XXI como la resolución de problemas, la creatividad, y el pensamiento lógico” (European Schoolnet, 2015, p. 4-6).

Los resultados que presenta el informe se basan en una encuesta realizada a 21 Ministerios de Educación (correspondientes a 20 países europeos, más Israel), en la cual se les interroga sobre la situación de la programación en sus respectivos sistemas educativos. Los países participantes en el estudio son, por orden alfabético: Austria, Bélgica-Flandes, Bélgica-Valonia, Bulgaria, Dinamarca, Eslovaquia, España, Estonia, Finlandia, Francia, Hungría, Irlanda, Israel, Lituania, Malta, Noruega, Países Bajos, Polonia, Portugal, Reino Unido y República Checa.

Siendo 16 países los cuales incluyen la programación en su currículum, ya sea a nivel nacional, regional o local: Austria, Bulgaria, Dinamarca, Eslovaquia, España, Estonia, Francia, Hungría, Irlanda, Israel, Lituania, Malta, Polonia, Portugal, Reino Unido y República Checa.

La Tabla 2.5, contiene un cumulo de experiencias entorno a la enseñanza de programación a niños y joven, contenidos a ser considerados.

Tabla 2.5

Cuadro comparativo de experiencias de la enseñanza de la programación a niños y jóvenes

Autores y año	País o países de aplicación	Herramienta seleccionada	Temática presentada y contexto en el que aplica	Población objetivo (estudiantes) y presaberes	Técnica aplicada	Modelo pedagógico aplicado
---------------	-----------------------------	--------------------------	---	---	------------------	----------------------------

Abramovich, 2013	Austria, Canadá, Rep. Checa, Israel, Serbia, EUA	Utilitarios de matemáticas: GeoGebra, VisualMath, Maple	Matemáticas; contexto dependiente del tema	Diversos niveles educativos; con conocimientos previos de matemáticas	Resolución de problemas según el método Polya	Ninguno propuesto
Liu, Cheng y Huang, 2011	Taiwán	Juego de simulación Train B&P	Programación; contexto sobre control de transporte de un sistema de ferrocarril	Primer año de universidad; sin experiencia en programación	Diseño experimental, simulación y generación de hipótesis, interpretación de datos, resolución de problemas según el método Polya	Prueba y error, aprendizaje por ejemplo, razonamiento, aprendizaje por construcción
Valente, 2004	Dinamarca	Cartas computacionales	Teoría de las ciencias de la computación y la información; contexto sobre circuitos básicos y dinámicos, problemas de probabilidad y transmisión de información	Primaria, de 8 a 10 años; sin experiencia en programación, incluso con limitación de formación matemática	Armado de circuitos de flujo (ubicación de cartas) y aplicación de reglas formales, armado de circuitos especiales para reutilización	Manipulación directa y aprendizaje por realización (construcción), desarrollo de proyectos top-down y bottom-up
Felleisen, Fidler, Flatt, y Krishnamurthi, 2009	EUA	Entorno de desarrollo DrScheme	Matemática plana, álgebra y programación funcional; contextos diferentes para crear simulaciones, animaciones y juegos	Bachillerato, de 10 a 14 años; con presaberes de matemática básica	Programación funcional mediante modelos matemáticos	Modelamiento matemático
Wolz, Leitner, Malan, y Maloney, 2009	EUA	Entorno de desarrollo Scratch	Programación básica y avanzada (eventos, concurrencia, hilos); contexto diferente de acuerdo con la temática	Primer semestre universidad, con conocimiento nulo o muy básico en programación	Enseñanza de programación mediante armado, transición a lenguaje de programación real	Aprendizaje por construcción
Tomcsányiová y Tomcsányi, 2011	Eslovaquia	Cada tarea bajo un applet	Uso de TIC para comunicación, diagramas de flujo, resolución de problemas, pensamiento algorítmico, principios de TIC, sociedad de la información; varios contextos con diferentes niveles de dificultad	Primaria, estudiantes de 2° a 4° (7-9 años), con habilidad mental	Desarrollo de torneo competitivo sobre resolución de tareas interesantes y evaluación de resultados	Competencia y evaluación. Trabajo con objetos concretos sin uso de abstracción. Clasificación, adecuación, Formulación de tareas
Doerschuk, Liu y Mann, 2012	EUA	Plataforma GreenFoot	Conceptos básicos de programación, POO; contexto para programación de juegos	Jóvenes de grados 10° y 11°; ningún conocimiento sobre programación	Campamento de programación intensivo	Enseñanza de material instruccional preparado por estudiantes de niveles superiores

Rogozhkina y Kushnirenko, 2011	Rusia	Entorno de desarrollo Scratch	Conceptos fundamentales de programación; contexto sobre juego de control de robot para llenado de casillas	Educ. infantil (5.5 a 7 años); manejo de mouse, ninguna o poca habilidad de lectura	Juego para cumplimiento de objetivos, sin uso y con uso de computador	Instrucción básica, pruebas escritas gráficas y aplicadas en computador
Burke y Kafai, 2010	EUA	Entorno de desarrollo Scratch	Programación básica, orientación a objetos. Contexto: creación de historias	Estudiantes de educación media, entre 10 y 14 años	Programación básica para narración de historias y creación de juegos	Generación de sentido narrativo secuencial, capacidad de escritura
Javidi y Sheybani, 2009	EUA	Entornos de programación Kahootz y Squeak	Programación para videojuegos. Contexto: temas sobre conservación del medio ambiente y salud.	Estudiantes de educación media de áreas urbanas y rurales	Diseño de videojuegos, creación de ideas con historias, preparación para Liga Lego	Trabajo en equipo, exploración de software instruccional, autorreflexión
Kelleher, Pausch, y Kiesler, 2007	EUA	Entorno de desarrollo Storytelling Alice	Programación básica. Contexto: creación de historias	Estudiantes mujeres de educación básica y media (5° a 9°)	Enseñanza de herramientas y conceptos de programación, quiz de conceptos y actitudes, construcción de historias	Enseñanza a través de tutorial, uso de constructos de programación básica
Lee y Ko, 2011	EUA	Juego de programación Gidget	Programación básica, sintaxis para programación. Contexto: juego para ayudar a un robot a corregir sus fallas de programación para cumplir misiones	Personas desde bachillerato hasta doctorado alrededor del mundo, sin conocimiento previo de programación	Enseñanza de comandos básicos, composición de comandos más complejos, comprensión de errores	Enseñanza a través de tutorial, presentación de variaciones sintácticas con errores. Aprendizaje por error
Meyers, Cole, Korth y Pluta, 2009	EUA	Entorno de programación Processing	Programación básica y avanzada (funciones, gráficos y fractales), procesamiento de sonidos. Contexto: análisis y generación de melodías e imágenes	Estudiantes entre 12 y 17 años, con preferencia en matemáticas y conocimiento en música	Enseñanza de programación y música computacional. Representación de imágenes, manipulación y sintetización de sonidos en forma electrónica	Construcción de imágenes y música
Rizvi et al., 2011	EUA	Entorno de programación Scratch	Programación básica. Implementación de proyectos. Contexto: creación de proyecto multimedia con documentación	Estudiantes recién ingresados a universidad	Curso de programación utilizando Scratch, explicación de constructos básicos y avanzados de programación	Aprendizaje tradicional con uso de TIC
Rodger et al., 2010	EUA	Entorno de programación Alice	Matemáticas, música, programación. Contexto: uso de videojuegos para diferentes contextos dependiendo de la temática	Todos los niveles educativos, desde primaria hasta universidad	Implementación de proyectos 3D de tipo videojuegos, para aplicar en matemáticas, programación y música	Aprendizaje tradicional con uso de TIC

Sipitakiat y Nusen, 2012	Tailandia	Sistema de construcción por bloques físicos Robo-Blocks	Programación para robótica. Contexto: creación de minirobots	Niños entre 8 y 9 años	Creación de robots mediante construcción de bloques. Enseñanza de modelo de depuración	Aprendizaje por construcción
Tarkan et al., 2010	EUA	Lenguaje de programación para niños Toque	Diseño de órdenes secuenciales. Contexto: Escenarios de cocina para preparación de recetas	Niños entre 7 y 11 años	Proceso de diseño e implementación de recetas de cocina	Aprendizaje por construcción
Zuckerman, Arida y Resnick, 2005.	EUA	Bloques de construcción electrónicos FlowBlocks y SystemBlocks	Conceptos matemáticos concretos y abstractos. Contexto: construcción de flujos para diversos escenarios	Niños entre 4 y 11 años	Estructuras genéricas comparadas con objetos reales, nivel de abstracción, asociación semántica, analogías	Manipulativos digitales inspirados en la metodología Montessori

Nota. Fuente: (Palma & Sarmiento, 2015)

2.8. ALGORITMOS Y PROGRAMACIÓN EN LA EDUCACIÓN ESCOLAR

Existe actualmente, dentro de la comunidad educativa, un consenso general a nivel mundial sobre la necesidad de superar el tipo de enseñanza basada en la transmisión de contenidos y reemplazarla por desarrollar capacidades. Investigaciones y estudios recientes proponen diversos conjuntos de habilidades que la educación debe fomentar para que los estudiantes puedan tener éxito en el mundo digital y globalizado en el que van a vivir. Este planteamiento exige, sin dilaciones, implementar estrategias que contribuyan efectivamente al desarrollo de esas habilidades consideradas como fundamentales para la educación en el Siglo XXI (21stcenturyskills, 2017). En la mayoría de conjuntos de habilidades propuestos figura el desarrollo de la destreza para solucionar problemas; por esta razón, se requiere seleccionar estrategias efectivas para ayudar a que los estudiantes la adquieran. Para atender esta necesidad, el uso de lenguajes de programación constituye una alternativa efectiva pues compromete a los estudiantes en la consideración de diversos aspectos importantes para solucionar problemas, tales como: decidir sobre la naturaleza del problema, seleccionar una representación que ayude a resolverlo y monitorear tanto sus propios pensamientos (metacognición) como las estrategias de solución utilizadas, aspectos estos que deben desarrollarse desde la edad temprana.

El desarrollo de conseguir habilidades heurísticas en los estudiantes es una competencia compleja de “alto nivel” relacionada con un modelo de conceptualización específica de los seres humanos que desarrolla ideas y vinculada con el pensamiento abstracto-matemático y con el pragmático-ingenieril que se aplica en múltiples aspectos de nuestra vida diaria. Por lo cual el pensamiento computacional no es sinónimo de capacidad para programar un ordenador, puesto que requiere pensar en diferentes niveles de abstracción y es independiente de los dispositivos. Se puede desarrollar pensamiento computacional sin utilizar ordenadores (basta papel y lápiz), si bien los dispositivos digitales nos permiten abordar problemas que sin ellos no nos atreveríamos a enfrentar. Por otra parte, es una competencia básica que todo ciudadano debería conocer para desenvolverse en la sociedad digital, pero no es una habilidad «rutinaria» o «mecánica», ya que es una forma de resolver problemas de manera inteligente e imaginativa (calidades humanas que no poseen los ordenadores). Además, posee la característica de combinar abstracción y pragmatismo, puesto que se fundamenta en las Matemáticas, un mundo de ideas, y se desarrolla a partir de proyectos de ingeniería que interactúan con el mundo real. Los conceptos computacionales se utilizan para enfocar y resolver problemas reales, comunicarnos con otras personas y gestionar múltiples aspectos de nuestra vida cotidiana (Wing, 2006).

CAPÍTULO III MÉTODO CONECTIVO BAJO PRESIÓN V-BLOOM

3.1. EL MÉTODO CONECTIVO BAJO PRESIÓN V-BLOOM

La búsqueda para mejorar los procesos de enseñanza aprendizaje deriva en el planteamiento de diversos métodos. El cambio de época, obliga a la incorporación de tecnología en dichos métodos de enseñanza aprendizaje, (Area, 2007) afirma que no sólo es importante emplear ordenadores y demás artilugios digitales, sino que los métodos debieran responder a ciertos principios y criterios científicos.

En tal sentido, el propósito de este capítulo plantear las características correspondientes del denominado método conectivo bajo presión V-Bloom, el cual será evaluado posteriormente en cada una de las etapas de la Olimpiada Boliviana de Informática 2017.

Para lograr este propósito, se consideró todos los aspectos conceptuales propios de la teoría del aprendizaje y programación, realizando el análisis para conseguir conceptualizar por completo el método a ser aplicado.

3.2. CONCEPTUALIZACIÓN DEL MÉTODO

Los métodos didácticos o de enseñanza, siguen un enfoque científico o “estilo educativo” consistente en lograr la mayor efectividad posible en el proceso de aprendizaje de los estudiantes. Como se vio anteriormente, integra un conjunto de principios, una descripción de la praxis, actividades y normalmente el sistema de evaluación. La elección del método o métodos de enseñanza que se utilizará depende en gran parte de la información o habilidad que se está enseñando, en este caso de investigación se trabajará las habilidades cognitivas, las cuales serán afectadas por el contenido de aprendizaje y el nivel de los estudiantes. Un esquema se propone en la **¡Error! No se encuentra el origen de la referencia.** y resumen en la Tabla 3.3.

A continuación, se procede a la conceptualización del método experimental.

3.2.1. ENFOQUE

El método Conectivo Bajo Presión V-Bloom integra las siguientes teorías de aprendizaje:

- **Constructivismo Social.** El estudiante deberá ser activo en la construcción de su conocimiento, capaz de mejorar su autonomía además de interactuar con sus compañeros para consolidar su aprendizaje. El profesor deberá ser mediador del conocimiento, capaz de desarrollar habilidades cognitivas en el estudiante optimizando su proceso de razonamiento, para luego transferir gradualmente la responsabilidad del aprendizaje al estudiante conjuntamente deberá generar espacios de interacción con entre los miembros de su comunidad de estudiantes.
- **Conectivismo.** El estudiante deberá conectarse a diversas fuentes de conocimiento, siendo participe de comunidades participando en las mismas y compartiendo contenido, capaz de dirigir su proceso de aprendizaje. El profesor juega un papel de mediador, enseñando a los estudiantes a crear sus propias redes, mejorando sus formas de comunicación para pedir ayuda, brindando aprendizaje colaborativo virtual basado en tecnologías, a través de actividades aprovechando las TIC.

En tiempos modernos hace falta más de una teoría de aprendizaje para lograr la formación integral de los individuos, es acá donde el constructivismo, social constructivismo y conectivismo, se amalgaman ofreciendo una solución real y adaptativa al mundo cambiante. (Sobrino, 2011).

3.2.2. DISEÑO

Establecido el enfoque del método, prosigue el desarrollo de un diseño sistemático el cual considera:

- a. Objetivos del método.** Desarrollar habilidades propias de la programación competitiva en un tiempo óptimo.

b. Selección y organización el plan de estudios. El plan de estudios viene definido por la OCEPB en la Tabla 3.1.

Tabla 3.1
Contenidos temáticos a abordar

Resolución de problemas	Divisibilidad
Problemas lógicos	Aritmética Modular
Aritmética	Estrategia “descomposición funcional” (uso de funciones).
Álgebra	Construcción de Funciones y paso de parámetros
Definición de dato	Estructuras de datos estáticas:
Tipos de datos	Cadenas: funciones básicas con cadenas, concatenación, sub-cadenas
Constantes y Variables	Vectores (Arreglos): recorridos, impresión
Construcción de algoritmos	Matrices (Arreglos bidimensionales): recorridos, impresión
Expresiones y asignaciones	Búsquedas: búsqueda de caracteres en cadenas, búsqueda de sub-cadenas en cadenas.
Entrada y salida	Métodos de ordenación simples: por inserción, por selección.
Estructuras de control condicionales	Construcción de Tipos de Datos Abstractos. Por ejemplo, estructuras estáticas
Estructuras de control iterativas	(struct en C++), objetos.
Codificación de algoritmos en los lenguajes permitidos	Estructuras de datos dinámicas lineales:
Sintaxis básica y semántica	Vector
Entrada y salida de datos	Pila

Estructuras de control	Cola
Condicionales: if-then-else, switch	Manejo de estructuras con bibliotecas STLs
Iteración: for, while, do-while	Operaciones de bits: not, and, or, xor, left shift, right shift
Manejo de Series: aritméticas, geométricas, especiales	Teoría de números: factores primos, criba de Eratóstenes.
Operaciones con números: Descomposición de números, Máximo Común Divisor	Búsqueda de patrones en cadenas
(MCD), Mínimo Común Múltiplo (mcm), factorial.	Combinatoria: permutaciones, combinaciones, coeficiente binomial
Números Primos: identificar un número primo	Teoría de grafos: Representación de un grafo con listas de adyacencia, Recorrido por amplitud (BFS), Recorrido por profundidad (DFS).

Nota. Fuente: Convocatoria para el área de Informática de la 7ma OCEPB

La organización de actividades se la realizó a través de un Diagrama Gantt.

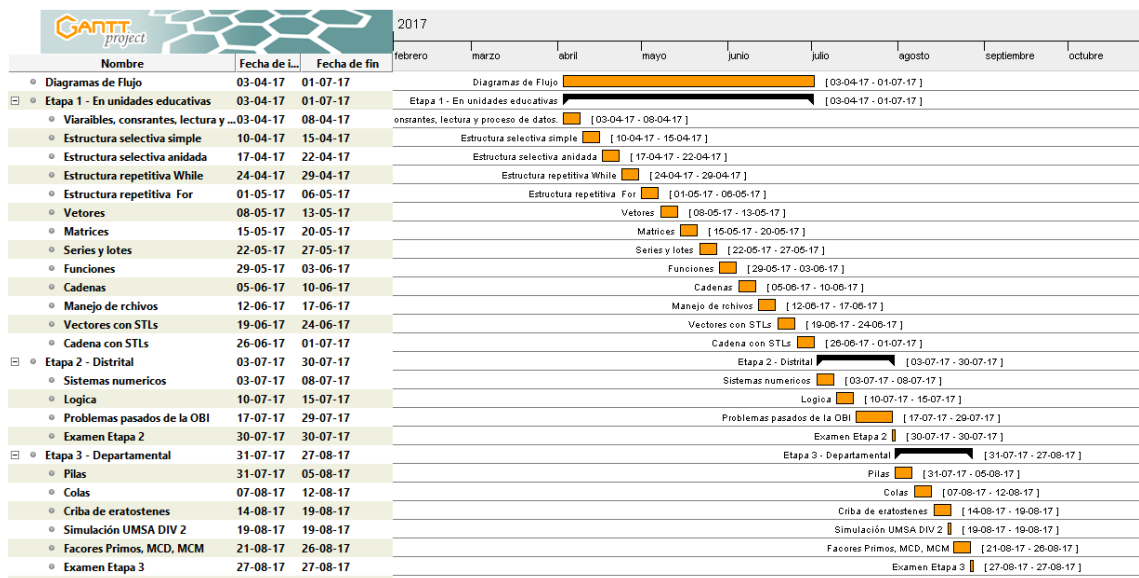


Figura 3.1 Cronograma de actividades. Fuente: Elaboración propia 2017

- c. Actividades de enseñanza y tareas de aprendizaje.** Las actividades de enseñanza son: diseño, adecuación, contextualización, de las actividades destinadas para promover la construcción autónoma y consciente del sistema cognitivo–instrumental de cada estudiante. Por lo tanto, las actividades de aprendizaje son diseñadas por el profesor las cuales son: prácticas y resolución de problemas lógico matemáticos, resolución de problemas algorítmicos, construcción de diagramas de flujo, pruebas de escritorio, desarrollo de problemas en C++, solución de problemas en juez virtual y generación de espacios de socialización virtual.
- d. Roles de los estudiantes.** Perseverante, proactivo, critico, social, empático, creativo, flexible y responsable.
- e. Roles de los profesores.** Desarrollador de espacios de interacción social, gestor de resultados, gestor de redes sociales y servicios de mensajería además de agente motivador.
- f. Materiales de instrucción.** Los materiales a usar: pizarra, textos físicos, computadoras, textos digitales, guía de ejercicios, presentaciones, video tutoriales, redes sociales (Facebook), servicios de mensajería (WhatsApp) y juez virtual.

3.2.3. PROCEDIMIENTOS

- a.** Inicialmente se procede al llenado del plan de clase del método (ver Tabla 3.2).

Al llenar la plantilla el profesor se compromete a responder el objetivo del método, cumplir con los temas y subtemas. Establecer objetivos para las dimensiones del individuo: Ser, Saber y Hacer. También se estaría ajustando el avance de contenido sincronizando con el cronograma, además de distribuir cada uno de los niveles del método en los momentos de la clase: Inicio, desarrollo y cierre. Conjuntamente cada nivel estaría vinculado a un recurso.

Tabla 3.2
Plantilla para la gestión de contenidos

PROFESOR:		AÑO:					
OBJETIVO DEL METODO:		FECHA:					
TEMA:		SEMANA:					
SUB TEMA:		N° DE ESTUDIANTES:					
OBJETIVO ACTITUDINAL:							
OBJETIVO CONGNITIVO:							
OBJETIVO CONDUCTUAL:							
MOMENTO DE LA CLASE		METODO V-BLOOM	ACIÓN DEL PROFESOR	ACIÓN DEL ESTUDIANTE	RECURSOS	TIEMPO	EVALUACIÓN
		BACKTRACKING					INICIAL
	INICIO	ETAPA 1					FORMACIÓN
		ETAPA 2					
	DESARROLLO	ETAPA 3					
		ETAPA 4					
		ETAPA5					
	CIERRE	ETAPA 6					
	ETAPA7					SUMATIVA	
PENDIENTES:							

Nota. Fuente: Elaboración propia 2017

b. Las formas en que las actividades de enseñanza particulares se utilizan para practicar.

Fase 1 - Recordar

Acción del profesor

El profesor deberá brindar un explicación clara y detallada de todo aquello que desea transmitir a sus estudiantes, tomando en cuenta el cronograma. Seleccionará los recursos más adecuado relacionado a la temática haciendo uso de decurso como presentaciones, libros, textos impresos, documentos PDF y videos digitales.

Acción del estudiante

En este primer nivel el estudiante deberá colocar toda la atención posible, comprender y recordad toda la teoría o principio en su forma original, terminologías (palabras, términos técnicos, etc.) y sintaxis. Podrá emplear apuntes, tomar fotografías, grabar el audio de clase o incluso grabar la explicación brindada por el profesor.

Fase 2 – Entender

Acción del profesor

El profesor dentro el aula deberá generar un espacio de reflexión y debate para realizar las aclaraciones correspondientes y despejar dudas, este paso requiere una mayor capacidad de pensamiento crítico. En el equivalente virtual podrá usar redes sociales como Facebook y el servicio de mensajería WhatsApp, los cuales brindan soporte para abrir espacios de debate y reflexión.

Acción del estudiante.

En este segundo nivel el estudiante debe afianzar su aprendizaje, participando de los debates y cuestionamientos del profesor y sus compañeros dentro el aula. En cuanto al aspecto virtual, el estudiante deberá ser reflexivo y participar en las preguntas planteadas en redes sociales y el servicio de mensajería.

Fase 3 – Analizar

Acción del profesor.

El profesor en esta fase debe plantear problemas algorítmicos los cuales deben ser afrontados por los estudiantes, en este caso se puede recurrir a la pizarra, hoja de problemas y diapositivas. De forma virtual podrá citar preguntas de juez virtual, ya sea dentro o fuera del aula. Adicionalmente podrá recurrir al uso de un cronometro para controlar el tiempo de solución del problema.

Acción del estudiante.

Esta fase demanda del estudiante uso extremo del pensamiento abstracto en todas las situaciones. Pueden presentarse en forma de ideas generales, reglas, procedimiento o métodos generalizados o también pueden ser principios, ideas y

teorías que deben recordarse de memoria. Ya sea dentro o fuera del aula, esta fase exige de sobremanera el uso de conocimientos previos del estudiante. Es importante destacar que la respuesta del estudiante dependerá mucho de su capacidad de lectura de comprensión, lógica, matemática y creatividad. Así también de la capacidad de sobrellevar la enseñanza aprendizaje bajo presión.

Fase 4 - Aplicar

Acción del profesor

El profesor podrá realiza la representación de la posible solución de los problemas planteados llegando a hacer uso de Diagramas de Flujo y algún lenguaje de programación como C++. La proporción de repuestas será gradual, pidiendo ser una solución parcial y en casos extremos una solución integra en función del tipo de problema ya sea ejemplo o un problema planteado. Esto podrá realizarse tanto dentro como fuera del aula. Fuera del aula en forma de capturas de pantalla.

Acción del estudiante

Esta fase requiere que el estudiante haga uso de heurística y presente la mejor solución al problema, pudiendo realizar esta representación mediante Diagramas de Flujo o C++. Según el tipo de problema, si se trata de una solución proporcionada por parte del profesor deberá el procedimiento a seguir es reflexivo y copiado al cuaderno de apuntes para un posterior estudio fuera del aula. Si el problema fue planteado el estudiante deberá exponer su solución en redes sociales o el servicio de mensajería en forma de captura de pantalla.

Fase 5 - Evaluar

Acción profesor

El profesor debe evaluar el conjunto de elementos que hace funcional a una respuesta generada, de esta forma se da paso a las pruebas de escritorio y la compilación de código. Fuera del aula el profesor podrá hacer uso del Juez Virtual.

Acción estudiante

Luego de trabajar con diagramas o código, organizarlos para formar una solución, será indispensable hacer la evaluación del análisis y aplicación, los juicios se desarrollarán en función a criterios de comparación según alguna respuesta esperada ya sea en la hoja de preguntas o en el juez virtual.

Nivel 6 - Solucionar

Acción profesor

En este punto el profesor si se encuentra en el aula podrá ofrecer una aclaración respecto al problema abordado y su solución si se trata de un problema de tipo ejercicio. Si el profesor planteo un problema, deberá evaluar haciendo una comparación con los resultados esperados. Fuera del aula, el profesor solo es un observador de los resultados obtenidos en el juez virtual, pidiendo revisar el código de solución de forma remota.

Acción estudiante.

En función a la respuesta retornada ya sea por el profesor o juez virtual, en caso de ser positiva, se deberá reforzar la experiencia cognitiva de crear. Si la respuesta por parte del profesor o juez es negativa, el estudiante deberá retornar a la fase 3 hasta

lograr una experiencia cognitiva satisfactoria, habiendo logrado crear una solución integra.

Fase 7 – Conectar y socializa

Acción Profesor

La socialización es el pilar fundamental del método, por lo tanto y de acuerdo al juicio del profesor, se podrá designar a ciertos estudiantes identificados que hayan superado las 6 fases anteriores dando la autorización para enseñar a alguno de sus compañeros. Fuera del aula de deberá crear la apertura de espacios de socialización virtual para la propuesta y solución de variados problemas algorítmicos en las redes sociales y los sistemas de mensajería. En esta fase, el profesor fomentará en sus estudiantes el poder compartir pistas para el grupo de estudio virtual.

Aspecto Cognitivo

Gracias al fortalecimiento de los aspectos cognitivos de los niveles anteriores, los estudiantes deben responde a su entorno realizando interacciones con sus equivalentes, propiamente el acto de enseñar para fortalecer su aprendizaje llegando de esta forma a compartir su conocimiento. Fuera del aula queda la obligación de participar en las redes sociales como Facebook y el servicio de mensajería WhatsApp.

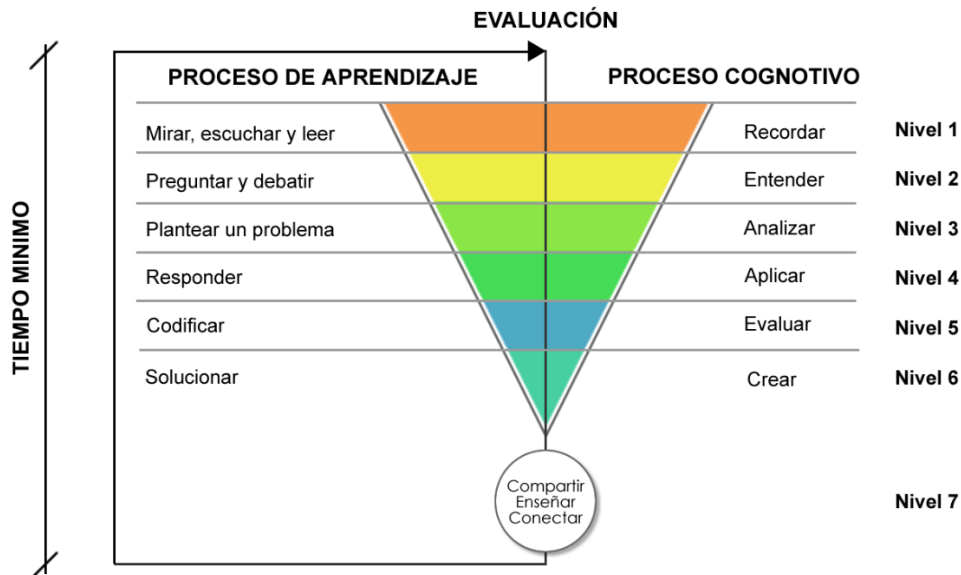


Figura 3.2 Descripción de niveles, acciones, procesos cognitivos y herramientas didácticas del método conectivo bajo presión V-Bloom Fuente de elaboración propia 2017

# FASE	ACCIÓN	PROCESO COGNITIVO	RECURSO DIDÁCTICO
Fase 1	Mirar, escuchar y leer	Recordar	Presentaciones, libros, textos impresos, documentos PDF y videos digitales.
Fase 2	Preguntar y debatir	Comprender	Debate presencial y virtual
Fase 3	Plantear un problema	Analizar	Papel y lápiz
Fase 4	Responder	Aplicar	Diagramas de Flujo y C++
Fase 5	Codificar	Evaluar	Pruebas de escritorio y cálculo de tiempo de ejecución
Fase 6	Solucionar	Crear	Juez en línea
Fase 7	Socializar	Compartir	Redes Sociales

Nota. Fuente: Elaboración propia 2017

Tabla 3.3

Cuadro descriptivo de los niveles del método V-Bloom.

Fase 3	Plantear un problema	Analizar	Papel y lápiz
Fase 4	Responder	Aplicar	Diagramas de Flujo y C++
Fase 5	Codificar	Evaluar	Pruebas de escritorio y cálculo de tiempo de ejecución
Fase 6	Solucionar	Crear	Juez en línea
Fase 7	Socializar	Compartir	Redes Sociales

CAPITULO IV EVALUACIÓN DEL MÉTODO

4.1. INVESTIGACIÓN

En este apartado se presenta una descripción completa de las pruebas estadísticas a las cuales fue sometido el método conectivo bajo presión V-Bloom, el cual fue utilizado para llevar a cabo la investigación. También se brindará información de la muestra y datos que sirvieron para dar respuesta a la hipótesis. Asimismo, se describen los instrumentos que se emplearon y los procedimientos que se siguieron para recolectar los datos y el tratamiento posterior para su interpretación.

4.1.1. ENFOQUE DE LA INVESTIGACIÓN.

La investigación es de tipo cuantitativa, el diseño tiene dos grupos y se evalúa a ambos con una variable dependiente (calificaciones) que en este caso son los puntajes obtenidos por cada estudiante de acuerdo a una de las cuatro etapas de Olimpiada Científica Plurinacional 2017 (Nivel 2), llegándose a observar la aplicación del método V-Bloom en tres ocasiones (Etapa Distrital, Etapa Departamental y Etapa Nacional), tanto en el grupo experimental (con método V-Bloom) como en el grupo de control (con otro método), los grupos corresponden a la variable independiente (método). En este diseño los sujetos a los cuales se aplicó el método experimental no fueron asignados aleatoriamente pues fueron el resultado de un proceso de selección voluntaria.

4.1.2. TIPO DE INVESTIGACIÓN

Se trata de una investigación cuantitativa, cuasi experimental. En esta investigación se pretende probar el efecto del método V-Bloom sobre las calificaciones obtenidas en cada una de las etapas de la Olimpiada Boliviana de Informática, disponiendo de un grupo sobre el cual se aplica el método V-Bloom frente a otro que aplica otros métodos. No existiendo mediciones a priori, pero sí a posteriori.

4.1.3. OPERACIONALIZACIÓN DE VARIABLES

La operacionalización de variables puede apreciarse a continuación (ver Tabla 4.1).

Tabla 4.1
Tabla para la operacionalización de variables.

VARIABLES	DEFINICIÓN	INDICADORES	INSTRUMENTOS
Independiente	Con método V-Bloom. El grupo de estudiantes que participo de la capacitación bajo este método.	Separados por un identificador primer grupo el cual fue capacitado con el método V-Bloom y un segundo grupo que se capacito sin el método V-Bloom.	Reporte de resultados brindado por el Ministerio de Educación del Estado Plurinacional de Bolivia.
Con método V-Bloom			
Sin método V-Bloom	Sin método V-Bloom. El grupo de estudiantes que se capacito sin el método V-Bloom.		
Dependiente	Representa el dato calificación el cual es cuantificable, obtenido por el estudiante durante la etapa de selección correspondiente.	Calificación obtenida en la etapa correspondiente de selección.	Reporte de resultados brindado por el Ministerio de Educación del Estado Plurinacional de Bolivia.
Calificación obtenida en la etapa (2, 3 o 4)			

Nota. Fuente: Elaboración propia

4.1.4. POBLACIÓN

El tamaño de la población es de 446, el reporte se extrajo de la documentación proporcionada por TitanCod, Juez Virtual de evaluación de la 7ma Olimpiada Científica Estudiantil Plurinacional Boliviana Nivel 2 (ver Anexo A).

4.1.5. MUESTRA

Los grupos se distribuyeron de la siguiente forma: grupo de control 437 estudiantes (sin método V-Bloom) y grupo experimental 9 estudiantes (Con método V-Bloom), entre ellos varones y mujeres, con rango de edad entre 11 a 15 años con grado de escolaridad correspondiente a primaria y secundaria, inscritos en el Nivel 2 de la Olimpiada Boliviana

de Informática 2017. El grupo experimental corresponde a estudiantes de la ciudad de La Paz y el grupo de control está distribuido en los 9 departamentos del estado Plurinacional de Bolivia. La selección de la muestra experimental fue voluntaria.

4.1.6. TÉCNICAS PARA EL ANÁLISIS E INTERPRETACIÓN DE DATOS

El primer paso a seguir es determinar si la muestra procede de una población con distribución normal. Este problema es muy frecuente, ya que son muchas las pruebas de inferencia estadística que exigen como condición imprescindible que la población de procedencia de la muestra sea normal.

Por lo tanto, se aplicó las pruebas de Kolmogorov-Smirnova y Shapiro-Wilks las cuales verifican la normalidad, posteriormente la prueba U de MannWhitney y Kolmogorov-Smirnova, una prueba no paramétrica aplicada sobre dos muestras independientes. Es, de hecho, una prueba cuantitativa, equivalente a la versión no paramétrica de la habitual prueba t de Student.

Finalmente se procedió con la prueba estadística para conocer diferencias significativas de las calificaciones obtenidas mediante la prueba de Mann-Whitney, estableciendo una comparación de medianas. La investigación pretende demostrar la diferencia significativa entre las calificaciones obtenidas, utilizando el valor alfa de ,05 y con un intervalo de confianza de un 95% para aportar un análisis más preciso en cada respuesta.

4.2. VALIDACIÓN DEL MÉTODO

4.2.1. BASES DE LA PROPUESTA.

Durante cada fase se realiza una prueba de normalidad de acuerdo a las calificaciones obtenidas de la muestra, seguidamente se aplicará una prueba de normalidad Kolmogorov-Smirnov si la muestra es mayor a 30 estudiantes o una prueba de Shapiro-Wilks si la muestra es menor o igual a 30. De acuerdo al anterior resultado se podrá proceder con una prueba paramétrica T de Student o caso contrario con una prueba no paramétrica de Mann-

Whitney. El análisis será realizado mediante el programa estadístico SPSS, la precisión de los cálculos permitirá realizar un análisis e interpretación final además de un contraste apoyado en diagramas de caja.

4.2.2. DESARROLLO DE LA PROPUESTA.

4.2.2.1. VALIDACIÓN ETAPA 1

La primera etapa está destinada a las inscripciones, por lo cual no existe evaluación alguna.

La estructura numérica del experimento se encuentra definida en la Tabla 4.2.

Tabla 4.2
Cantidad de inscritos de la primera etapa

	Grupo de control	Grupo experimental	Total
Etapa 1 – Unidades Educativas	437	9	446

Nota. El grupo de control corresponde a 437 estudiantes distribuidos en toda Bolivia a los cuales sus profesores aplicaron diversos métodos de enseñanza aprendizaje. El grupo experimental está formado por 9 estudiantes a los cuales se aplicó el método V-Bloom. Fuente: TitanCod OBI 2017 (ver Anexos A)

4.2.2.2. VALIDACIÓN ETAPA 2

La etapa 2, estuvo sujeta a una evaluación de connotación ICPC (ver Anexo B), donde el sistema evaluador fue TitanCod juez virtual oficial de la OBI, bajo las correcciones y validación de los distintos encargados departamentales. El puntaje máximo esperado fue de 300 puntos y mínimo de 1 punto, para pasar a la siguiente fase, el número de participantes fue una directa consecuencia de los inscritos en la etapa anterior (ver Tabla 4.2). Los datos están descritos en el Anexo A.

En esta etapa la muestra es total de 446 estudiantes, por lo tanto, se ajusta al aproximado para realizar una prueba de normalidad Kolmogorov-Smirnov, bajo las siguientes hipótesis:

H_0 : La variable sigue una distribución normal (Hipótesis Nula).

H_1 : La variable no sigue una distribución normal (Hipótesis Alterna).

Se selecciona la hipótesis nula cuando el valor calculado Sig. es mayor a ,5 o caso contrario si está por debajo de ,5 se aceptaría la hipótesis alterna.

Los resultados calculados por SPSS (ver Tabla 4.3) son:

Tabla 4.3

Prueba de normalidad Kolmogorov-Smirnov generada con SPSS para datos de la etapa 2:

	Kolmogorov-Smirnov ^a		
	Estadístico	gl	Sig.
Calificación de la etapa 2	,444	446	,000

Nota. Fuente: Formulación elaborada a través de la investigación 2017

Como Sig. = ,000 no llegando a ser mayor a ,05 se acepta la Hipótesis alterna.

H₁: La variable no sigue una distribución normal (Hipótesis Alterna).

Puesta en evidencia que las muestras no tienen comportamiento normal, se aplicara el procedimiento correspondiente aplicando pruebas de estadística no paramétrica, en este caso corresponde una prueba U de Mann-Whitney, procederá dividir la muestra en dos grupos para determinar una diferencia de medias, la cual no se deba al azar es decir que la diferencia sea estadísticamente significativa. Nuevamente se recurre al planteamiento de una hipótesis

H₀: Las calificaciones obtenidas por el grupo que aplicó el método V-Bloom frente al grupo que no aplico el método V-Bloom son iguales (Hipótesis Nula).

H₁: Las calificaciones obtenidas por el grupo que aplicó el método V-Bloom frente al grupo que no aplico el método V-Bloom son diferentes (Hipótesis Alterna).

Los resultados calculados por SPSS (ver

Tabla 4.4) son:

Tabla 4.4
Prueba de U de Mann-Whitney diferencia de promedios generada con SPSS para los grupos de control y experimental de la etapa 2

Rangos				
	Entrenamiento	N	Rango promedio	Suma de rangos
Calificación de la etapa 2	Con método V-Bloom	9	436,67	3930,00
	Con otro método	437	219,11	95751,00
	Total	446		

Estadísticos de contraste	
	Calificación de la etapa 2
U de Mann-Whitney	48,000
W de Wilcoxon	95751,000
Z	-6,789
Sig. asintótica (bilateral)	,000
Significación exacta (bilateral)	,000
Significación exacta (unilateral)	,000
Probabilidad en el punto	,000

Se selecciona la hipótesis nula, cuando el valor calculado Significación exacta (unilateral) es mayor a ,5 o caso contrario si está por debajo de ,5 se aceptaría la hipótesis alterna.

En este caso el nivel de significancia Sig. exacta (unilateral) es menor o igual a ,05 se tiene que rechazar la hipótesis nula, y aceptar la hipótesis alterna.

H₁: Las calificaciones obtenidas por el grupo que aplicó el método V-Bloom frente al grupo que no aplicó el método V-Bloom son diferentes (Hipótesis Alterna).

Por tanto, las calificaciones obtenidas por los grupos indican que existe una diferencia significativa con un nivel de confianza del 95%, donde las calificaciones más altas obtenidas mediante el método V-Bloom señalan mayor efectividad en la etapa 2, ya que el

rango promedio de este grupo (436,67), es significativamente mayor y diferente en comparación al grupo que no aplicó el método V-Bloom (219,11).

La afirmación anterior puede ser contrastada con un diagrama de cajas (ver Figura 4.1).

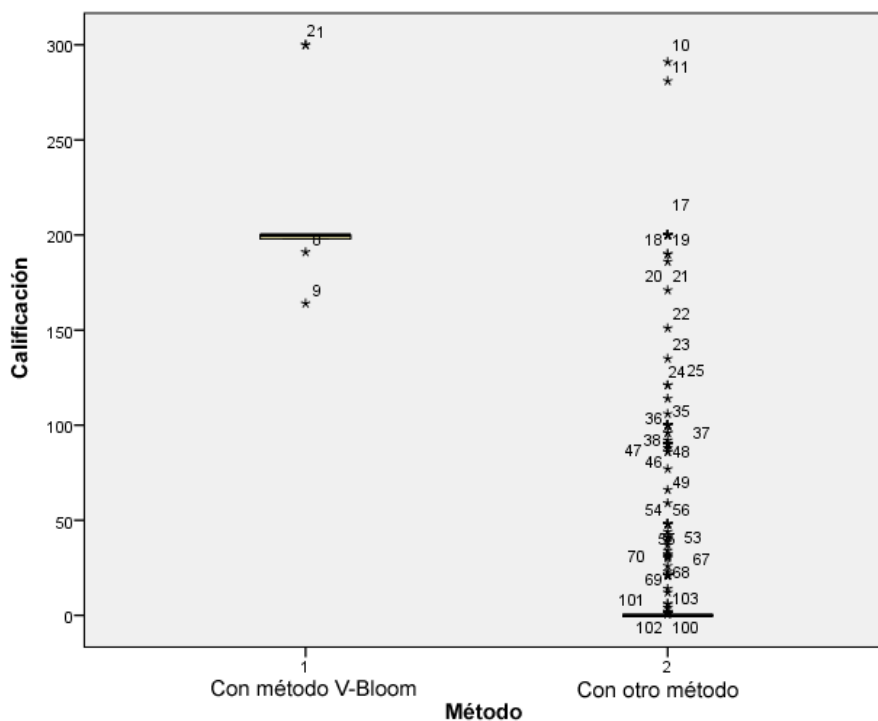


Figura 4.1 Diagrama de cajas correspondiente a los grupos y puntajes obtenidos en la etapa 2.

El diagrama de caja del Grupo Experimental (método V-Bloom) muestra una calificación máxima de 200 y mínima de 200 puntos. Con puntos varios atípicos, pero dando consistencia a notas concentradas en su mayoría en los 200 puntos.

El diagrama de caja del grupo que no empleó el método V-Bloom, muestra un comportamiento atípico con una concentración atípica 25 a 100, con tendencia a calificación máxima de 0 y mínima de 0 puntos.

Se puede concluir que las calificaciones obtenidas por el método V-Bloom, son diferentes y con valores más altos.

Finalmente, los números post-test de esta etapa son presentados en la Tabla 4.5.

Tabla 4.5
Numero de aprobados post-test segunda etapa

	Grupo de control	Grupo experimental	Total
Etapa 2 - Distrital	116	9	125

Nota. El grupo de control corresponde a 116 estudiantes distribuidos en toda Bolivia a los cuales sus profesores aplicaron diversos métodos de enseñanza aprendizaje. El grupo experimental está formado por 9 estudiantes a los cuales se les aplicó el método V-Bloom. Fuente: TitanCod OBI 2017 (Ver Anexo A)

4.2.2.3. VALIDACIÓN ETAPA 3.

La etapa 3, estuvo sujeta a una evaluación de connotación ICPC, donde el sistema evaluador fue TitanCod juez virtual oficial de la OBI, bajo las correcciones y validación de los distintos encargados departamentales. El puntaje máximo esperado fue de 300 puntos, donde solo se pudo considerar las 7 mejores calificaciones obtenidas para pasar a la siguiente etapa, el número de participantes fue una directa consecuencia de los aprobados de la anterior etapa (ver Tabla 4.5). Los datos están descritos en el Anexo C.

Es así la muestra para esta etapa de 125 estudiantes en total, por lo tanto, se ajusta al aproximado para realizar una prueba de normalidad Kolmogorov-Smirnov, bajo las siguientes hipótesis:

H_0 : La variable sigue una distribución normal (Hipótesis Nula).

H_1 : La variable no sigue una distribución normal (Hipótesis Alterna).

Se selecciona la hipótesis nula cuando el valor calculado Sig. es mayor a ,5 o caso contrario si está por debajo de ,5 se aceptaría la hipótesis alterna.

Los resultados calculados por SPSS (ver Tabla 4.6) son:

Tabla 4.6

Prueba de normalidad Kolmogorov-Smirnov generada con SPSS para datos de la etapa 3:

	Kolmogorov-Smirnov ^a		
	Estadístico	gl	Sig.
Calificación de la etapa 3	,449	125	,000

Nota. Formulación elaborada a través de la investigación

Como Sig. = ,000 no llegando a ser mayor a ,05 se acepta la Hipótesis alterna.

H₁: La variable no sigue una distribución normal (Hipótesis Alterna).

Puesta en evidencia que la muestra tiene comportamiento normal, se aplicará el procedimiento correspondiente empleando una prueba estadística no paramétrica, en este caso corresponde una prueba U de Mann-Whitney, procederá dividir la muestra en dos grupos para determinar una diferencia de medias, la cual no se deba al azar es decir que la diferencia sea estadísticamente significativa. Nuevamente se recurre al planteamiento de una hipótesis

H₀: Las calificaciones obtenidas por el grupo que aplicó el método V-Bloom frente al grupo que no aplicó el método V-Bloom son iguales (Hipótesis Nula).

H₁: Las calificaciones obtenidas por el grupo que aplicó el método V-Bloom frente al grupo que no aplicó el método V-Bloom son diferentes (Hipótesis Alterna).

Los resultados calculados por SPSS (ver Tabla 4.7) son:

Tabla 4.7

Prueba de U de Mann-Whitney diferencia de promedios generada con SPSS para los grupos de control y experimental de la etapa 3

Rangos

	Entrenamiento	N	Rango promedio	Suma de rangos
Calificación de	Con método V-Bloom	9	108,94	980,50

la etapa 3	Sin método V-Bloom	116	59,44	6894,50
	Total	125		

Estadísticos de contraste

	Calificación de la etapa 3
U de Mann-Whitney	108,500
W de Wilcoxon	6894,500
Z	-4,860
Sig. asintótica (bilateral)	,000
Significación exacta (bilateral)	,000
Significación exacta (unilateral)	,000
Probabilidad en el punto	,000

En este caso el nivel de significancia Sig. exacta (unilateral) es menor o igual a ,05 se tiene que rechazar la hipótesis nula, y aceptar la hipótesis alterna.

H_1 : Las calificaciones obtenidas por el grupo que aplicó el método V-Bloom frente al grupo que no aplicó el método V-Bloom son diferentes (Hipótesis Alterna).

Por tanto, las calificaciones obtenidas por los grupos indican que existe una diferencia significativa con un nivel de confianza del 95%, donde las calificaciones más altas obtenidas mediante el método V-Bloom señalan mayor efectividad en la etapa 3, ya que el rango promedio de este grupo (108,94), el cual es significativamente mayor y diferente en comparación al grupo que no aplicó el método V-Bloom (59,44).

La afirmación anterior puede ser contrastada con un diagrama de cajas (ver Figura 4.2).

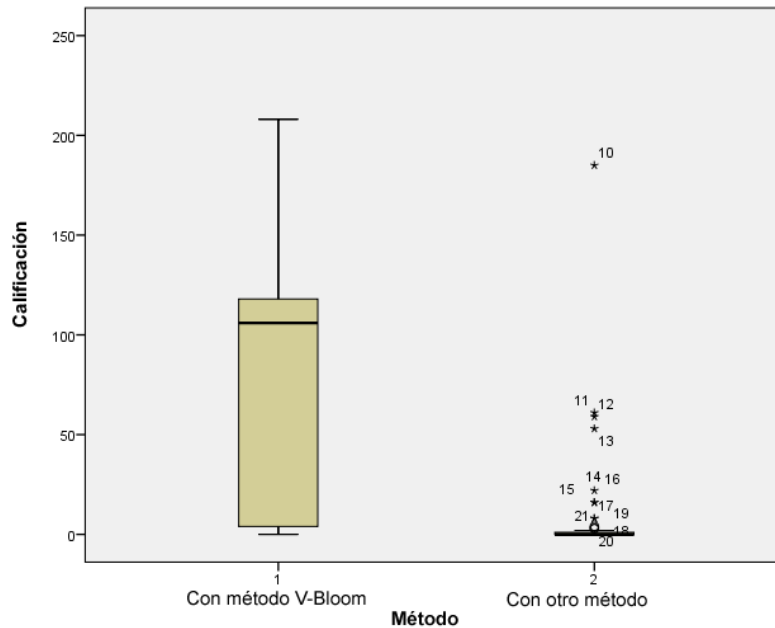


Figura 4.2 Diagrama de cajas correspondiente a los grupos y puntajes obtenidos en la etapa 3.

El diagrama de caja del Grupo Experimental (V-Bloom) muestra una calificación máxima de 210 y mínima de 0 puntos, la calificación se encuentra más dispersas entre 5 y 105 siendo mucho más concentradas entre 105 y 120, con tendencia mayor a notas entre 120 y 210.

El diagrama de caja del grupo que no empleo el método V-Bloom, muestra un comportamiento atípico con calificación máxima de 0 y mínima de 0 puntos. Con varios puntos atípicos.

Se puede concluir que las calificaciones obtenidas por el método V-Bloom, son diferentes y con valores más altos.

Destacar en este apartado que los estudiantes bajo el método V-Bloom cubrieron los 7 cupos destinados para el departamento de La Paz con 1 medalla de oro, 1 de plata, 1 bronce y 4 menciones de honor (ver Figura 4.3).

Tabla 4.8
Numero de aprobados post-test segunda fase

	Grupo de control	Grupo experimental	Total
Etapa 3 - Distrital	31	7	38

Nota. El grupo de control corresponde a 31 estudiantes distribuidos en toda Bolivia a los cuales sus profesores aplicaron diversos métodos de enseñanza aprendizaje. El grupo experimental está formado por 7 estudiantes a los cuales se les aplicó el método V-Bloom. Fuente: TitanCod OBI 2017 (Ver Anexo C)



Figura 4.3 Premiación 3 etapa (departamental), estudiantes que fueron capacitados bajo el método V-Bloom, el documento que respalda esta afirmación puede apreciarse en el Anexo D.

4.2.2.4. AVANCES EN EL JUEZ VIRTUAL PRE-TEST ETAPA 4.

En este punto de la investigación, es importante realizar una cuantificación previa y observar los reportes del juez virtual de la Universidad Mayor de San Andrés (Juez Patito) de la carrera de Informática, habiendo realizado simulaciones bajo el método V-Bloom entre el 8 de agosto y 13 de octubre de 2017, fechas previas a la etapa final (ver Figura 4.4).

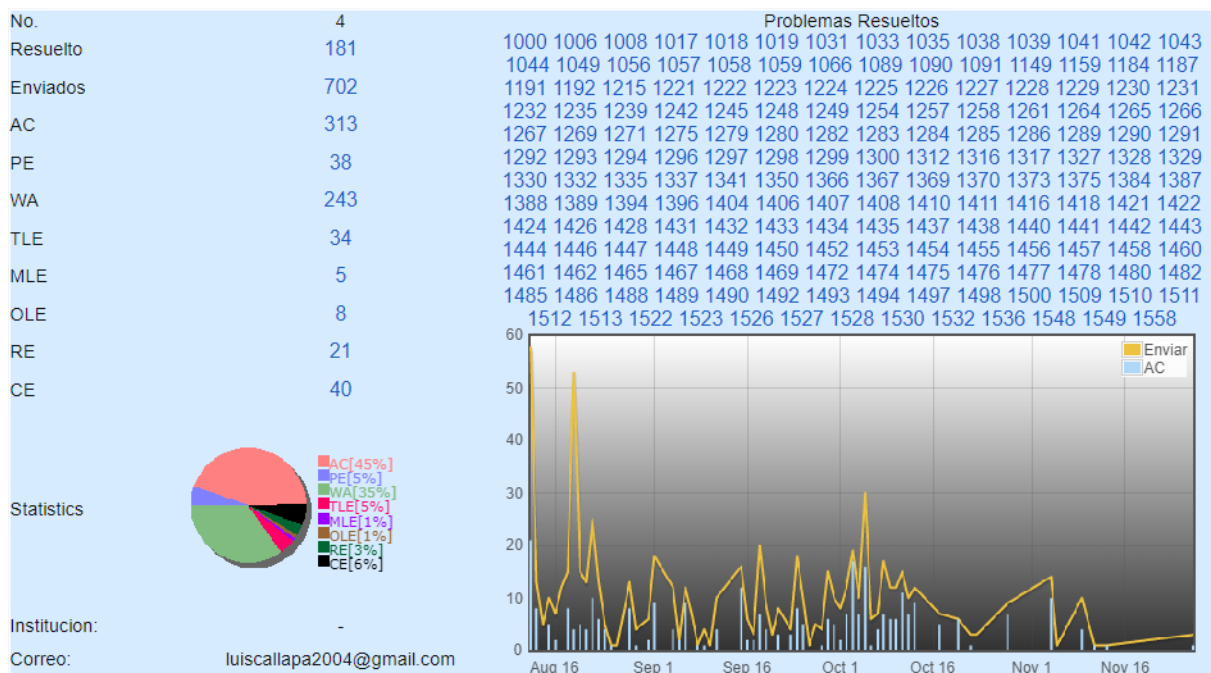


Figura 4.4 Reporte de desempeño del mejor estudiante del grupo experimental.
Fuente: Juez Patito UMSA Carrera de Informática.

El significado de las siglas correspondientes al reporte del Juez Patito se puede ver en la Tabla 4.9.

Tabla 4.9
Significado de siglas del reporte del Juez Patito

SIGLA	SIGNIFICADO
AC	Accepted: OK! todo está super.
PE	Pending Rejudge: Los datos de prueba se actualizaron y se revisarán nuevamente.
WA	Wrong Answer: Tu código no corre para todos los casos.
TLE	Time Limit Exceeded: Tu programa no corre dentro los límites de tiempo.
MLE	Memory Limit Exceeded: Tu programa consume mucha memoria.
OLE	Output Limit Exceeded: Tu programa intento escribir demasiada información de salida
RE	Runtime Error: Tu programa se desborda o hay división entre cero.
CE	Compile Error: Hay un error en la compilación.

Nota. Fuente: FAQ Juez Patito UMSA 2017

Por lo cual se hace necesario presentar un resumen de los 7 estudiantes del grupo experimental que llegaron a la etapa 4 nacional bajo el método V-Bloom (ver Tabla 4.10).

Tabla 4.10
Resumen de resultados de las prácticas en Juez Virtual Patito Informática UMSA

Estudiante	Cod Usuario	# Juez Patito	Problemas Enviados	Problemas Resueltos
1	luchos	4	702	181
2	gpacajes	9	544	152
3	DanerZein	11	477	144
4	chubyxdxd	12	415	141
5	Amy_on_ice	16	377	136
6	Masterdan	25	429	106
7	sebasrvillena2005	93	221	48

Nota. El detalle completo de los 7 participantes se puede apreciar en el Anexo E.

Fuente: Reporte Juez Patito UMSA 2017

4.2.2.5. VALIDACIÓN ETAPA 4.

La etapa 4, estuvo sujeta a una evaluación de connotación ICPC, donde el sistema evaluador fue TitanCod juez virtual oficial de la OBI, bajo las correcciones y validación de los jueces físicos y el comité académico nacional de informática.

El puntaje máximo esperado fue de 400 puntos y mínimo de 70 puntos para obtener medallas, 69 a 30 puntos para obtener menciones de honor, se consideró las mejores calificaciones en esta última fase, donde el número de participantes fue una directa consecuencia de los aprobados de la anterior etapa (ver Tabla 4.5). El acta de resultados nacional se encuentra en los Anexo F.

En esta etapa la muestra es de 446 estudiantes, por lo tanto, se ajusta al aproximado para realizar una prueba de normalidad Kolmogorov-Smirnov, bajo las siguientes hipótesis:

H_0 : La variable sigue una distribución normal (Hipótesis Nula).

H₁: La variable no sigue una distribución normal (Hipótesis Alternativa).

Se selecciona la hipótesis nula cuando el valor calculado Sig. es mayor a ,5 o caso contrario si está por debajo de ,5 se aceptaría la hipótesis alternativa.

Los resultados calculados por SPSS (ver Tabla 4.11) son:

Tabla 4.11

Prueba de normalidad Kolmogorov-Smirnov generada con SPSS para datos de la etapa 4:

	Kolmogorov-Smirnov		
	Estadístico	gl	Sig.
Calificación de la etapa 4	,401	38	,000

Nota. Formulación elaborada a través de la investigación

Como Sig. = ,000 no llegando a ser mayor a ,05, se acepta la Hipótesis alternativa H₁: La variable no sigue una distribución normal (Hipótesis Alternativa).

H₁: La variable no sigue una distribución normal (Hipótesis Alternativa).

Puesta en evidencia que las muestras tienen comportamiento normal, se aplicará el procedimiento correspondiente empleando pruebas de estadística no paramétrica, en este caso corresponde una prueba U de Mann-Whitney, procederá dividir la muestra en dos grupos para determinar una diferencia de medias, la cual no se deba al azar es decir que la diferencia sea estadísticamente significativa. Nuevamente se recurre al planteamiento de las hipótesis.

H₀: Las calificaciones obtenidas por el grupo que aplicó el método V-Bloom frente al grupo que no aplicó el método V-Bloom son iguales (Hipótesis Nula).

H₁: Las calificaciones obtenidas por el grupo que aplicó el método V-Bloom frente al grupo que no aplicó el método V-Bloom son diferentes (Hipótesis Alternativa).

Los resultados calculados por SPSS (ver Tabla 4.12) son:

Tabla 4.12
Prueba de U de Mann-Whitney diferencia de promedios generada con SPSS para los grupos de control y experimental de la etapa 4

Rangos				
	Entrenamiento	N	Rango promedio	Suma de rangos
Calificación de la etapa 4	Con método V-Bloom	7	34,36	240,50
	Sin método V-Bloom	31	16,15	500,50
	Total	38		

Estadísticos de contraste	
	Calificación de la etapa 4
U de Mann-Whitney	4,500
W de Wilcoxon	500,500
Z	-4,891
Sig. asintótica (bilateral)	,000
Significación exacta (bilateral)	,000
Significación exacta (unilateral)	,000
Probabilidad en el punto	,000

Se selecciona la hipótesis nula, cuando el valor calculado Significación exacta (unilateral) es mayor a ,5 o caso contrario si está por debajo de ,5 se aceptaría la hipótesis alterna.

En este caso el nivel de significancia Sig. exacta (unilateral) es menor o igual a ,05 se tiene que rechazar la hipótesis nula, y aceptar la hipótesis alterna.

H₁: Las calificaciones obtenidas por el grupo que aplicó el método V-Bloom frente al grupo que no aplicó el método V-Bloom son diferentes (Hipótesis Alterna).

Por tanto, las calificaciones obtenidas por los grupos indican que existe una diferencia significativa con un nivel de confianza del 95%, donde las calificaciones más altas obtenidas mediante el método V-Bloom señalan mayor efectividad en la etapa 4, ya que el rango promedio de este grupo es (34,36), el cual es significativamente mayor y diferente en comparación al grupo que no aplicó el método V-Bloom (16,15).

La afirmación anterior puede ser contrastada con un diagrama de cajas (ver Figura 4.5).

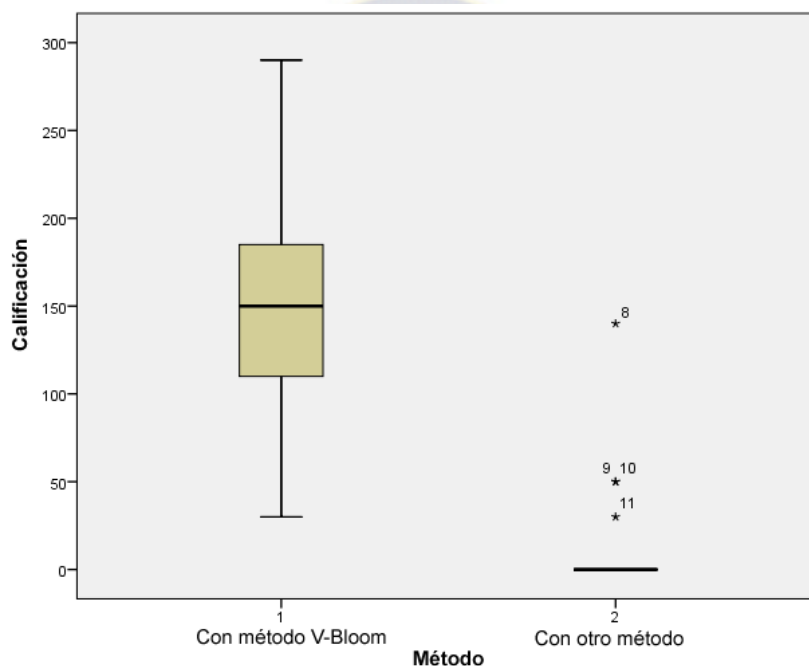


Figura 4.5 Diagrama de cajas correspondiente a los grupos y puntajes obtenidos en la etapa 4.

El diagrama de caja del Grupo Experimental (método V-Bloom) muestra una calificación máxima de 290 y mínima de 25 puntos, la calificación se encuentra igualmente concentradas entre 105 y 180, con tendencia mayor a notas entre 180 y 290.

El diagrama de caja del grupo que no empleo el método V-Bloom, muestra un comportamiento atípico con calificación máxima de 0 y mínima de 0 puntos. Con varios puntos atípicos.

Se puede concluir que las calificaciones obtenidas por el método V-Bloom, son diferentes y con valores más altos.

Finalmente, los números post-test de esta etapa son presentados en la Tabla 4.13.

Tabla 4.13
Numero de aprobados post-test segunda fase

	Grupo de control	Grupo experimental	Total
Etapa 4 - Nacional	3	7	10

Nota. Fuente: acta Olimpiada Boliviana de Informática Sucre 2017 (Ver Anexo F)

Destacar en este apartado que los estudiantes bajo el método V-Bloom consiguieron los siguientes logros en la etapa nacional: 1 medalla de oro, 2 de plata, 3 bronce y 1 mención de honor (ver Figura 4.3).



Figura 4.6 Premiación 4 etapa (Nacional), estudiantes que fueron capacitados bajo el método V-Bloom, el documento que respalda esta afirmación puede apreciarse en el Anexo F conjuntamente con las calificaciones.

CAPITULO V CONCLUSIONES Y RECOMENDACIONES

5.1. CONCLUSIONES

Cabe recordar que el objetivo general de esta investigación fue plantear un método que mejore los resultados obtenidos por los estudiantes de nivel 2 en las distintas etapas de la Olimpiada Boliviana de Informática 2017.

Ahora bien, partiendo de la pregunta de investigación formulada y considerando los resultados obtenidos en las fases de estudio realizados se puede mencionar las siguientes conclusiones:

- Con el método conectivo bajo presión V-Bloom, se puede abordar la enseñanza aprendizaje de programación competitiva en estudiantes de 11 a 15 años los cuales participan de la Olimpiada Boliviana de Informática.
- Teniendo en cuenta los resultados estadísticos obtenidos se puede afirmar que las calificaciones obtenidas en las 4 etapas de selección de la OBI 2017 a través del método conectivo bajo presión V-Bloom son diferentes y mucho más altas en relación a aquellos estudiantes que no emplearon el método.
- El llenado de la plantilla del plan de clases permite una planificación más estricta de tal modo que permite cumplir con los objetivos del método.
- El empleo del Juez Virtual Patito de la UMSA, permitió una simulación cercana a las evaluaciones planteadas por la OBI asimismo permitió cuantificar los resultados de los estudiantes bajo el método V-Bloom.
- El método V-Bloom, debido a su carácter conectivo utiliza redes sociales como Facebook para establecer comunicación, además del servicio de mensajería como lo es WhatsApp. De esta forma los estudiantes y el profesor generan un ambiente competitivo fuera del aula.

5.2. RECOMENDACIONES

Algunas de las recomendaciones a tener en cuenta para la correcta implementación del método son:

- Generalmente los estudiantes no están familiarizados con ningún lenguaje de programación, es importante que el entrenador tome el tiempo necesario para la capacitación previa. Por tal razón es aconsejable crear un módulo introductorio.
- La administración del tiempo necesario para los entrenamientos debe ser estricta y sujeta a un cronograma de actividades.
- El entrenador debe motivar al estudiante para que participe en clases, tomando en cuenta su desempeño en las tareas individuales y grupales.
- Es recomendable que el docente cree un ambiente organizacional positivo con los estudiantes, con el propósito de estimular positivamente su aprendizaje.
- El uso de redes sociales es fundamental para la difusión y socialización de preguntas y respuestas, de esta forma poder realizar un entrenamiento atemporal.
- Una vez conseguida la consolidación de conocimientos básicos de programación, las simulaciones de competencias, realizadas en los distintos jueces virtuales representan el avance más importante en términos de resultados se refiere.

BIBLIOGRAFÍA

- 21stcenturyskills. (2017). Logros indispensables para los estudiantes del siglo XXI. *Extraído el 31 de Octubre de 2017 desde <http://eduteka.icesi.edu.co/articulos/SeisElementos>.*
- Aho, A. (2012). Computation and computational thinking. *The Computer Journal*, 832-835.
- Alcalde, E., & Garcia, M. (1992). *Metodología de la Programación*. McGraw Hill.
- Alles, M. A. (2003). *Gestión por competencias. El diccionario*. Buenos Aires: Granica.
- Anderson, L., & Krathwohl, D. (2001). *Taxonomy for Learning, Teaching and Assessing: a Revision of Bloom's Taxonomy of Educational Objectives*. New York: Longman.
- Area, M. (2007). *Algunos principios para el desarrollo de buenas prácticas pedagógicas con las TIC en el aula*.
- Arévalo, C., & Solano, L. (2013). *atrones de Comportamiento de Estudiantes de Programación al Utilizar una Herramienta de Visualización de Protocolos Verbales. Conferencias LACLO*.
- Arnal, D. (2007). *Conceptos de web 2.0 y biblioteca 2.0: origen, definiciones y retos*. 12.
- Barker, L., Coope, S., McGettrick, A., Thatcher, J., & Topi, H. (2015). Towards Grand Challenges in Computing Education Across Disciplines. *Proceedings of the 46th ACM Technical Symposium on Computer Science Education*, 438-439.
- Beekman, G. (1995). *Computación e Informática hoy*. Oregon State University. Estados Unidos: Addison Wesley Iberoamericana.
- Berrocoso, J., Sánchez, M., & Arroyo, F. (2015). El pensamiento computacional y las nuevas ecologías del aprendizaje. *Revista de Educación a Distancia*, 46.
- Bond, M. (2007). Tracking bad apples, Reporting the origin of null and undefined value errors. *Proceedings of the 22nd annual ACM SIGPLAN conference on Object-oriented*, 405-422.

- Brito, M., & Sá-Soares, F. (2010). Computer programming: fail fast to learn sooner. *Technology Enhanced Learning. Quality of Teaching and Educational Reform*.
- Brooks, F. (1986). No Silver Bullet. *University of North Carolina at Chapel Hill*.
- Budd, T. (2002). *Introducción a la Programación Orientada a Objetos*. Addison Wesley.
- Burge, J., & Brinkman, B. (2017). Teaching and Learning Under Pressure: Intensive (Accelerated, Block) Computer Science Courses . *ACM Digitla Library*.
- Burton, B. (2010). Encouraging algorithmic thinking without a computer. *Olympiads in Informatics*, 3-14.
- Ceria, S. (2014). *La enseñanza de computación en las escuelas*. Revisado el 1 de Noviembre de 2016: <http://www.lanacion.com.ar/1655879-la-ensenanza-de-computacion-en-las-escuelas>.
- code.org. (2017). <https://hourofcode.com/es>. Consulta 1 de Noviembre de 2017.
- Dale, N., Dale, C., & Weems, C. (s.f.). *Programación y resolución de problemas con C++*. McGraw-Hill.
- Decker, R., & Hirshfield, S. (1992). Pascal's Triangle. Reading, Written and Reasoning. *PWS-Keny*.
- Edward, S., & Bentley, M. (2011). *Methods of Teaching in the Classroom*.
- Gairin, J. M. (2002). *Números y algoritmos*. Síntesis.
- Garzón, N. E. (2015). *Una propuesta metodológica para la enseñanza de programación en ingeniería multimedia*. Universidad Militar Nueva Granada.
- Gillespie, C. W. (2004). La visión de Seymour Papert para la educación de la niñez: Estudio descriptivo de estudiantes de Head Start y kindergarten en aulas basadas en el descubrimiento y enriquecidas con el lenguaje de programación Logo. *Escuela de Educación de Drake University*.
- Glenn, J. (1995). *Introducción a las ciencias de la computación*. Iberoamericana Addison-Wesley.

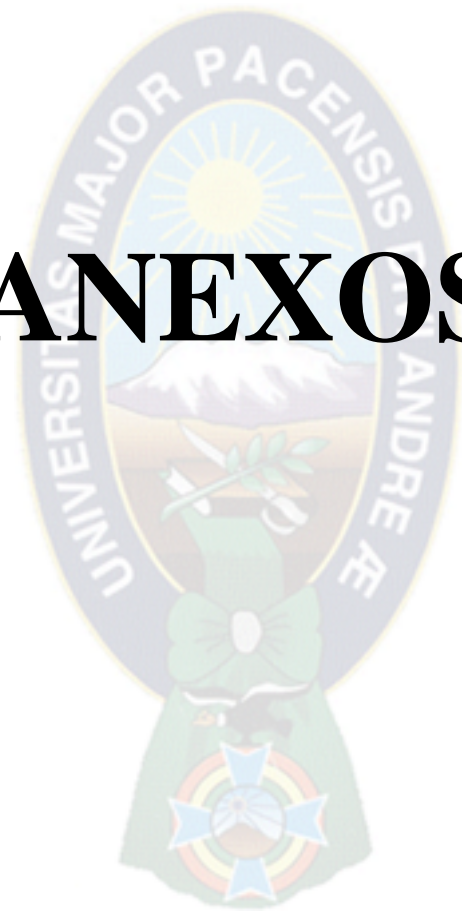
- Gonzalez, C. (2004). *The Role of Blended Learning in the World of Technology*. Recuperado el 1 de Noviembre de 2017: <http://www.unt.edu/benchmarks/archives/2004/september04/eis.htm>.
- González, F. (2008). *El mapa conceptual y el diagrama V. Recursos para la enseñanza superior en el siglo XXI*. Narsea S.A.
- Gosling, P. E. (1985). *Programación Estructurada para microcomputadoras*. McGraw Hill.
- Guerrero, M. (2002). *Diseño instruccional para la enseñanza de algoritmos*. Barquisimeto - Venezuela.: Trabajo para optar al Grado de Magíster Scientiarum.
- Guzdia, M. J. (2008). Education Paving the way for computational thinking. *Communications of the ACM*, 51.
- Hackbarth, S. (1996). *The educational technology handbook: a comprehensive guide: process and products for learning*. Educational Technology.
- Hakulinen, L. (2011). Survey on Informatics Competitions : Developing Tasks. *Olympiads in Informatics*, 12-25.
- Hakulinen, L. (2011). Survey on Informatics Competitions : Developing Tasks. *Olympiads in Informatics*, 12–25.
- Halim, S., & Halim, F. (2010). Competitive Programming in National University of Singapore . *Singapoure*, 22.
- Harel, D. (2004). *Algorithmics, the Spirit of Computing*. Addison Wesley.
- Ifrah, G. (1998). *Historia universal de las cifras* . México: S.L.U. ESPASA Libros.
- INTEF, I. N. (2015). *Programando nuestro futuro: Programación y codificación*. *Programando nuestro futuro: Programación y codificación*. Consultado el 1 de Octubre de 2017: http://educalab.es/documents/10180/509115/Informe_Programacion_Codificacion_2015/dd89d8c7-8a13-4a75-a36a-975a9affedc9.
- James, J., & Fletcher, L. (2009). Thinking About Computational Thinking Categories and Subject Descriptors. *Sigcse*, 260–264.

- Joyanes, L. (1987). *Metodología de la Programación. Diagramas de Flujo y programación*. McGraw Hill.
- Joyanes, L. (2003). *Fundamentos de Programación, Algoritmos, Estructuras de Datos y Objetos*. Madrid: McGraw-Hill.
- Kendall, K., & Kendall, J. (1997). *Análisis y Diseño de sistemas*. Pearson.
- Kereki, I. (2008). Scratch: Applications in Computer Science 1. *IEEE Frontiers in Education Conference*.
- Knuth, D. (1997). *The Art of Computer Programming, Vol. 1*. Addison Wesley.
- Kubica, M., & Radoszewski, J. (2010). Algorithms without Programming. *Olympiads in Informatics*, 52-66.
- Levine, G. (1989). *Introducción a la computación y a la programación estructurada*. McGraw-Hill.
- Malan, D., & Leitner, H. (2007). Scratch for budding computer scientists. *ACM SIGCSE Bulletin*, 223-227.
- Malliarakis, C., Satratzemi, M., & Xinogalos, S. (2014). Designing Educational Games for Computer Programming: A Holistic Framework. *Electronic Journal of e-Learning*.
- Martínez, J. (2004). *Estrategias metodológicas y técnicas para la investigación social*. México D.F.: Universidad Mesoamericana.
- McMillan, C., & González, R. (1977). *Análisis de Sistemas*. Trillas.
- Meinke, K. (2004). Automated black-box testing of functional correctness using function. *Proceedings of the 2004 ACM SIGSOFT international symposium on* , 143-153.
- Mercado, C., Andrade, E., & Reynoso, J. (2008). The effect of learning objects on a C++ programming lesson. *Proceedings of the 19th Annual International Information Management Association*.
- Miller, G. E. (1990). The assessment of clinical skills/competence/performance. *Academic medicine*, S63.

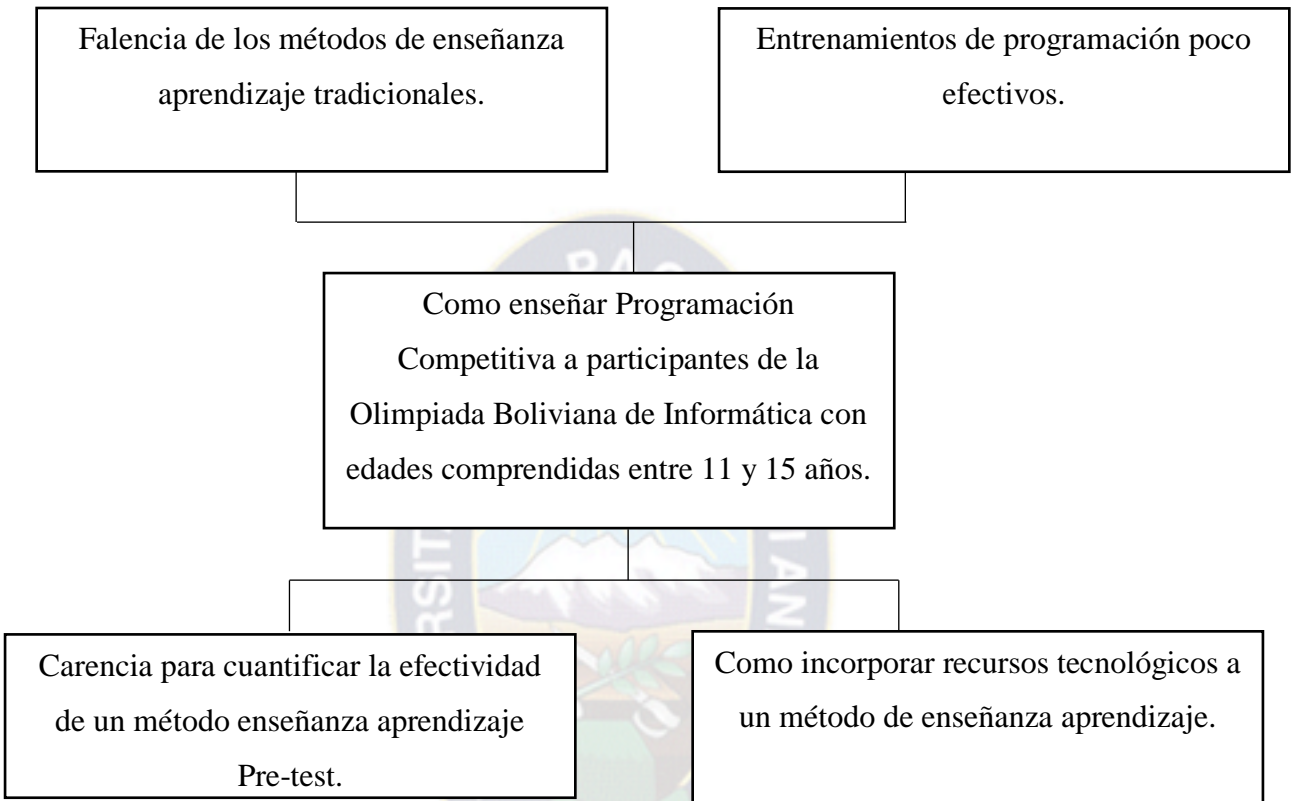
- Pérez, A. (2008). *“¿Competencias o pensamiento práctico? La construcción de los significados de representación y de acción.* Madrid: Morata.
- Perry, G. (1999). *Aprendiendo principios de programación en 24 horas.* Prentice Hall.
- Prensky, M. (2001). *Digital Natives, Digital Immigrants. NCB University Press, Vol. 9 No.*
- Pressman, R. S. (2010). *Ingeniería del Software. Un enfoque Práctico 7ma Edición.* McGraw-Hill.
- Ragonis, N. (2012). Type of Questions – The Case of Computer Science. *Olympiads in Informatics*, 115-132.
- Recio, C., Díaz, J., Saucedo, M., & Jiménez, S. (2017). VII Congreso Virtual Iberoamericano de Calidad en Educación Virtual y a Distancia.
- Revilla, M., Manzoor, S., & Liu, R. (2008). Competitive Learning in Informatics : The UVa Online Judge Experience. *Olympiads in Informatics*, 131–148.
- Richards, J., & Rodgers, T. (1963). *Approaches and methods in language teaching.*
- Rios, G. C. (2015). SCRATCH + ABP, Como estrategia para el desarrollo del Pensamiento Computacional. *PhD tesis, UNIVERSIDAD EAFIT.*
- Roa, R. (2001). *Algoritmos de cálculo.* España: Universidad de La Rioja.
- Rodríguez, O. (2011). *Paradigmas de Programación.* Valladolid: Universidad de Valladolid.
- Rojas, P. (2004). *Guía de Algoritmos. Teoría, Ejemplos y Ejercicios.* Barquisimeto - Venezuela: Colegio Fermín Toro.
- Ruiz, J., López, M., & Brito, J. (2016). Pensamiento sistémico y desarrollo de competencias, en el aprendizaje de los lenguajes de programación. *ANFEI Digital.*
- Senn, J. A. (1987). *Análisis y diseño de sistemas de información.* McGraw-Hill.
- Sobrino, A. (2011). Proceso de enseñanza-aprendizaje y web 2.0: valoración del conectivismo como teoría de aprendizaje post-constructivista.

- Society, R. (2012). Shut down or restart: The way forward for computing in UK schools. *[Informe Técnico]* , Revisado <http://royalsociety.org/education/policy/computing-in-schools/report/> .
- Sommerville, I. (2011). *Ingeniería De Software (Novena Edición)*. Pearson Educación.
- Toledo, F. (2002). *Fundamentos de Informática y Programación*. Valencia: Universidad de Valencia.
- Unda, S. M. (2004). *Modulo Instruccional Introducción a la Computación*. Barquisimeto - Venezuela.
- Verhoeff, T. (2009). 20 years of ioi competition tasks. *Olympiads in Informatics*, 149-166.
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*.
- Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions. Series A, Mathematical, Physical, and Engineering Sciences*, 3717-3725.
- Wolovick, N., & Martínez, C. (2016). Enseñar a Programar y Programar para Aprender. *Facultad de Matemática, Astronomía, Física y Computación.*, 8-10.
- Yourdon, E., & Constantine, L. (1979). *Structured design: Fundamentals of a discipline of computer program and systems design*. Prentice-Hall.
- Záhore, J., Hašková, A., & Munk, M. (2014). Assessment of Selected Aspects of Teaching Programming in SK and CZ. *Informatics in Education*, 157-178.

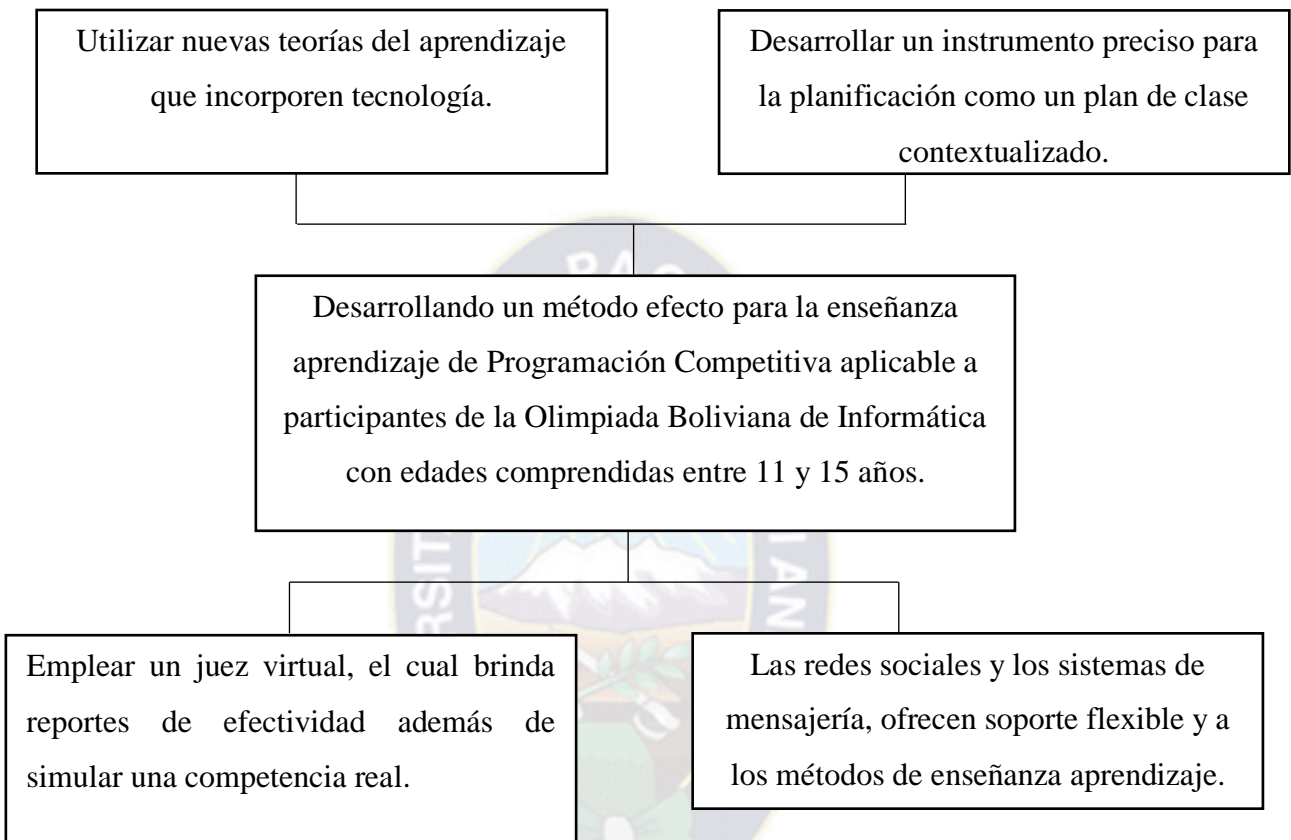
ANEXOS



ÁRBOL DE PROBLEMAS



ÁRBOL DE SOLUCIONES



ANEXO A

Calificaciones Etapa 2 (Distrital 2017)

URL de referencia: <http://www.titancod.net/obi2017/posiciones?nivel=2&pag=1>

Núm	RUDE	Nota	Departamento	Provincia	Localidad	U. Educativa
1	81480168200877	300	La Paz	Murillo	LA PAZ (CON MÉTODO V-BLOOM)	ROSA GATTORNO
2	8073010320071720	300	La Paz	Murillo	LA PAZ (CON MÉTODO V-BLOOM)	DON BOSCO A
3	812302422008124	291	Oruro	Cercado	ORURO	ALEMAN
4	819806362009477	281	Santa Cruz	Andres Ibañez	SANTA CRUZ DE LA SIERRA	EVANGELICO BUENAS NUEVAS B
5	812301842007928	200	Oruro	Cercado	ORURO	ALEMAN
6	8098011720103060	200	Chuquisaca	Oropeza	SUCRE	LUIS BOETO
7	8073066420102590	200	La Paz	Murillo	LA PAZ (CON MÉTODO V-BLOOM)	SAN IGNACIO
8	807300792008934	200	La Paz	Murillo	LA PAZ (CON MÉTODO V-BLOOM)	INGLES CATOLICO
9	807303972007119	200	La Paz	Murillo	LA PAZ (CON MÉTODO V-BLOOM)	SAGRADOS CORAZONES A
10	80730460201025000	200	La Paz	Murillo	LA PAZ (CON MÉTODO V-BLOOM)	AMERINST
11	8073011720072580	200	La Paz	Murillo	LA PAZ	MARIEN GARTEN C
12	814800302008766	200	Potosí	Tomas Frias	POTOSI	DON BOSCO A
13	8148004720071570	200	Potosí	Tomas Frias	POTOSI	DON BOSCO A
14	819806362007745	200	Santa Cruz	Andres Ibañez	SANTA CRUZ DE LA SIERRA	EVANGELICO BUENAS NUEVAS B
15	80730397200734	198	La Paz	Murillo	LA PAZ (CON MÉTODO V-BLOOM)	SAGRADOS CORAZONES A
16	812300692007264	191	La Paz	Murillo	LA PAZ (CON MÉTODO V-BLOOM)	SAGRADOS CORAZONES A
17	812302422008104	190	Oruro	Cercado	ORURO	ALEMAN
18	81981333200811200	190	Santa Cruz	Andres Ibañez	SANTA CRUZ DE LA SIERRA	EVANGELICO BUENAS NUEVAS D
19	812301842007629	186	Oruro	Cercado	ORURO	ALEMAN
20	814801612007186	171	Potosí	Tomas Frias	POTOSI	DON BOSCO A
21	807303592008448	164	La Paz	Murillo	LA PAZ (CON MÉTODO V-BLOOM)	MARIA AUXILIADORA I
22	814801562007453	151	Potosí	Tomas Frias	POTOSI	DON BOSCO A
23	814801442007292	135	Potosí	Tomas Frias	POTOSI	SANTA MARIA
24	814800302008500	121	Potosí	Tomas Frias	POTOSI	DON BOSCO A
25	8198012320091720	121	Santa Cruz	Andres Ibañez	SANTA CRUZ DE LA SIERRA	EVANGELICO BUENAS NUEVAS D
26	812300692007404	114	Oruro	Cercado	ORURO	ALEMAN
27	8148016320082380	106	Potosí	Tomas Frias	POTOSI	SANTA MARIA
28	8123006920071290	100	Oruro	Cercado	ORURO	JESUS MARIA 2
29	8148008920071360	100	Potosí	Tomas Frias	POTOSI	SANTA MARIA
30	8148005720071560	100	Potosí	Tomas Frias	POTOSI	SUCRE
31	819806362009118	100	Santa Cruz	Andres Ibañez	SANTA CRUZ DE LA SIERRA	EVANGELICO BUENAS NUEVAS D
32	819806362009128	100	Santa Cruz	Andres Ibañez	SANTA CRUZ DE LA SIERRA	EVANGELICO BUENAS NUEVAS D
33	4073029220114650	100	La Paz	Murillo	EL ALTO	SANTA BARBARA
34	4073016020081620	100	La Paz	Murillo	EL ALTO	SANTA BARBARA
35	40730347200881	100	La Paz	Murillo	La Paz	Santa Barbara
36	814800432007849	96	Potosí	Tomas Frias	POTOSI	DON BOSCO A
37	814801572008262	96	Potosí	Tomas Frias	POTOSI	DON BOSCO A
38	812300922008338	92	Oruro	Cercado	ORURO	PANTALEON DALENCE 1
39	804500452008260	90	Chuquisaca	Tomina	PADILLA	SAN PIO X SECUNDARIA
40	8148013920071370	90	Potosí	Tomas Frias	POTOSI	SANTA MARIA
41	804500452008260	90	Chuquisaca	Tomina	PADILLA	SAN PIO X SECUNDARIA
42	812301842007704	90	Oruro	Cercado	ORURO	ALEMAN
43	819813662008521	90	Santa Cruz	Andres Ibañez	SANTA CRUZ DE LA SIERRA	CRISTO REY
44	8148013920071370	90	Potosí	Tomas Frias	POTOSI	SANTA MARIA
45	813800522007107	90	Potosí	Tomas Frias	POTOSI	LITORAL C
46	8123006920073660	88	Oruro	Cercado	ORURO	ALEMAN
47	81230242200779	86	Oruro	Cercado	ORURO	ALEMAN
48	7044002820073550	86	Chuquisaca	Tomina	PADILLA	SAN PIO X SECUNDARIA
49	82200067200980	77	Beni	Yacuma	SANTA ANA DEL YACUMA	ANDRES ALBERDI SALEGUI
50	814801682008728	66	Potosí	Tomas Frias	POTOSI	SANTA MARIA

51	814800522007104A	59	Potosi	Tomas Frias	POTOSI	LITORAL C
52	4073033520075210	48	La Paz	Murillo	EL ALTO	PUERTO DE MEJILLONES
53	822201782009990	48	La Paz	Murillo	EL ALTO	PUERTO DE MEJILLONES
54	4073033520075220	48	La Paz	Murillo	EL ALTO	PUERTO DE MEJILLONES
55	4073028320071930	48	La Paz	Murillo	EL ALTO	PUERTO DE MEJILLONES
56	407303352007529A	48	La Paz	Murillo	EL ALTO	PUERTO DE MEJILLONES
57	8141000120071670	44	Potosi	Sur Chichas	TUPIZA	SANTA ANA C
58	807304852009169	41	La Paz	Murillo	LA PAZ	MARIA AUXILIADORA I
59	614700012008744	41	Potosi	Rafael Bustillo	SIGLO XX	SIGLO XX AMERICA
60	809803762009133	41	Cochabamba	Cercado	COCHABAMBA	SAN AGUSTIN
61	8073022920081420	39	Beni	Vaca Diez	RIBERALTA	MARANATHA
62	805401962008196	37	La Paz	Caranavi	CALAMA	MEJILLONES II
63	30610001200767	34	La Paz	Aroma	COLLANA	TUPAC KATARI
64	613700012008202	32	Potosi	ANTONIO QUIJARRO	PORCO	JOSE MARIA LINARES
65	613700012008286	32	Potosi	ANTONIO QUIJARRO	PORCO	JOSE MARIA LINARES
66	816800722008466	31	Tarija	BURNET O OCONNOR	ENTRE RIOS	GRAL. FRANCISCO BOURDETH O CONNOR II
67	81980643200834	31	Santa Cruz	Andres Ibañez	SANTA CRUZ DE LA SIERRA	EVANGELICO BUENAS NUEVAS D
68	8168007320101800	30	Tarija	BURNET O OCONNOR	ENTRE RIOS	GRAL. FRANCISCO BOURDETH O CONNOR II
69	30610001200716A	26	La Paz	Aroma	COLLANA	TUPAC KATARI
70	814801862008209	23	Potosi	Tomas Frias	POTOSI	SANTA MARIA
71	809000402008698	21	Cochabamba	QUILLACOLLO	QUILLACOLLO - CIUDAD QUILLACOLLO	CALAMA
72	8090001420082150	21	Cochabamba	QUILLACOLLO	QUILLACOLLO - CIUDAD QUILLACOLLO	CALAMA
73	8090001520083280	21	Cochabamba	QUILLACOLLO	QUILLACOLLO - CIUDAD QUILLACOLLO	CALAMA
74	809000232008471	21	Cochabamba	QUILLACOLLO	QUILLACOLLO - CIUDAD QUILLACOLLO	CALAMA
75	8090001420072210	21	Cochabamba	QUILLACOLLO	QUILLACOLLO - CIUDAD QUILLACOLLO	CALAMA
76	814501072007569	21	Cochabamba	QUILLACOLLO	QUILLACOLLO - CIUDAD QUILLACOLLO	CALAMA
77	8098015020091590	21	Cochabamba	QUILLACOLLO	QUILLACOLLO - CIUDAD QUILLACOLLO	CALAMA
78	809802172008116	21	Cochabamba	QUILLACOLLO	QUILLACOLLO - CIUDAD QUILLACOLLO	CALAMA
79	8090001520073660	21	Cochabamba	QUILLACOLLO	QUILLACOLLO - CIUDAD QUILLACOLLO	CALAMA
80	509000062007252	21	Cochabamba	QUILLACOLLO	QUILLACOLLO - CIUDAD QUILLACOLLO	CALAMA
81	812301052008172	14	Oruro	Cercado	ORURO	ALEMAN
82	814500412008210	12	Oruro	Cercado	ORURO	JUAN LECHIN OQUENDO SECUNDARIA
83	8090000220079180	6	Cochabamba	Quillacollo	SUTICOLLO	JUAN XXIII
84	8090001520081900	6	Cochabamba	Quillacollo	SUTICOLLO	JUAN XXIII
85	809803912007425	4	Cochabamba	Cercado	COCHABAMBA	FEDERICO FROEBEL
86	80730523200738A	2	La Paz	Murillo	LA PAZ	CUMBRE
87	814800512007292	1	Chuquisaca	Tomina	PADILLA	DEOGRACIAS VEGA
88	8045004420073480	1	Chuquisaca	Tomina	PADILLA	DEOGRACIAS VEGA
89	80730494200762	1	La Paz	Murillo	LA PAZ	AMERICANO NESTOR PEÑARANDA 2
90	817301772009562	1	Tarija	Cercado	TARIJA	HNO. FELIPE PALAZON
91	8173018420082580	1	Tarija	Cercado	TARIJA	HNO. FELIPE PALAZON
92	817301772009809	1	Tarija	Cercado	TARIJA	HNO. FELIPE PALAZON
93	81730181200751A	1	Tarija	Cercado	TARIJA	HNO. FELIPE PALAZON
94	81730177200723000	1	Tarija	Cercado	TARIJA	HNO. FELIPE PALAZON
95	407300172008935	1	La Paz	Murillo	LA PAZ	AMERICANO NESTOR PEÑARANDA 2
96	4073016420081990	1	La Paz	Murillo	EL ALTO	SANTA BARBARA
97	809802082008126A	1	Cochabamba	Quillacollo	SUTICOLLO	JUAN XXIII
98	80980333200729	1	Cochabamba	Cercado	COCHABAMBA	FEDERICO FROEBEL
99	809803292008289	1	Cochabamba	Cercado	COCHABAMBA	PEDRO PAVEDA
100	809803232007495	1	Cochabamba	Cercado	COCHABAMBA	SAN AGUSTIN
101	8045004420073480	1	Chuquisaca	Tomina	PADILLA	DEOGRACIAS VEGA
102	817301772009809	1	Tarija	Cercado	TARIJA	HNO. FELIPE PALAZON
103	81730177200721500	1	Tarija	Cercado	TARIJA	HNO. FELIPE PALAZON
104	809000892008382	0	Cochabamba	QUILLACOLLO	QUILLACOLLO - CIUDAD QUILLACOLLO	NUESTRA SEÑORA DE URCUPIÑA
105	822000082008272	0	Beni	Yacuma	SANTA ANA DEL YACUMA	MONSEÑOR CARLOS ANASAGASTI ZULUETA
106	822000092007240	0	Beni	Yacuma	SANTA ANA DEL YACUMA	MONSEÑOR CARLOS ANASAGASTI ZULUETA
107	819600112009967	0	Cochabamba	Chapare	SACABA	HUGHES SCHOOLS
108	6090004520106120	0	Cochabamba	Chapare	SACABA	HUGHES SCHOOLS
109	809804392007100	0	Cochabamba	Chapare	SACABA	HUGHES SCHOOLS
110	808900262008498	0	Cochabamba	Chapare	CHINATA	FRANZ TAMAYO

111	822201622007324	0	Beni	Vaca Diez	RIBERALTA	MARANATHA
112	80980349200718	0	Cochabamba	Cercado	COCHABAMBA	PEDRO POVEDA
113	80890011200825	0	Cochabamba	Cercado	COCHABAMBA	PEDRO POVEDA
114	809803502008133A	0	Cochabamba	Cercado	COCHABAMBA	FEDERICO FROEBEL
115	809803912007467	0	Cochabamba	Cercado	COCHABAMBA	FEDERICO FROEBEL
116	809803912007222	0	Cochabamba	Cercado	COCHABAMBA	FEDERICO FROEBEL
117	8098040420097380	0	Cochabamba	Cercado	COCHABAMBA	PEDRO POVEDA
118	80480059200754	0	Cochabamba	Cercado	COCHABAMBA	EJERCITO COCHABAMBA
119	80980404200750	0	Cochabamba	Cercado	COCHABAMBA	PEDRO POVEDA
120	80980404200767	0	Cochabamba	Cercado	COCHABAMBA	PEDRO POVEDA
121	8044005920081380	0	Chuquisaca	Tomina	PADILLA	SAN PIO X SECUNDARIA
122	809200172008208	0	Cochabamba	Capinota	CAPINOTA	RODOLFO MONTAÑO VALDERRAMA
123	809000892008382	0	Cochabamba	QUILLACOLLO	QUILLACOLLO - CIUDAD QUILLACOLLO	NUESTRA SEÑORA DE URCUPIÑA
124	408700362007354	0	Cochabamba	Carrasco	VALLE SAJTA	JUAN PABLO II
125	408700362007765	0	Cochabamba	Carrasco	VALLE SAJTA	JUAN PABLO II
126	408700362007221	0	Cochabamba	Carrasco	VALLE SAJTA	JUAN PABLO II
127	4087003620083080	0	Cochabamba	Carrasco	VALLE SAJTA	JUAN PABLO II
128	4087003620083560	0	Cochabamba	Carrasco	VALLE SAJTA	JUAN PABLO II
129	408700362007632	0	Cochabamba	Carrasco	VALLE SAJTA	JUAN PABLO II
130	708700882007113	0	Cochabamba	Carrasco	VALLE SAJTA	JUAN PABLO II
131	4087003620083600	0	Cochabamba	Carrasco	VALLE SAJTA	JUAN PABLO II
132	8098002020094390	0	Cochabamba	Cercado	COCHABAMBA	CENDI
133	809801952008347	0	Cochabamba	Cercado	COCHABAMBA	EJERCITO COCHABAMBA
134	80980349200718	0	Cochabamba	Cercado	COCHABAMBA	PEDRO POVEDA
135	809803502008133A	0	Cochabamba	Cercado	COCHABAMBA	FEDERICO FROEBEL
136	809803912007467	0	Cochabamba	Cercado	COCHABAMBA	FEDERICO FROEBEL
137	809803912007222	0	Cochabamba	Cercado	COCHABAMBA	FEDERICO FROEBEL
138	8098040420097380	0	Cochabamba	Cercado	COCHABAMBA	PEDRO POVEDA
139	80480059200754	0	Cochabamba	Cercado	COCHABAMBA	EJERCITO COCHABAMBA
140	80980404200750	0	Cochabamba	Cercado	COCHABAMBA	PEDRO POVEDA
141	408700362008268A	0	Cochabamba	Carrasco	VALLE SAJTA	JUAN PABLO II
142	4087003620082690	0	Cochabamba	Carrasco	VALLE SAJTA	JUAN PABLO II
143	408700362008353A	0	Cochabamba	Carrasco	VALLE SAJTA	JUAN PABLO II
144	809000892008382	0	Cochabamba	QUILLACOLLO	QUILLACOLLO - CIUDAD QUILLACOLLO	NUESTRA SEÑORA DE URCUPIÑA
145	408700362007354	0	Cochabamba	Carrasco	VALLE SAJTA	JUAN PABLO II
146	408700362007765	0	Cochabamba	Carrasco	VALLE SAJTA	JUAN PABLO II
147	408700362007221	0	Cochabamba	Carrasco	VALLE SAJTA	JUAN PABLO II
148	4087003620083080	0	Cochabamba	Carrasco	VALLE SAJTA	JUAN PABLO II
149	4087003620083560	0	Cochabamba	Carrasco	VALLE SAJTA	JUAN PABLO II
150	408700362007632	0	Cochabamba	Carrasco	VALLE SAJTA	JUAN PABLO II
151	708700882007113	0	Cochabamba	Carrasco	VALLE SAJTA	JUAN PABLO II
152	4087003620083600	0	Cochabamba	Carrasco	VALLE SAJTA	JUAN PABLO II
153	819600112009967	0	Cochabamba	Chapare	SACABA	HUGHES SCHOOLS
154	6090004520106120	0	Cochabamba	Chapare	SACABA	HUGHES SCHOOLS
155	809804392007100	0	Cochabamba	Chapare	SACABA	HUGHES SCHOOLS
156	808900262008498	0	Cochabamba	Chapare	CHIÑATA	FRANZ TAMAYO
157	8089011020081500	0	Cochabamba	Chapare	SACABA	HUGHES SCHOOLS
158	809801952008347	0	Cochabamba	Cercado	COCHABAMBA	EJERCITO COCHABAMBA
159	8098040420097380	0	Cochabamba	Cercado	COCHABAMBA	PEDRO POVEDA
160	80980404200750	0	Cochabamba	Cercado	COCHABAMBA	PEDRO POVEDA
161	708900312008834	0	Cochabamba	Carrasco	VALLE SAJTA	JUAN PABLO II
162	408700362010177A	0	Cochabamba	Carrasco	VALLE SAJTA	JUAN PABLO II
163	4087003620101780	0	Cochabamba	Carrasco	VALLE SAJTA	JUAN PABLO II
164	4087003620083050	0	Cochabamba	Carrasco	VALLE SAJTA	JUAN PABLO II
165	408700362009125A	0	Cochabamba	Carrasco	VALLE SAJTA	JUAN PABLO II
166	408700282008101	0	Cochabamba	Carrasco	VALLE SAJTA	JUAN PABLO II
167	408700362008268A	0	Cochabamba	Carrasco	VALLE SAJTA	JUAN PABLO II
168	4087003620082690	0	Cochabamba	Carrasco	VALLE SAJTA	JUAN PABLO II
169	809000892008382	0	Cochabamba	QUILLACOLLO	QUILLACOLLO - CIUDAD QUILLACOLLO	NUESTRA SEÑORA DE URCUPIÑA
170	4087003620083600	0	Cochabamba	Carrasco	VALLE SAJTA	JUAN PABLO II

171	819600112009967	0	Cochabamba	Chapare	SACABA	HUGHES SCHOOLS
172	6090004520106120	0	Cochabamba	Chapare	SACABA	HUGHES SCHOOLS
173	809804392007100	0	Cochabamba	Chapare	SACABA	HUGHES SCHOOLS
174	809802422008182	0	Cochabamba	Cercado	COCHABAMBA	AYNI PACHA
175	81360020200832	0	Cochabamba	Cercado	COCHABAMBA	AYNI PACHA
176	809803022008643	0	Cochabamba	Cercado	COCHABAMBA	AYNI PACHA
177	80980173200875A	0	Cochabamba	Cercado	COCHABAMBA	AYNI PACHA
178	8098017420078830	0	Cochabamba	Cercado	COCHABAMBA	AYNI PACHA
179	80980404200750	0	Cochabamba	Cercado	COCHABAMBA	PEDRO POVEDA
180	80980404200767	0	Cochabamba	Cercado	COCHABAMBA	PEDRO POVEDA
181	708900312008834	0	Cochabamba	Carrasco	VALLE SAJTA	JUAN PABLO II
182	408700362010177A	0	Cochabamba	Carrasco	VALLE SAJTA	JUAN PABLO II
183	4087003620101780	0	Cochabamba	Carrasco	VALLE SAJTA	JUAN PABLO II
184	4087003620083050	0	Cochabamba	Carrasco	VALLE SAJTA	JUAN PABLO II
185	408700362009125A	0	Cochabamba	Carrasco	VALLE SAJTA	JUAN PABLO II
186	408700282008101	0	Cochabamba	Carrasco	VALLE SAJTA	JUAN PABLO II
187	408700362008268A	0	Cochabamba	Carrasco	VALLE SAJTA	JUAN PABLO II
188	4087003620082690	0	Cochabamba	Carrasco	VALLE SAJTA	JUAN PABLO II
189	408700362008353A	0	Cochabamba	Carrasco	VALLE SAJTA	JUAN PABLO II
190	408700362007354	0	Cochabamba	Carrasco	VALLE SAJTA	JUAN PABLO II
191	408700362007765	0	Cochabamba	Carrasco	VALLE SAJTA	JUAN PABLO II
192	408700362007221	0	Cochabamba	Carrasco	VALLE SAJTA	JUAN PABLO II
193	4087003620083080	0	Cochabamba	Carrasco	VALLE SAJTA	JUAN PABLO II
194	8089011020081500	0	Cochabamba	Chapare	SACABA	HUGHES SCHOOLS
195	809802422008182	0	Cochabamba	Cercado	COCHABAMBA	AYNI PACHA
196	81360020200832	0	Cochabamba	Cercado	COCHABAMBA	AYNI PACHA
197	809803022008643	0	Cochabamba	Cercado	COCHABAMBA	AYNI PACHA
198	80980173200875A	0	Cochabamba	Cercado	COCHABAMBA	AYNI PACHA
199	80730235200737700	0	Cochabamba	Quillacollo	SUTICOLLO	JUAN XXIII
200	8098017420078830	0	Cochabamba	Cercado	COCHABAMBA	AYNI PACHA
201	70470038200817	0	Chuquisaca	Tomina	PADILLA	SAN PIO X SECUNDARIA
202	8044005920081380	0	Chuquisaca	Tomina	PADILLA	SAN PIO X SECUNDARIA
203	809700702008133	0	Cochabamba	Campero	AIQUILE	SIMON RODRIGUEZ
204	809802422008182	0	Cochabamba	Cercado	COCHABAMBA	AYNI PACHA
205	609000452008153	0	Cochabamba	Quillacollo	TIQUIPAYA	TIQUIPAYA
206	80980120200896	0	Cochabamba	Cercado	COCHABAMBA	CENDI
207	80980404200884	0	Cochabamba	Cercado	COCHABAMBA	PEDRO POVEDA
208	80980349200718	0	Cochabamba	Cercado	COCHABAMBA	PEDRO POVEDA
209	80890011200825	0	Cochabamba	Cercado	COCHABAMBA	PEDRO POVEDA
210	809803502008133A	0	Cochabamba	Cercado	COCHABAMBA	FEDERICO FROEBEL
211	809803912007467	0	Cochabamba	Cercado	COCHABAMBA	FEDERICO FROEBEL
212	809803912007222	0	Cochabamba	Cercado	COCHABAMBA	FEDERICO FROEBEL
213	8098040420097380	0	Cochabamba	Cercado	COCHABAMBA	PEDRO POVEDA
214	80480059200754	0	Cochabamba	Cercado	COCHABAMBA	EJERCITO COCHABAMBA
215	80980404200750	0	Cochabamba	Cercado	COCHABAMBA	PEDRO POVEDA
216	80980404200767	0	Cochabamba	Cercado	COCHABAMBA	PEDRO POVEDA
217	6145002720073510	0	Chuquisaca	Oropeza	CORTIJO	DOROTEO HERNANDEZ
218	8043004420092250	0	Chuquisaca	Oropeza	CORTIJO	DOROTEO HERNANDEZ
219	809803462008186	0	Cochabamba	Cercado	COCHABAMBA	SAN AGUSTIN
220	809803522007282	0	Cochabamba	Cercado	COCHABAMBA	SAN AGUSTIN
221	708900312008834	0	Cochabamba	Carrasco	VALLE SAJTA	JUAN PABLO II
222	408700362010177A	0	Cochabamba	Carrasco	VALLE SAJTA	JUAN PABLO II
223	4087003620083050	0	Cochabamba	Carrasco	VALLE SAJTA	JUAN PABLO II
224	408700362009125A	0	Cochabamba	Carrasco	VALLE SAJTA	JUAN PABLO II
225	408700282008101	0	Cochabamba	Carrasco	VALLE SAJTA	JUAN PABLO II
226	408700362008268A	0	Cochabamba	Carrasco	VALLE SAJTA	JUAN PABLO II
227	4087003620082690	0	Cochabamba	Carrasco	VALLE SAJTA	JUAN PABLO II
228	408700362008353A	0	Cochabamba	Carrasco	VALLE SAJTA	JUAN PABLO II
229	408700362007354	0	Cochabamba	Carrasco	VALLE SAJTA	JUAN PABLO II
230	408700362007765	0	Cochabamba	Carrasco	VALLE SAJTA	JUAN PABLO II

231	408700362007221	0	Cochabamba	Carrasco	VALLE SAJTA	JUAN PABLO II
232	809802422008182	0	Cochabamba	Cercado	COCHABAMBA	AYNI PACHA
233	81360020200832	0	Cochabamba	Cercado	COCHABAMBA	AYNI PACHA
234	809803022008643	0	Cochabamba	Cercado	COCHABAMBA	AYNI PACHA
235	80980173200875A	0	Cochabamba	Cercado	COCHABAMBA	AYNI PACHA
236	8098017420078830	0	Cochabamba	Cercado	COCHABAMBA	AYNI PACHA
237	4073010620079440	0	La Paz	Murillo	EL ALTO	NAZARENO BASIL MILLER
238	809803522007282	0	Cochabamba	Cercado	COCHABAMBA	SAN AGUSTIN
239	80730759200753	0	Cochabamba	Cercado	COCHABAMBA	PEDRO POVEDA
240	809803912007906	0	Cochabamba	Cercado	COCHABAMBA	FEDERICO FROEBEL
241	708900312008834	0	Cochabamba	Carrasco	VALLE SAJTA	JUAN PABLO II
242	408700362010177A	0	Cochabamba	Carrasco	VALLE SAJTA	JUAN PABLO II
243	4087003620101780	0	Cochabamba	Carrasco	VALLE SAJTA	JUAN PABLO II
244	4087003620083050	0	Cochabamba	Carrasco	VALLE SAJTA	JUAN PABLO II
245	408700362009125A	0	Cochabamba	Carrasco	VALLE SAJTA	JUAN PABLO II
246	408700282008101	0	Cochabamba	Carrasco	VALLE SAJTA	JUAN PABLO II
247	408700362008268A	0	Cochabamba	Carrasco	VALLE SAJTA	JUAN PABLO II
248	4087003620082690	0	Cochabamba	Carrasco	VALLE SAJTA	JUAN PABLO II
249	408700362008353A	0	Cochabamba	Carrasco	VALLE SAJTA	JUAN PABLO II
250	408700362007354	0	Cochabamba	Carrasco	VALLE SAJTA	JUAN PABLO II
251	408700362007221	0	Cochabamba	Carrasco	VALLE SAJTA	JUAN PABLO II
252	4087003620083080	0	Cochabamba	Carrasco	VALLE SAJTA	JUAN PABLO II
253	4087003620083560	0	Cochabamba	Carrasco	VALLE SAJTA	JUAN PABLO II
254	408700362007632	0	Cochabamba	Carrasco	VALLE SAJTA	JUAN PABLO II
255	708700882007113	0	Cochabamba	Carrasco	VALLE SAJTA	JUAN PABLO II
256	4087003620083600	0	Cochabamba	Carrasco	VALLE SAJTA	JUAN PABLO II
257	81360020200832	0	Cochabamba	Cercado	COCHABAMBA	AYNI PACHA
258	809803022008643	0	Cochabamba	Cercado	COCHABAMBA	AYNI PACHA
259	80980173200875A	0	Cochabamba	Cercado	COCHABAMBA	AYNI PACHA
260	8098017420078830	0	Cochabamba	Cercado	COCHABAMBA	AYNI PACHA
261	708900312008834	0	Cochabamba	Carrasco	VALLE SAJTA	JUAN PABLO II
262	408700362008268A	0	Cochabamba	Carrasco	VALLE SAJTA	JUAN PABLO II
263	4087003620082690	0	Cochabamba	Carrasco	VALLE SAJTA	JUAN PABLO II
264	408700362008353A	0	Cochabamba	Carrasco	VALLE SAJTA	JUAN PABLO II
265	809000892008382	0	Cochabamba	QUILLACOLLO	QUILLACOLLO - CIUDAD QUILLACOLLO	NUUESTRA SEÑORA DE URCUPIÑA
266	408700362007354	0	Cochabamba	Carrasco	VALLE SAJTA	JUAN PABLO II
267	408700362007765	0	Cochabamba	Carrasco	VALLE SAJTA	JUAN PABLO II
268	408700362007221	0	Cochabamba	Carrasco	VALLE SAJTA	JUAN PABLO II
269	4087003620083080	0	Cochabamba	Carrasco	VALLE SAJTA	JUAN PABLO II
270	4087003620083560	0	Cochabamba	Carrasco	VALLE SAJTA	JUAN PABLO II
271	408700362007632	0	Cochabamba	Carrasco	VALLE SAJTA	JUAN PABLO II
272	708700882007113	0	Cochabamba	Carrasco	VALLE SAJTA	JUAN PABLO II
273	4087003620083600	0	Cochabamba	Carrasco	VALLE SAJTA	JUAN PABLO II
274	819600112009967	0	Cochabamba	Chapare	SACABA	HUGHES SCHOOLS
275	6090004520106120	0	Cochabamba	Chapare	SACABA	HUGHES SCHOOLS
276	809802422008182	0	Cochabamba	Cercado	COCHABAMBA	AYNI PACHA
277	81360020200832	0	Cochabamba	Cercado	COCHABAMBA	AYNI PACHA
278	809803022008643	0	Cochabamba	Cercado	COCHABAMBA	AYNI PACHA
279	80980173200875A	0	Cochabamba	Cercado	COCHABAMBA	AYNI PACHA
280	8098017420078830	0	Cochabamba	Cercado	COCHABAMBA	AYNI PACHA
281	407303572007965	0	La Paz	Murillo	EL ALTO	REPUBLICA DE BOLIVIA
282	407302002008436	0	La Paz	Murillo	EL ALTO	REPUBLICA DE BOLIVIA
283	407301672007828	0	La Paz	Murillo	EL ALTO	SANTISIMA TRINIDAD
284	407302342007370	0	La Paz	Murillo	EL ALTO	NAZARENO BASIL MILLER
285	40730303200730	0	La Paz	Murillo	EL ALTO	NAZARENO BASIL MILLER
286	809803912007467	0	Cochabamba	Cercado	COCHABAMBA	FEDERICO FROEBEL
287	809803912007222	0	Cochabamba	Cercado	COCHABAMBA	FEDERICO FROEBEL
288	8098040420097380	0	Cochabamba	Cercado	COCHABAMBA	PEDRO POVEDA
289	40730295200753	0	La Paz	Murillo	EL ALTO	SAN LUIS DE GONZAGA B
290	80980404200750	0	Cochabamba	Cercado	COCHABAMBA	PEDRO POVEDA

291	4073029820077580	0	La Paz	Murillo	EL ALTO	SAGRADO CORAZON DE JESUS
292	80980404200767	0	Cochabamba	Cercado	COCHABAMBA	PEDRO POVEDA
293	4073024620081150	0	La Paz	Murillo	EL ALTO	NAZARENO BASIL MILLER
294	407304472008293	0	La Paz	Murillo	EL ALTO	SANTISIMA TRINIDAD
295	4073017820071940	0	La Paz	Murillo	EL ALTO	UNION EUROPEA B
296	60660014200771	0	La Paz	Murillo	EL ALTO	SANTISIMA TRINIDAD
297	81130006200890	0	La Paz	Murillo	EL ALTO	UNION EUROPEA B
298	4073037420089310	0	La Paz	Murillo	EL ALTO	REPUBLICA DE BOLIVIA
299	8148016320083390	0	La Paz	Murillo	EL ALTO	SAGRADO CORAZON DE JESUS
300	8148016320083400	0	La Paz	Murillo	EL ALTO	SAGRADO CORAZON DE JESUS
301	4087003620083050	0	Cochabamba	Carrasco	VALLE SAJTA	JUAN PABLO II
302	408700362009125A	0	Cochabamba	Carrasco	VALLE SAJTA	JUAN PABLO II
303	408700282008101	0	Cochabamba	Carrasco	VALLE SAJTA	JUAN PABLO II
304	408700362008268A	0	Cochabamba	Carrasco	VALLE SAJTA	JUAN PABLO II
305	809000892008382	0	Cochabamba	QUILLACOLLO	QUILLACOLLO - CIUDAD QUILLACOLLO	NUUESTRA SEÑORA DE URCUPIÑA
306	8090001520073350	0	Cochabamba	Quillacollo	VINTO	EBENEZER
307	809000782008578	0	Cochabamba	Quillacollo	VINTO	EBENEZER
308	407302922010489A	0	La Paz	Murillo	EL ALTO	SANTA BARBARA
309	80730304200992	0	La Paz	Murillo	EL ALTO	NAZARENO BASIL MILLER
310	7062003520098A	0	La Paz	Murillo	EL ALTO	SANTA BARBARA
311	40730249201010300	0	La Paz	Murillo	EL ALTO	NAZARENO BASIL MILLER
312	819600112009967	0	Cochabamba	Chapare	SACABA	HUGHES SCHOOLS
313	6090004520106120	0	Cochabamba	Chapare	SACABA	HUGHES SCHOOLS
314	809804392007100	0	Cochabamba	Chapare	SACABA	HUGHES SCHOOLS
315	808900262008498	0	Cochabamba	Chapare	CHIÑATA	FRANZ TAMAYO
316	8089011020081500	0	Cochabamba	Chapare	SACABA	HUGHES SCHOOLS
317	609000452008153	0	Cochabamba	Quillacollo	TIQUIPAYA	TIQUIPAYA
318	80730235200737700	0	Cochabamba	Quillacollo	SUTICOLLO	JUAN XXIII
319	306100072007217	0	La Paz	Aroma	MARQUIVIRI	TCNL. MAX TOLEDO
320	807200022007335	0	La Paz	Omasuyos	ACHACACHI	MCAL. SANTA CRUZ
321	7094002520101160	0	Cochabamba	Arani	RODEO	RODEO
322	408700362008268A	0	Cochabamba	Carrasco	VALLE SAJTA	JUAN PABLO II
323	4087003620082690	0	Cochabamba	Carrasco	VALLE SAJTA	JUAN PABLO II
324	8090001520073350	0	Cochabamba	Quillacollo	VINTO	EBENEZER
325	809000782008578	0	Cochabamba	Quillacollo	VINTO	EBENEZER
326	407302922010489A	0	La Paz	Murillo	EL ALTO	SANTA BARBARA
327	80730304200992	0	La Paz	Murillo	EL ALTO	NAZARENO BASIL MILLER
328	7062003520098A	0	La Paz	Murillo	EL ALTO	SANTA BARBARA
329	40730249201010300	0	La Paz	Murillo	EL ALTO	NAZARENO BASIL MILLER
330	808900262008498	0	Cochabamba	Chapare	CHIÑATA	FRANZ TAMAYO
331	807000072009229	0	La Paz	Murillo	EL ALTO	REPUBLICA DE BOLIVIA
332	808900262008525	0	Cochabamba	Chapare	CHIÑATA	FRANZ TAMAYO
333	407303572007463	0	La Paz	Murillo	EL ALTO	REPUBLICA DE BOLIVIA
334	8089011020081500	0	Cochabamba	Chapare	SACABA	HUGHES SCHOOLS
335	407303572007965	0	La Paz	Murillo	EL ALTO	REPUBLICA DE BOLIVIA
336	609000452008153	0	Cochabamba	Quillacollo	TIQUIPAYA	TIQUIPAYA
337	407302002008436	0	La Paz	Murillo	EL ALTO	REPUBLICA DE BOLIVIA
338	80730235200737700	0	Cochabamba	Quillacollo	SUTICOLLO	JUAN XXIII
339	306100072007217	0	La Paz	Aroma	MARQUIVIRI	TCNL. MAX TOLEDO
340	807200022007335	0	La Paz	Omasuyos	ACHACACHI	MCAL. SANTA CRUZ
341	7094002520101160	0	Cochabamba	Arani	RODEO	RODEO
342	408700362007765	0	Cochabamba	Carrasco	VALLE SAJTA	JUAN PABLO II
343	8090001520073350	0	Cochabamba	Quillacollo	VINTO	EBENEZER
344	809000782008578	0	Cochabamba	Quillacollo	VINTO	EBENEZER
345	407302922010489A	0	La Paz	Murillo	EL ALTO	SANTA BARBARA
346	80730304200992	0	La Paz	Murillo	EL ALTO	NAZARENO BASIL MILLER
347	7062003520098A	0	La Paz	Murillo	EL ALTO	SANTA BARBARA
348	40730249201010300	0	La Paz	Murillo	EL ALTO	NAZARENO BASIL MILLER
349	807000072009229	0	La Paz	Murillo	EL ALTO	REPUBLICA DE BOLIVIA
350	407303572007463	0	La Paz	Murillo	EL ALTO	REPUBLICA DE BOLIVIA

351	407303572007965	0	La Paz	Murillo	EL ALTO	REPUBLICA DE BOLIVIA
352	407302002008436	0	La Paz	Murillo	EL ALTO	REPUBLICA DE BOLIVIA
353	407301672007828	0	La Paz	Murillo	EL ALTO	SANTISIMA TRINIDAD
354	407302342007370	0	La Paz	Murillo	EL ALTO	NAZARENO BASIL MILLER
355	40730303200730	0	La Paz	Murillo	EL ALTO	NAZARENO BASIL MILLER
356	807302262007129	0	La Paz	Murillo	EL ALTO	SANTISIMA TRINIDAD
357	609000452009989	0	Cochabamba	Quillacollo	TIQUIPAYA	TIQUIPAYA
358	8098036220093860	0	Cochabamba	Quillacollo	TIQUIPAYA	TIQUIPAYA
359	306100072007217	0	La Paz	Aroma	MARQUIVIRI	TCNL. MAX TOLEDO
360	807200022007335	0	La Paz	Omasuyos	ACHACACHI	MCAL. SANTA CRUZ
361	8090001520073350	0	Cochabamba	Quillacollo	VINTO	EBENEZER
362	809000782008578	0	Cochabamba	Quillacollo	VINTO	EBENEZER
363	407302922010489A	0	La Paz	Murillo	EL ALTO	SANTA BARBARA
364	80730304200992	0	La Paz	Murillo	EL ALTO	NAZARENO BASIL MILLER
365	7062003520098A	0	La Paz	Murillo	EL ALTO	SANTA BARBARA
366	40730249201010300	0	La Paz	Murillo	EL ALTO	NAZARENO BASIL MILLER
367	807000072009229	0	La Paz	Murillo	EL ALTO	REPUBLICA DE BOLIVIA
368	407303572007463	0	La Paz	Murillo	EL ALTO	REPUBLICA DE BOLIVIA
369	407303572007965	0	La Paz	Murillo	EL ALTO	REPUBLICA DE BOLIVIA
370	407302002008436	0	La Paz	Murillo	EL ALTO	REPUBLICA DE BOLIVIA
371	407301672007828	0	La Paz	Murillo	EL ALTO	SANTISIMA TRINIDAD
372	407302342007370	0	La Paz	Murillo	EL ALTO	NAZARENO BASIL MILLER
373	40730303200730	0	La Paz	Murillo	EL ALTO	NAZARENO BASIL MILLER
374	807302262007129	0	La Paz	Murillo	EL ALTO	SANTISIMA TRINIDAD
375	306100072007217	0	La Paz	Aroma	MARQUIVIRI	TCNL. MAX TOLEDO
376	4073010620079440	0	La Paz	Murillo	EL ALTO	NAZARENO BASIL MILLER
377	40730295200753	0	La Paz	Murillo	EL ALTO	SAN LUIS DE GONZAGA B
378	4073029820077580	0	La Paz	Murillo	EL ALTO	SAGRADO CORAZON DE JESUS
379	807200022007335	0	La Paz	Omasuyos	ACHACACHI	MCAL. SANTA CRUZ
380	4073024620081150	0	La Paz	Murillo	EL ALTO	NAZARENO BASIL MILLER
381	40730249201010300	0	La Paz	Murillo	EL ALTO	NAZARENO BASIL MILLER
382	807000072009229	0	La Paz	Murillo	EL ALTO	REPUBLICA DE BOLIVIA
383	407303572007463	0	La Paz	Murillo	EL ALTO	REPUBLICA DE BOLIVIA
384	407303572007965	0	La Paz	Murillo	EL ALTO	REPUBLICA DE BOLIVIA
385	407302002008436	0	La Paz	Murillo	EL ALTO	REPUBLICA DE BOLIVIA
386	407301672007828	0	La Paz	Murillo	EL ALTO	SANTISIMA TRINIDAD
387	407302342007370	0	La Paz	Murillo	EL ALTO	NAZARENO BASIL MILLER
388	40730303200730	0	La Paz	Murillo	EL ALTO	NAZARENO BASIL MILLER
389	807302262007129	0	La Paz	Murillo	EL ALTO	SANTISIMA TRINIDAD
390	4073010620079440	0	La Paz	Murillo	EL ALTO	NAZARENO BASIL MILLER
391	40730295200753	0	La Paz	Murillo	EL ALTO	SAN LUIS DE GONZAGA B
392	4073029820077580	0	La Paz	Murillo	EL ALTO	SAGRADO CORAZON DE JESUS
393	4073024620081150	0	La Paz	Murillo	EL ALTO	NAZARENO BASIL MILLER
394	407304472008293	0	La Paz	Murillo	EL ALTO	SANTISIMA TRINIDAD
395	4073017820071940	0	La Paz	Murillo	EL ALTO	UNION EUROPEA B
396	60660014200771	0	La Paz	Murillo	EL ALTO	SANTISIMA TRINIDAD
397	81130006200890	0	La Paz	Murillo	EL ALTO	UNION EUROPEA B
398	4073037420089310	0	La Paz	Murillo	EL ALTO	REPUBLICA DE BOLIVIA
399	8148016320083390	0	La Paz	Murillo	EL ALTO	SAGRADO CORAZON DE JESUS
400	8148016320083400	0	La Paz	Murillo	EL ALTO	SAGRADO CORAZON DE JESUS
401	407302822007402	0	La Paz	Murillo	EL ALTO	ANDRES BELLO TARDE
402	70650013200778	0	La Paz	Murillo	EL ALTO	ANDRES BELLO TARDE
403	407301672007828	0	La Paz	Murillo	EL ALTO	SANTISIMA TRINIDAD
404	407302342007370	0	La Paz	Murillo	EL ALTO	NAZARENO BASIL MILLER
405	40730303200730	0	La Paz	Murillo	EL ALTO	NAZARENO BASIL MILLER
406	807302262007129	0	La Paz	Murillo	EL ALTO	SANTISIMA TRINIDAD
407	4073010620079440	0	La Paz	Murillo	EL ALTO	NAZARENO BASIL MILLER
408	40730295200753	0	La Paz	Murillo	EL ALTO	SAN LUIS DE GONZAGA B
409	4073029820077580	0	La Paz	Murillo	EL ALTO	SAGRADO CORAZON DE JESUS
410	4073024620081150	0	La Paz	Murillo	EL ALTO	NAZARENO BASIL MILLER

411	407304472008293	0	La Paz	Murillo	EL ALTO	SANTISIMA TRINIDAD
412	4073017820071940	0	La Paz	Murillo	EL ALTO	UNION EUROPEA B
413	60660014200771	0	La Paz	Murillo	EL ALTO	SANTISIMA TRINIDAD
414	81130006200890	0	La Paz	Murillo	EL ALTO	UNION EUROPEA B
415	4073037420089310	0	La Paz	Murillo	EL ALTO	REPUBLICA DE BOLIVIA
416	8148016320083390	0	La Paz	Murillo	EL ALTO	SAGRADO CORAZON DE JESUS
417	8148016320083400	0	La Paz	Murillo	EL ALTO	SAGRADO CORAZON DE JESUS
418	407300822007336	0	La Paz	Murillo	EL ALTO	SANTISIMA TRINIDAD
419	4073023420071040	0	La Paz	Murillo	EL ALTO	NAZARENO BASIL MILLER
420	407303572007533	0	La Paz	Murillo	EL ALTO	NAZARENO BASIL MILLER
421	407302822007402	0	La Paz	Murillo	EL ALTO	ANDRES BELLO TARDE
422	70650013200778	0	La Paz	Murillo	EL ALTO	ANDRES BELLO TARDE
423	8073069220075740	0	La Paz	Murillo	EL ALTO	CATOLICO SAN AGUSTIN
424	407303932007147	0	La Paz	Murillo	EL ALTO	ANDRES BELLO TARDE
425	807300202008230	0	La Paz	Murillo	EL ALTO	SAN JUAN BAUTISTA DE LA SALLE
426	407302342007370	0	La Paz	Murillo	EL ALTO	NAZARENO BASIL MILLER
427	40730303200730	0	La Paz	Murillo	EL ALTO	NAZARENO BASIL MILLER
428	807302262007129	0	La Paz	Murillo	EL ALTO	SANTISIMA TRINIDAD
429	4073010620079440	0	La Paz	Murillo	EL ALTO	NAZARENO BASIL MILLER
430	40730295200753	0	La Paz	Murillo	EL ALTO	SAN LUIS DE GONZAGA B
431	4073029820077580	0	La Paz	Murillo	EL ALTO	SAGRADO CORAZON DE JESUS
432	4073024620081150	0	La Paz	Murillo	EL ALTO	NAZARENO BASIL MILLER
433	407304472008293	0	La Paz	Murillo	EL ALTO	SANTISIMA TRINIDAD
434	4073017820071940	0	La Paz	Murillo	EL ALTO	UNION EUROPEA B
435	60660014200771	0	La Paz	Murillo	EL ALTO	SANTISIMA TRINIDAD
436	81130006200890	0	La Paz	Murillo	EL ALTO	UNION EUROPEA B
437	4073037420089310	0	La Paz	Murillo	EL ALTO	REPUBLICA DE BOLIVIA
438	407300822007336	0	La Paz	Murillo	EL ALTO	SANTISIMA TRINIDAD
439	4073023420071040	0	La Paz	Murillo	EL ALTO	NAZARENO BASIL MILLER
440	407303572007533	0	La Paz	Murillo	EL ALTO	NAZARENO BASIL MILLER
441	407301392007652	0	La Paz	Murillo	EL ALTO	CEP BOLIVIA
442	407304472008293	0	La Paz	Murillo	EL ALTO	SANTISIMA TRINIDAD
443	60660014200771	0	La Paz	Murillo	EL ALTO	SANTISIMA TRINIDAD
444	81130006200890	0	La Paz	Murillo	EL ALTO	UNION EUROPEA B
445	4073037420089310	0	La Paz	Murillo	EL ALTO	REPUBLICA DE BOLIVIA
446	8148016320083390	0	La Paz	Murillo	EL ALTO	SAGRADO CORAZON DE JESUS

Anexo B – Examen de la 4ta Etapa (Nacional Sucre 2017)

URL de referencia: <http://www.titancod.net/problemas>

Problema 1 - Fibonacci y Pitágoras

Por Jorge Teran

El famoso teorema de Pitágoras indica que la hipotenusa c de un triángulo rectángulo de lados a, b se puede calcular con la fórmula $c^2 = a^2 + b^2$. Este teorema se ha hecho tan famoso que muchos números se han denominado pitagóricos si pueden hallarse como la suma de dos números enteros elevados al cuadrado.

¿Qué tiene que ver Fibonacci con Pitágoras?. Bien primero recordemos que la sucesión de Fibonacci se define Matemáticamente con la ecuación $f(n)=f(n-1)+f(n-2)$. Los primeros números de la serie son: 0,1,1,2,3,5,8,13..etc.

Tratando de ver si se pueden encontrar números pitagóricos formados por exclusivamente números de Fibonacci, se ve lo siguiente: El primer número pitagórico que se puede hallar es el número $1 = 0^2 + 1^2$, el segundo es el $2 = 1^2 + 1^2$, así se forma una secuencia de números que son de Fibonacci y pitagóricos, simultáneamente. Los primeros elementos de esta serie son: 1, 2, 5, 13.... todos formados exclusivamente con números de Fibonacci.

Entrada

La entrada consiste de varios casos de prueba. La primera línea indica cuantos casos de prueba existen. Cada caso de prueba contiene un número, que indica la posición en la secuencia de los números Fibonacci pitagóricos que queremos hallar.

Salida

Imprima en la salida el número Pitagórico indicado.

Ejemplo de Entrada

4
0
1
2
3

Ejemplo de Salida

1
2
5
13

Anexo B – Examen de la 4ta Etapa (Nacional Sucre 2017)

Problema 2 - Cinta Multicolor

Por María Leticia Blanco Coca

Está de moda las cosas multi, y en esta ocasión se tienen las cintas multicolor, estas cintas tienen una peculiaridad y es que nueve colores están presentes en la misma. Vienen en diferentes anchos y los colores se presentan de forma sesgada y cada color está representado por un dígito. Un ejemplo se observa a continuación:

1111
1112
1122
1222
2222
2223
2233
2333
3333
3334
3344
3444
4444
4445
4455
4555
5555
5556
5566
5666
6666

....
....

Esta es una cinta de ancho 4, para efectos de nuestro problema la cinta puede ser tan larga como te imagines. Marita se está preguntando que colores estarán presentes si corta la cinta de largo n . En el ejemplo si se hace un corte de largo 12 los colores en los que termina la cinta serán 3444 pero si se corta de una longitud 41 los colores son 2222. Pues la tarea es ayudar a Marita a saber en qué colores termina la cinta si se tiene un corte de largo n

Entrada

Se tiene varios casos de prueba, cada caso de prueba se encuentra en una línea que consta de dos valores a y c el ancho y el largo de corte de la cinta respectivamente.

Los casos terminan cuando ya no hay más datos.

Salida

Por cada caso de prueba debes imprimir en una línea los colores en los que termina la cinta.

Ejemplo de Entrada

4 41
5 36
9 5
41 15

Ejemplo de Salida

2222
88888
111112222
11111111111111111111111111112222222222222222

Anexo B – Examen de la 4ta Etapa (Nacional Sucre 2017)

Problema 3 - Sub Suma

Por Jorge Teran

Dado un vector V se quiere hallar la suma de un grupo de elementos contiguos cuya suma sea un valor dado S . No se puede re ordenar el vector.

Por ejemplo, si se tiene un vector con 6 elementos:

1	2	3	4	5	6
2	8	2	6	3	5

Ahora se quiere hallar cuales son los elementos cuya suma es 16, se observa que los elementos 2,3,4 suman 16 ($8 + 2 + 6 = 16$). También los elementos $2 + 6 + 3 + 5 = 16$. Sin embargo, la respuesta correcta es 2, 4 que significa que debemos sumar los valores desde la posición 2 hasta la 4 inclusive.

Entrada

La entrada consiste de varios datos de prueba. Cada caso de prueba consiste de dos líneas. La primera línea tiene dos números, el primero es el número de elementos del vector ($0 \leq n \leq 107$) y el segundo es la suma ($1 \leq s \leq 1014$) que de elementos contiguos que busquemos.

La segunda línea contiene los n elementos del vector separados por un espacio.

Los datos de entrada terminan cuando no hay más datos.

Salida

En la salida imprima dos números que representan la posición del primer elemento y el último elemento del arreglo, cuya suma de los elementos entre estos valores es la suma buscada. En caso de que no exista imprima -1.

Ejemplo de Entrada

6 16
2 8 2 6 3 5
7 5

1 4 4 1 0 0 5
8 6
7 1 0 9 4 9 2 6
9 3 2
7 9 3 6 4 3 2 5 2
10 4
2 5 7 5 8 4 9 2 6 9
Ejemplo de Salida
2 4
1 2
8 8
1 6
6 6

Anexo B – Examen de la 4ta Etapa (Nacional Sucre 2017)

Problema 4 - Equilibrio

Por Maria Leticia Blanco Coca

Los números tienen varias propiedades, una de ellas es la primalidad, es así que todos los primos son equilibrados en sus factores, ya que el factor primo menor y mayor que los divide es el mismo número. Pero lo curioso es que hay algunos números que también son equilibrados sin ser primos; por ejemplo, el 81, su factor primo menor y mayor es 3, ya que no tiene otro factor primo.

Tu tarea es, decidir si un número tiene equilibrio, caso contrario indicar el factor primo menor y mayor del número.

Entrada

La primera línea contiene un número entero positivo c que indica cuántos casos de prueba existirán.

En seguida se tienen c líneas, cada una tiene n que es el número del cual se quiere saber si tiene equilibrio o no.

Salida

Para cada caso se debe imprimir en una línea el factor mínimo y el factor máximo del número en caso de no tener equilibrio; caso contrario se debe imprimir *Equilibrio*.

El formato de la salida para los números que no tienen equilibrio es $[factorMin, factorMax]$.

Ejemplo de Entrada

9
5

27

87

80

45

60

90

42

52

Ejemplo de Salida

Equilibrio

Equilibrio

[3,29]

[2,5]

[3,5]

[2,5]

[2,5]

[2,7]

[2,13]



ANEXO C

Calificaciones Etapa 3 (Departamental 2017)

URL de referencia: <http://www.titancod.net/obi2017/posicionesdep?nivel=2&pag=1>

Núm.	RUDE	Nota	Tiempo	Departamento	Provincia	Localidad	U. Educativa
1	812300692007264	208	28691s.	La Paz	Murillo	LA PAZ (CON MÉTODO V-BLOOM)	SAGRADOS CORAZONES A
2	8073010320071720	192	18427s.	La Paz	Murillo	LA PAZ (CON MÉTODO V-BLOOM)	DON BOSCO A
3	814801682008728	185	17059s.	Potosí	Tomas Frias	POTOSI	SANTA MARIA
4	80730460201025000	118	18241s.	La Paz	Murillo	LA PAZ (CON MÉTODO V-BLOOM)	AMERINST
5	807303972007119	107	13771s.	La Paz	Murillo	LA PAZ (CON MÉTODO V-BLOOM)	SAGRADOS CORAZONES A
6	81480168200877	106	11828s.	La Paz	Murillo	LA PAZ (CON MÉTODO V-BLOOM)	ROSA GATTORNO
7	80730397200734	82	15700s.	La Paz	Murillo	LA PAZ (CON MÉTODO V-BLOOM)	SAGRADOS CORAZONES A
8	812301842007928	61	3184s.	Oruro	Cercado	ORURO	ALEMAN
9	8123019220104390	59	803s.	Oruro	Cercado	ORURO	IGNACIO LEON 2
10	8198115320081910	53	13s.	Santa Cruz	Andres Ibañez	SANTA CRUZ DE LA SIERRA	ALEMAN
11	813800522007107	22	20426s.	Potosí	Tomas Frias	POTOSI	LITORAL C
12	81981333200811200	16	11548s.	Santa Cruz	Andres Ibañez	SANTA CRUZ DE LA SIERRA	EVANGELICO BUENAS NUEVAS D
13	819806362007745	16	14369s.	Santa Cruz	Andres Ibañez	SANTA CRUZ DE LA SIERRA	EVANGELICO BUENAS NUEVAS B
14	819813662008521	8	3735s.	Santa Cruz	Andres Ibañez	SANTA CRUZ DE LA SIERRA	CRISTO REY
15	804500452008260	8	10094s.	Chuquisaca	Tomina	PADILLA	SAN PIO X SECUNDARIA
16	8073066420102590	4	6647s.	La Paz	Murillo	LA PAZ (CON MÉTODO V-BLOOM)	SAN IGNACIO
17	7044002820073550	4	8944s.	Chuquisaca	Tomina	PADILLA	SAN PIO X SECUNDARIA
18	4073016020081620	3	6674s.	La Paz	Murillo	EL ALTO	SANTA BARBARA
19	814801682008305	3	10625s.	Potosí	Tomas Frias	POTOSI	SANTA MARIA
20	8123006920071290	2	3997s.	Oruro	Cercado	ORURO	JESUS MARIA 2
21	81230235200729	2	18901s.	Oruro	Cercado	ORURO	JESUS MARIA 2
22	809803912007425	1	1884s.	Cochabamba	Cercado	COCHABAMBA	FEDERICO FROEBEL
23	809803292008289	1	2370s.	Cochabamba	Cercado	COCHABAMBA	PEDRO POVEDA
24	8223007920092500	1	3245s.	Beni	Cercado	TRINIDAD	LA SALLE
25	812302422008104	1	4561s.	Oruro	Cercado	ORURO	ALEMAN
26	807300792008934	1	4668s.	La Paz	Murillo	LA PAZ (CON MÉTODO V-BLOOM)	INGLES CATOLICO
27	812302422008124	1	5817s.	Oruro	Cercado	ORURO	ALEMAN
28	819806362009128	1	6259s.	Santa Cruz	Andres Ibañez	SANTA CRUZ DE LA SIERRA	EVANGELICO BUENAS NUEVAS D
29	812300692007404	1	7678s.	Oruro	Cercado	ORURO	ALEMAN
30	814800302008500	1	7952s.	Potosí	Tomas Frias	POTOSI	DON BOSCO A
31	812301052008172	1	7960s.	Oruro	Cercado	ORURO	ALEMAN
32	817301772009809	1	8671s.	Tarija	Cercado	TARIJA	HNO. FELIPE PALAZON
33	814801862008209	1	9497s.	Potosí	Tomas Frias	POTOSI	SANTA MARIA
34	814800522007104A	1	9513s.	Potosí	Tomas Frias	POTOSI	LITORAL C
35	814801572008262	1	9593s.	Potosí	Tomas Frias	POTOSI	DON BOSCO A
36	812301842007629	1	9688s.	Oruro	Cercado	ORURO	ALEMAN
37	407300172008935	1	9808s.	La Paz	Murillo	LA PAZ	AMERICANO NESTOR PEÑARANDA 2
38	8223007920092810	1	10536s.	Beni	Cercado	TRINIDAD	LA SALLE
39	807303592008448	0	0s.	La Paz	Murillo	LA PAZ (CON MÉTODO V-BLOOM)	MARIA AUXILIADORA I
40	40730347200881	0	0s.	La Paz	Murillo	La Paz	Santa Barbara
41	812200752007255	0	0s.	Oruro	Abaroa	CHALLAPATA	SAN JUAN DE DIOS
42	61470003200896	0	0s.	Oruro	Abaroa	CHALLAPATA	SAN JUAN DE DIOS
43	8045004420073480	0	0s.	Chuquisaca	Tomina	PADILLA	DEOGRACIAS VEGA
44	807303592008448	0	0s.	La Paz	Murillo	LA PAZ	MARIA AUXILIADORA I
45	4073033520075220	0	0s.	La Paz	Murillo	EL ALTO	PUERTO DE MEJILLONES
46	4073028320071930	0	0s.	La Paz	Murillo	EL ALTO	PUERTO DE MEJILLONES
47	40730347200881	0	0s.	La Paz	Murillo	La Paz	Santa Barbara
48	407303352007529A	0	0s.	La Paz	Murillo	EL ALTO	PUERTO DE MEJILLONES
49	80980333200729	0	0s.	Cochabamba	Cercado	COCHABAMBA	FEDERICO FROEBEL
50	8098035320089040	0	0s.	Cochabamba	Cercado	COCHABAMBA	CENDI

51	7117000720091990	0	0s.	Oruro	Pantaleon Dalence	MACHACAMARCA	REPUBLICA DE MEXICO
52	40650005200944	0	0s.	Oruro	Pantaleon Dalence	MACHACAMARCA	PEDRO DOMINGO MURILLO 2
53	711700062009193	0	0s.	Oruro	Pantaleon Dalence	MACHACAMARCA	PEDRO DOMINGO MURILLO 2
54	407302362007763	0	0s.	La Paz	Murillo	EL ALTO	PUERTO DE MEJILLONES
55	812200692008693	0	0s.	Oruro	Abaroa	CHALLAPATA	SAN JUAN DE DIOS
56	81220075200889	0	0s.	Oruro	Abaroa	CHALLAPATA	SAN JUAN DE DIOS
57	812200752008656	0	0s.	Oruro	Abaroa	CHALLAPATA	SAN JUAN DE DIOS
58	81220053200835	0	0s.	Oruro	Abaroa	CHALLAPATA	SANTA ROSA
59	812200782007686	0	0s.	Oruro	Abaroa	CHALLAPATA	SAN JUAN DE DIOS
60	812200752014211000	0	0s.	Oruro	Abaroa	CHALLAPATA	SAN JUAN DE DIOS
61	812200752007255	0	0s.	Oruro	Abaroa	CHALLAPATA	SAN JUAN DE DIOS
62	61470003200896	0	0s.	Oruro	Abaroa	CHALLAPATA	SAN JUAN DE DIOS
63	5064008220073570	0	0s.	Oruro	CERCADO	CARACOLLO	SIMON BOLIVAR C
64	7123002820091010	0	0s.	Oruro	CERCADO	CARACOLLO	SIMON BOLIVAR C
65	712300282007159	0	0s.	Oruro	CERCADO	CARACOLLO	SIMON BOLIVAR C
66	7123002820091230	0	0s.	Oruro	CERCADO	CARACOLLO	SIMON BOLIVAR C
67	807303592008448	0	0s.	La Paz	Murillo	LA PAZ	MARIA AUXILIADORA I
68	7117000720091990	0	0s.	Oruro	Pantaleon Dalence	MACHACAMARCA	REPUBLICA DE MEXICO
69	407302362007763	0	0s.	La Paz	Murillo	EL ALTO	PUERTO DE MEJILLONES
70	812200782007639	0	0s.	Oruro	Abaroa	CHALLAPATA	SAN JUAN DE DIOS
71	812200782008610	0	0s.	Oruro	Abaroa	CHALLAPATA	SAN JUAN DE DIOS
72	81220078200797A	0	0s.	Oruro	Abaroa	CHALLAPATA	SAN JUAN DE DIOS
73	812200802008175	0	0s.	Oruro	Abaroa	CHALLAPATA	SANTA ROSA
74	812200692008693	0	0s.	Oruro	Abaroa	CHALLAPATA	SAN JUAN DE DIOS
75	81220075200889	0	0s.	Oruro	Abaroa	CHALLAPATA	SAN JUAN DE DIOS
76	812200752008656	0	0s.	Oruro	Abaroa	CHALLAPATA	SAN JUAN DE DIOS
77	8122007820071580	0	0s.	Oruro	Abaroa	CHALLAPATA	SAN JUAN DE DIOS
78	81220053200835	0	0s.	Oruro	Abaroa	CHALLAPATA	SANTA ROSA
79	812200782007686	0	0s.	Oruro	Abaroa	CHALLAPATA	SAN JUAN DE DIOS
80	812200752014211000	0	0s.	Oruro	Abaroa	CHALLAPATA	SAN JUAN DE DIOS
81	812200752007255	0	0s.	Oruro	Abaroa	CHALLAPATA	SAN JUAN DE DIOS
82	61470003200896	0	0s.	Oruro	Abaroa	CHALLAPATA	SAN JUAN DE DIOS
83	5064008220073570	0	0s.	Oruro	CERCADO	CARACOLLO	SIMON BOLIVAR C
84	71230009200873	0	0s.	Oruro	CERCADO	CARACOLLO	SIMON BOLIVAR C
85	7123002820091010	0	0s.	Oruro	CERCADO	CARACOLLO	SIMON BOLIVAR C
86	712300282007159	0	0s.	Oruro	CERCADO	CARACOLLO	SIMON BOLIVAR C
87	7123002820091230	0	0s.	Oruro	CERCADO	CARACOLLO	SIMON BOLIVAR C
88	80980333200729	0	0s.	Cochabamba	Cercado	COCHABAMBA	FEDERICO FROEBEL
89	7117000720091990	0	0s.	Oruro	Pantaleon Dalence	MACHACAMARCA	REPUBLICA DE MEXICO
90	40650005200944	0	0s.	Oruro	Pantaleon Dalence	MACHACAMARCA	PEDRO DOMINGO MURILLO 2
91	812200782007639	0	0s.	Oruro	Abaroa	CHALLAPATA	SAN JUAN DE DIOS
92	812200782008610	0	0s.	Oruro	Abaroa	CHALLAPATA	SAN JUAN DE DIOS
93	81220078200797A	0	0s.	Oruro	Abaroa	CHALLAPATA	SAN JUAN DE DIOS
94	812200802008175	0	0s.	Oruro	Abaroa	CHALLAPATA	SANTA ROSA
95	812200692008693	0	0s.	Oruro	Abaroa	CHALLAPATA	SAN JUAN DE DIOS
96	81220075200889	0	0s.	Oruro	Abaroa	CHALLAPATA	SAN JUAN DE DIOS
97	8122007820071580	0	0s.	Oruro	Abaroa	CHALLAPATA	SAN JUAN DE DIOS
98	81220053200835	0	0s.	Oruro	Abaroa	CHALLAPATA	SANTA ROSA
99	812200782007686	0	0s.	Oruro	Abaroa	CHALLAPATA	SAN JUAN DE DIOS
100	812200752014211000	0	0s.	Oruro	Abaroa	CHALLAPATA	SAN JUAN DE DIOS

101	812301332008210	0	0s.	Oruro	Cercado	ORURO	ALEMAN
102	8123018520085760	0	0s.	Oruro	Cercado	ORURO	ALEMAN
103	81230184200766	0	0s.	Oruro	Cercado	ORURO	ALEMAN
104	81230184200772	0	0s.	Oruro	Cercado	ORURO	ALEMAN
105	812301052008386	0	0s.	Oruro	Cercado	ORURO	JESUS MARIA 2
106	812301462008951	0	0s.	Oruro	Cercado	ORURO	PANTALEON DALENCE 1
107	814500412008210	0	0s.	Oruro	Cercado	ORURO	JUAN LECHIN OQUENDO SECUNDARIA
108	81230184200725A	0	0s.	Oruro	Cercado	ORURO	ALEMAN
109	8123006920073660	0	0s.	Oruro	Cercado	ORURO	ALEMAN
110	4073016420081990	0	0s.	La Paz	Murillo	EL ALTO	SANTA BARBARA
111	4073033520075210	0	0s.	La Paz	Murillo	EL ALTO	PUERTO DE MEJILLONES
112	822201782009990	0	0s.	La Paz	Murillo	EL ALTO	PUERTO DE MEJILLONES
113	4073033520075220	0	0s.	La Paz	Murillo	EL ALTO	PUERTO DE MEJILLONES
114	4073028320071930	0	0s.	La Paz	Murillo	EL ALTO	PUERTO DE MEJILLONES
115	407303352007529A	0	0s.	La Paz	Murillo	EL ALTO	PUERTO DE MEJILLONES
116	40650005200944	0	0s.	Oruro	Pantaleon Dalence	MACHACAMARCA	PEDRO DOMINGO MURILLO 2
117	711700062009193	0	0s.	Oruro	Pantaleon Dalence	MACHACAMARCA	PEDRO DOMINGO MURILLO 2
118	407302362007763	0	0s.	La Paz	Murillo	EL ALTO	PUERTO DE MEJILLONES
119	812200782008610	0	0s.	Oruro	Abaroa	CHALLAPATA	SAN JUAN DE DIOS
120	711700072007405	0	0s.	Oruro	Pantaleon Dalence	MACHACAMARCA	REPUBLICA DE MEXICO
121	812301462008951	0	0s.	Oruro	Cercado	ORURO	PANTALEON DALENCE 1
122	812200752008656	0	0s.	Oruro	Abaroa	CHALLAPATA	SAN JUAN DE DIOS
123	8122007820071580	0	0s.	Oruro	Abaroa	CHALLAPATA	SAN JUAN DE DIOS
124	81220053200835	0	0s.	Oruro	Abaroa	CHALLAPATA	SANTA ROSA
125	812200782007686	0	0s.	Oruro	Abaroa	CHALLAPATA	SAN JUAN DE DIOS



ANEXO D

Lista de clasificados del departamento de La Paz, informática (Nivel 2) delegación La Paz

Informática	Nivel 2	ORO	CABRERA	GORDILLO	FABRICIO JAVIER	LA PAZ 2	SAGRADOS CORAZONES A
Informática	Nivel 2	PLATA	PACAJES	YUJRA	GUIDO GONZALO	LA PAZ 2	DON BOSCO A
Informática	Nivel 2	BRONCE	TONCONI	MENDOZA	DANER ZEIN	LA PAZ 1	AMERINST
Informática	Nivel 2	MENCION	ROJAS	MORALES	AMBAR MONICA	LA PAZ 2	SAGRADOS CORAZONES A
Informática	Nivel 2	MENCION	CALLAPA	QUILLA	LUIS ALEJANDRO	LA PAZ 2	ROSA GATTORNO
Informática	Nivel 2	MENCION	ENRIQUEZ	CORRALES	ROGER DANILO	LA PAZ 2	SAGRADOS CORAZONES A
Informática	Nivel 2	MENCION	VILLENA	ORTIZ	SEBASTIAN RODRIGO	LA PAZ 2	SAN IGNACIO

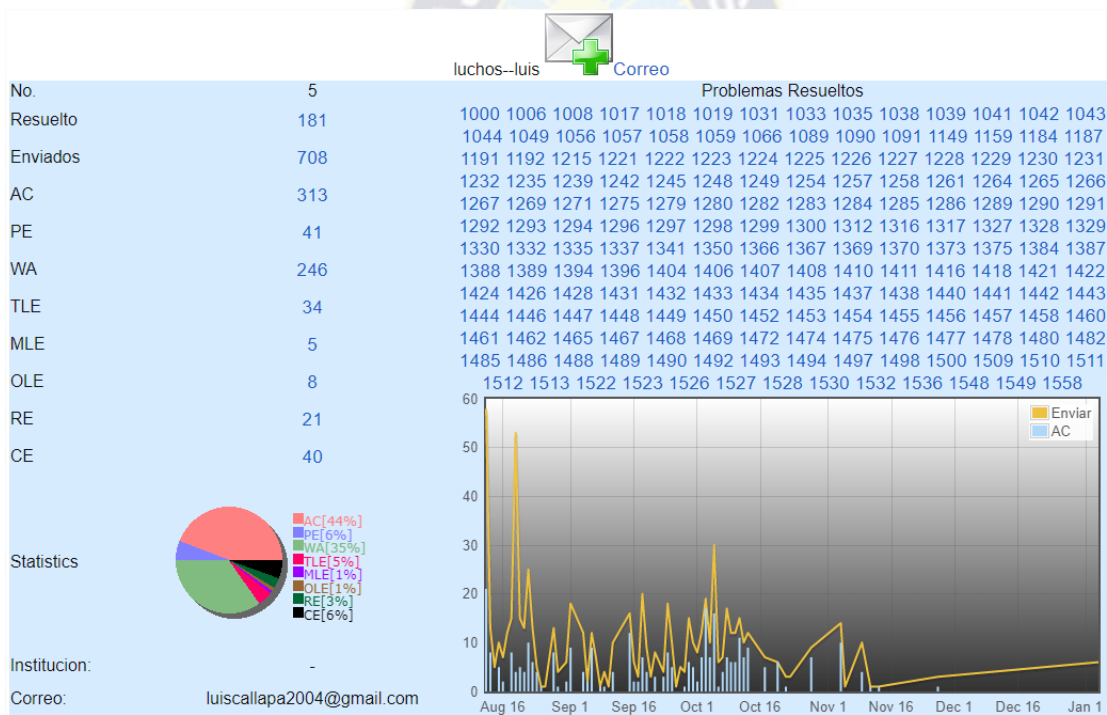
2017.

Fuente: 7ma. Olimpiada Científica Estudiantil Plurinacional Boliviana (Medallero departamental)

URL: <http://olimpiadas.educabolivia.bo/public/files/7ma/3ra/LaPaz.pdf>

Anexo E

Reporte de rendimiento de Juez Virtual – “Universidad Mayor de San Andrés”.

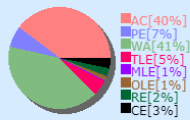




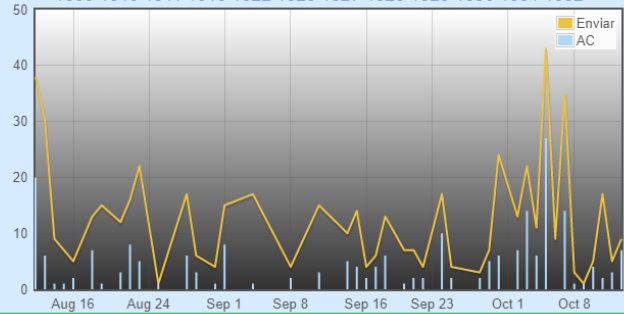
gpacajes--guido Correo

No.	10	Problemas Resueltos																											
Resuelto	152	1000	1006	1008	1017	1018	1019	1033	1035	1038	1039	1041	1042	1043	1044	1049	1056	1057	1058	1059	1066	1089	1090	1091	1147	1149	1157	1159	1184
Enviados	544	1187	1191	1192	1215	1221	1223	1224	1225	1226	1227	1228	1229	1230	1231	1235	1245	1248	1249	1254	1257	1261	1262	1264	1267	1269	1271	1275	1280
AC	216	1283	1284	1285	1286	1289	1290	1291	1292	1293	1296	1297	1298	1299	1312	1317	1327	1329	1332	1337	1341	1344	1367	1370	1373	1375	1385	1387	1388
PE	37	1389	1394	1396	1404	1406	1407	1411	1416	1418	1421	1422	1425	1426	1431	1432	1434	1435	1437	1438	1440	1442	1443	1444	1446	1447	1448	1449	1452
WA	224	1453	1454	1455	1456	1457	1458	1460	1461	1462	1467	1469	1472	1474	1476	1477	1478	1482	1484	1485	1486	1487	1488	1489	1490	1491	1492	1493	1500
TLE	26	1509 1510 1511 1513 1522 1526 1527 1528 1529 1530 1531 1532																											
MLE	3																												
OLE	7																												
RE	13																												
CE	18																												

Statistics



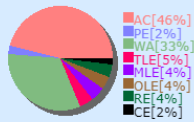
Institucion: -
Correo: gonzalopacajes@gmail.com



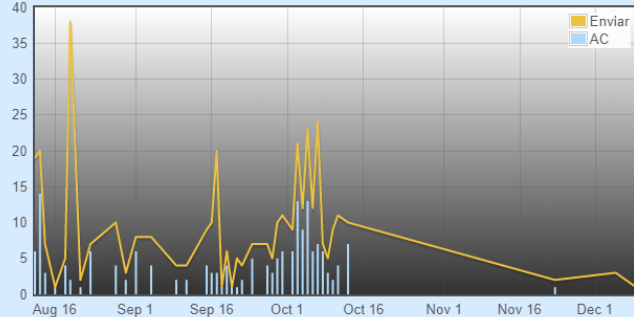


Amy_on_ice--Ambar Correo

No.	18	Problemas Resueltos																											
Resuelto	136	1000	1006	1008	1018	1019	1031	1033	1035	1038	1041	1042	1043	1044	1056	1057	1058	1059	1063	1066	1089	1090	1091	1124	1157	1184	1187	1215	1221
Enviados	381	1222	1223	1224	1225	1226	1227	1228	1229	1230	1231	1232	1235	1244	1245	1249	1253	1254	1261	1264	1265	1266	1267	1269	1271	1275	1280	1283	1284
AC	176	1285	1289	1290	1291	1292	1294	1296	1297	1298	1299	1312	1314	1316	1322	1329	1330	1332	1335	1351	1366	1369	1373	1375	1384	1388	1389	1396	1406
PE	9	1421	1422	1424	1428	1429	1431	1432	1434	1437	1440	1442	1443	1446	1449	1450	1452	1453	1454	1455	1456	1458	1460	1461	1467	1472	1474	1477	1478
WA	124	1482	1485	1486	1488	1489	1490	1492	1493	1500	1509	1511	1519	1520	1522	1525	1526	1527	1528	1529	1530	1531	1532	1536	1546				
TLE	18																												
MLE	17																												
OLE	15																												
RE	15																												
CE	7																												

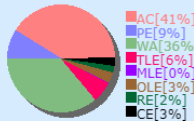


Institucion: -
Correo: ambarrojasm@gmail.com

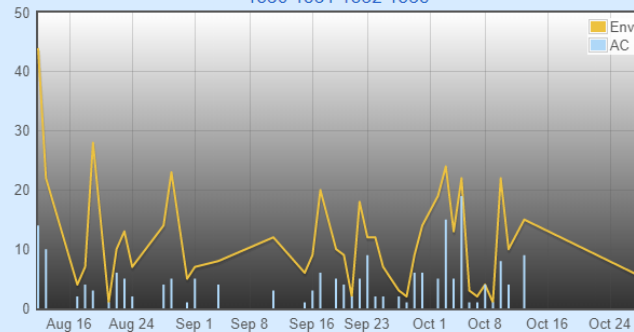


DanerZein--Daner Zein Correo

No.	14	Problemas Resueltos																											
Resuelto	144	1000	1008	1018	1019	1033	1038	1039	1041	1042	1043	1044	1049	1056	1057	1058	1059	1066	1089	1090	1091	1147	1149	1157	1159	1184	1187	1192	1215
Enviados	477	1221	1223	1224	1225	1226	1227	1228	1229	1230	1231	1235	1242	1245	1249	1254	1257	1259	1261	1262	1264	1265	1267	1269	1271	1275	1279	1280	1283
AC	196	1284	1285	1286	1289	1292	1296	1297	1298	1299	1312	1317	1323	1328	1332	1337	1375	1385	1388	1396	1404	1406	1407	1410	1416	1418	1421	1422	1428
PE	42	1429	1431	1432	1434	1437	1440	1441	1442	1443	1444	1445	1446	1447	1448	1449	1452	1453	1454	1455	1456	1457	1458	1460	1461	1462	1467	1468	1469
WA	172	1472	1474	1478	1480	1482	1484	1485	1486	1487	1488	1489	1490	1491	1492	1493	1494	1497	1498	1500	1509	1511	1513	1522	1523	1526	1527	1528	1529
TLE	29																												
MLE	1																												
OLE	16																												
RE	9																												
CE	12																												



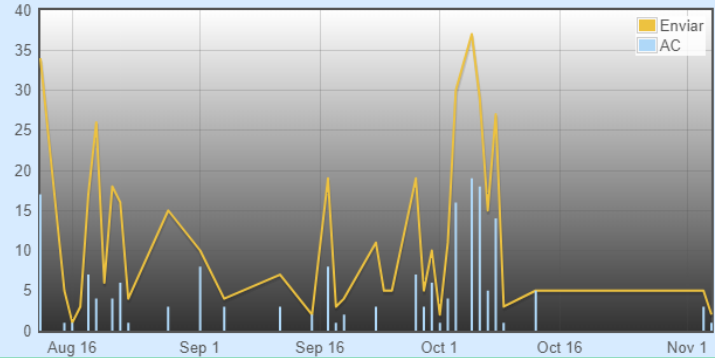
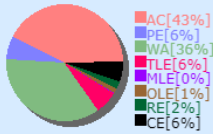
Institucion: -
Correo: noctow551@gmail.com





chubyxdx--fabricio Correo

No.	17	Problemas Resueltos																											
Resuelto	141	1000	1006	1008	1018	1019	1031	1035	1038	1041	1042	1043	1044	1049	1056	1057	1058	1059	1066	1089	1090	1091	1139	1149	1162	1187	1191	1221	1222
Enviados	415	1223	1224	1225	1226	1227	1228	1229	1230	1231	1232	1235	1245	1248	1249	1254	1257	1262	1266	1267	1269	1271	1275	1280	1282	1283	1285	1289	1290
AC	177	1292	1294	1296	1297	1298	1299	1312	1315	1317	1323	1327	1328	1329	1330	1332	1341	1366	1372	1373	1375	1384	1385	1387	1388	1389	1394	1396	1410
PE	25	1411	1418	1423	1424	1426	1427	1428	1431	1433	1434	1435	1437	1438	1442	1443	1447	1450	1452	1454	1455	1456	1462	1467	1472	1474	1475	1476	1477
WA	149	1478	1482	1484	1485	1486	1487	1488	1489	1490	1491	1492	1493	1494	1500	1509	1510	1511	1512	1513	1519	1520	1522	1523	1526	1527	1528	1529	1530
TLE	26	1532																											
MLE	1																												
OLE	6																												
RE	8																												
CE	23																												

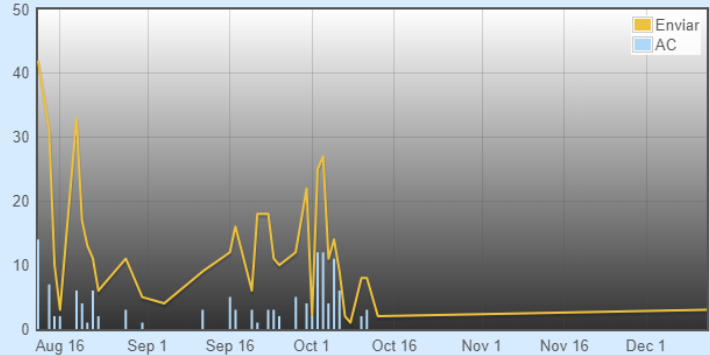
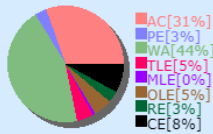


Institucion: -
Correo: fabriciojavierfedehotmail.com



Masterdan--danilo Correo

No.	25	Problemas Resueltos																											
Resuelto	106	1000	1006	1017	1018	1019	1035	1038	1041	1042	1043	1044	1056	1058	1066	1089	1090	1091	1162	1187	1192	1215	1221	1222	1223	1224	1225	1226	1227
Enviados	432	1228	1229	1230	1231	1232	1235	1245	1254	1257	1261	1262	1265	1267	1269	1271	1280	1283	1284	1285	1286	1289	1290	1292	1294	1296	1297	1298	1299
AC	132	1384	1385	1387	1388	1396	1404	1406	1407	1418	1421	1422	1424	1426	1431	1434	1437	1440	1442	1443	1449	1452	1454	1455	1456	1457	1461	1467	1472
PE	14	1474	1476	1477	1482	1485	1486	1488	1489	1490	1491	1492	1500	1509	1511	1513	1522	1526	1527	1528	1530	1532	1536						
WA	191																												
TLE	22																												
MLE	2																												
OLE	23																												
RE	13																												
CE	35																												

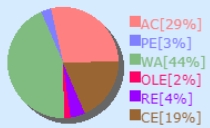


Institucion: -
Correo: rogergatoenriquez@gmail.com



sebasvillena2005--sebastian Correo

No.	93	Problemas Resueltos															
Resuelto	48	1000	1018	1019	1031	1033	1049	1056	1058	1091	1215	1221	1222	1223	1224		
Enviados	221	1225	1226	1227	1228	1229	1230	1235	1245	1265	1267	1269	1275	1283	1296		
AC	63	1297	1298	1317	1327	1384	1394	1396	1421	1424	1472	1485	1487	1492	1500		
PE	6																
WA	97																
OLE	4																
RE	9																
CE	42																



Statistics

Institucion:

Correo: sebasvillena2005@gmail.com



Anexo F

Acta de calificaciones fase 4 (Nacional Sucre 2017), asignación de medallas y menciones de honor.

ACTA DE RESULTADOS

En la ciudad de SUCRE, el día 14 de octubre después de 4 horas continuas de evaluación en el área de informática desde horas 11:00 a 15:00 en **NIVEL 2**; se tienen los siguientes resultados:

NOMBRE	APELLIDO PATERNO	APELLIDO MATERNO	DEPARTAMENTO	COLEGIO	LUGAR	NOTA (400)
AMBAR MONICA	ROJAS	MORALES	LA PAZ V-Bloom	SAGRADOS CORAZONES A	ORO	290
DANER ZEIN	TONCONI	MENDOZA	LA PAZ V-Bloom	AMERINST	PLATA	190
GUIDO GONZALO	PACAJES	YUJRA	LA PAZ V-Bloom	DON BOSCO A	PLATA	180
FABRICIO JAVIER	CABRERA	GORDILLO	LA PAZ V-Bloom	SAGRADOS CORAZONES A	BRONCE	150
LUIS ALEJANDRO	CALLAPA	QUILLA	LA PAZ V-Bloom	ROSA GATTORNO	BRONCE	150
LAURA	MERTENS	VELASQUEZ	SANTA CRUZ	ALEMAN	BRONCE	140
ROGER DANILO	ENRIQUEZ	CORRALES	LA PAZ V-Bloom	SAGRADOS CORAZONES A	BRONCE	70
CARLOS ANDRES	BRACAMONTE	ESPINOZA	POTOSI	SANTA MARIA	MENCION DE HONOR	50
EDWARD OSVALDO	GOMEZ	QUISPE	POTOSI	SANTA MARIA	MENCION DE HONOR	30
JHON SAMUEL	VEDIA	FAJARDO	POTOSI	SANTA MARIA	MENCION DE HONOR	30
SEBASTIAN RODRIGO	VILLENA	ORTIZ	LA PAZ V-Bloom	SAN IGNACIO	MENCION DE HONOR	30

En constancia de estos resultados firman al pie de la presente acta los miembros del comité académico nacional de informática.

SUCRE, 14 DE OCTUBRE DE 2017

Lic. Octavio Aguilar
TARIJA

Lic. Jorge Teran Pomier
LA PAZ

Ing. Juan Choque Uño
ORURO

Lic. Henry Villarroel T.
COCHABAMBA

Lic. Samuel Fuentes C.
PANDO

Ing. Evans Balcazar V.
SANTA CRUZ

Ing. Luis Antonio Molina
CHUQUISACA

Ing. Maria Yusara Farah Chavez
BENI

Lic. Maria Leticia Blanco
COMITÉ ACADÉMICO NACIONAL INFORMÁTICA