

UNIVERSIDAD TÉCNICA DE BABAHOYO

Facultad de Administración, Finanzas e Informática

Escuela en Sistemas



Tesis de Grado, previo a la Obtención del Título de:

INGENIERIA EN SISTEMAS

TEMA:

EL SOFTWARE LIBRE EN LA EVALUACION ACADEMICA A NIÑOS DE LOS PRIMEROS AÑOS DE EDUCACION BASICA.

AUTORA:

MAYRA ALEJANDRA LLERENA GOMEZ

DIRECTOR DE TESIS:

ING. RAUL RAMOS

BABAHOYO

2012

AGRADECIMIENTOS

Dando gracias....

... a Dios por haberme hecho sano normal y capaz

... a mis Padres por ser las personas que más creen y confían en mí, porque
Nunca les han faltado palabras de aliento, sabios consejos y su
Apoyo incondicional...

... a mis Profesores por enseñarme a ver la vida desde distintas
Perspectivas...

...y a mis amigos quienes me han apoyado siempre cuando los
He necesitado...

Mayra Llerena Gómez.

DEDICATORIA

Dedico este trabajo a mis Padres que me han enseñado a Perseguir los sueños hasta hacerlos realidad...
...a mis Hermanos quien siempre me han motivado y me han dado su amor ilimitado...
... y como no a todos mis amigos, por todos aquellos momentos Vividos de sacrificios y de regocijos...

Mayra Llerena Gómez

CERTIFICACIÓN DE AUDITORIA

Las ideas, hechos y doctrinas expuesto en esta tesis me corresponden exclusivamente, y el patrimonio intelectual de la misma a la Escuela de Sistemas de la Universidad Técnica de Babahoyo.

Por la cual certifico que el presente trabajo fue desarrollado por Mayra Alejandra Llerena Gómez bajo mi supervisión.

Firma:

Ing. Raúl Ramos
DIRECTOR DE PROYECTO

FIRMA DE RESPONSABILIDAD

Yo Mayra Alejandra Llerena Gómez declaro bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que hemos consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedemos nuestros derechos de propiedad intelectual correspondientes a este trabajo, a la Universidad Técnica de Babahoyo, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

La calificación de _____

Equivalente a _____

Fecha _____

Firman para corroborar su veracidad:

Presidente Tribunal de Defensa _____

Director de Tesis _____

Lector de Tesis _____

Secretario _____

CAPITULO I

EL PROBLEMA

1.1 PLANTEAMIENTO DEL PROBLEMA.

La educación es un derecho de todo ser humano y un deber social, la cual está fundamentada en el respeto, con la finalidad de desarrollar el potencial creativo de cada ser humano y el pleno ejercicio de su personalidad; para un mejor desenvolvimiento dentro de una sociedad basada en los valores éticos, el trabajo, la participación activa.

Desde el Ministerio de Educación Nacional se dice que una educación es de calidad en la medida en que los estudiantes logren los objetivos propuestos, o alcancen lo que se espera de ellos; en otras palabras: “que aprendan lo que tienen que aprender, en el momento en que lo tienen que aprender y que lo hagan en felicidad”.

Es por esta razón, que los sistemas de evaluación del país han centran su atención y sus esfuerzos en conocer el logro cognitivo de los estudiantes, bien sea en términos de conocimientos, competencias, habilidades del pensamiento, saberes, actitudes, etc.; pues se puede decir que en el aprendizaje de los estudiantes se “resume” o se debe reflejar la mayor parte de todo el quehacer educativo, de todo el trabajo escolar, ya que de muy poco sirve contar con docentes muy preparados, con un plan de estudios bien diseñado, con buenas instalaciones físicas y con excelentes recursos didácticos, si los estudiantes no aprenden, o si escasamente alcanzan las metas básicas propuestas. Aunque no todas las evaluaciones están orientadas a medir exactamente lo mismo, la mayoría de países coinciden en evaluar las áreas centrales del currículo: Lenguaje, Matemáticas, Ciencias Naturales y Ciencias Sociales, principalmente. En algunos casos también se evalúan campos no cognitivos como el desarrollo personal y social, aunque estas evaluaciones han tenido básicamente un carácter experimental y, por decirlo de alguna manera, se encuentran en su mayoría en fase de exploración en los países en los que se han implantado los sistemas de evaluación.

La evaluación del desarrollo y de los aprendizajes de los niños en el Nivel Inicial; específicamente entre los tres y los seis años, es concebida como un proceso permanente de valorización cualitativa de sus potencialidades y de los aprendizajes adquiridos, así como de las condiciones que lo afectan; siendo este proceso individualizado y en función de unos criterios previamente establecidos, referidos al aprendizaje y capacidades que deben desarrollar los niños en esta etapa. En este sentido, la evaluación es el principal instrumento del educador para tomar decisiones curriculares, ya que suministra información, tanto en lo referente a la marcha general

del proyecto educativo, institucional, como al desempeño del maestro y al proceso de desarrollo y aprendizaje de los niños.

Para ciertos autores este sentido de la evaluación en el Nivel Preescolar o Inicial se corresponde con el modelo holístico porque coincide con los planteamientos de los métodos que en éste modelo se basan; destacándose los siguientes aspectos:

- Se busca más bien la descripción y la interpretación, tomando como referencia los contextos en que se desarrollan los programas o aspectos a evaluar.
- No basta con prestar atención sólo a los resultados alcanzados, sino que se deben considerar aquellos objetivos no alcanzados, las dificultades surgidas. Se atiende tanto a los factores internos como a los externos.
- La mayor parte de los instrumentos son de carácter cualitativo: observación, entrevista, diálogo; así como las estrategias utilizadas.

La acción educativa en el Preescolar no puede estar dirigida a lograr objetivos de tipo formal o académico, sino a desarrollar las capacidades del niño; en este sentido la evaluación debe estar dirigida a comprobar en qué medida ese desarrollo va siguiendo una secuencia evolutiva deseada (evaluación cualitativa), no a la asignación de una puntuación de acuerdo a la ejecución del niño (cuantitativa). No quiere decir esto que en Preescolar no se deben hacer “mediciones” de tipo cuantitativo. Aspectos tales como talla y peso necesariamente son cuantitativos; y en otras áreas del desarrollo

también hay ciertos aspectos en los cuales podemos asignar letras o números (G. Veracoechea, 1994).

Cuando se trabaja con ciertas escalas en los instrumentos de registro o de observaciones que utilizamos en Preescolar, podemos asignar una letra o número a la ejecución del niño, la cual va a tener una significación cualitativa; en otros casos, la medición en sí implica un elemento cuantitativo, por ejemplo: “..Recuerda 4 de los 6 objetos de la lámina presentada”, “cuenta hasta 10 objetos imitando al adulto”, etc. Ambos tipos de mediciones van a dar al docente información acerca del niño, pero lo importante desde el punto de vista evaluativo es cómo el docente interprete y analice dichas mediciones basándose para ello en el conocimiento que tenga sobre los aspectos y procesos del desarrollo infantil. Estos tipos de mediciones van a proporcionar elementos para emitir el juicio evaluativo.

En cuanto a los parámetros de evaluación, en Preescolar utilizamos parámetros de comparación o de referencia, los cuales son: Pautas evolutivas y los Objetivos de la acción educativa. Las pautas evolutivas están referidas al conjunto de conductas características del niño en los diferentes niveles de edad, las cuales pueden ser consideradas como los patrones evolutivos típicos de un grupo de niños a una edad determinada y están clasificadas por áreas del desarrollo.

Además de este aspecto, es importante desde el punto de vista docente, tomar en consideración que en un grupo de niños, a pesar de que pueden tener todos la misma edad cronológica y de que existen las pautas evolutivas mencionadas anteriormente, también existen diferencias individuales y que el ritmo de aprendizaje va a ser distinto

entre estos niños, observándose ciertos límites de edad dentro de los cuales la aparición de determinadas conductas puede ser considerada normal.

En la Parroquia Balsapamba del Cantón San Miguel Prov. Bolívar se encuentra funcionando la Escuela Fiscal Mixta Manuel de Jesús Calle, ubicado junto a la vía Flores, la misma que tiene más de 200 alumnos de los cuales alrededor de 60 están cursando el primero y segundo año de Educación Básica. Como es una Institución pública no ha recibido el apoyo adecuado de parte del gobierno, es por ello que no cuenta con la ayuda de las nuevas tecnologías para el proceso educativo.

En base a las visitas realizadas se ha podido observar las siguientes deficiencias especialmente relacionadas con la evaluación académica en los niños de los primeros y segundos años de básica:

- Los tutores no tienen variantes en cuanto a las evaluaciones que realizan a los niños de los primeros años de básica.
- No se tiene el apoyo de las tecnologías (tics) en el proceso devaluatorio académico a nivel preescolar.
- Los niños llegan temerosos y nerviosos al proceso evaluatorio tradicional que se lleva a cabo en la escuela.
- Durante el proceso de evaluación muchos de los niños se distraen fácilmente dificultando con ello dicho proceso.

- El trabajo del profesor se aumenta el momento de volver a evaluar a niños con bajo aprovechamiento.

2.1.1 Delimitación del Problema.

Objeto de Estudio: Ingeniería en Sistemas.

Campo de Acción: El campo de acción está definido directamente al software libre orientado a la evaluación académica

2.1.2 Formulación del problema.

¿Cómo mejorar el proceso de evaluación académica en los niños de los primeros y segundos años de básica en la escuela Fiscal Mixta Manuel de Jesús Calle?

1.3.1 OBJETIVOS.

3.1 Objetivo General

Elaborar un estudio teórico-práctico de varios paquetes de software libre orientados al ámbito educativo, específicamente en el campo de la evaluación académica para mejorar el proceso evaluativo en los niños de los primeros años de básica en la escuela Fiscal Mixta Manuel de Jesús Calle.

3.2 Objetivos Específicos

- Analizar fuentes bibliográficas referentes al software libre orientado a la educación, específicamente al campo evaluativo y así como también sobre la evaluación académica en los primeros años de básica.

- Diagnosticar los procesos actuales de evaluación académica en los primeros y segundos años de básica en la escuela Fiscal Mixta Manuel de Jesús Calle.
- Desarrollar un conjunto de evaluaciones académicas automatizadas en algunos programas de software libre.

1.3 JUSTIFICACIÓN

El problema encontrado en la escuela fiscal mixta “Manuel de Jesús Calle” es apreciable en el momento de las evaluaciones a los niños por el simple y sencillo hecho que los alumnos se encuentran nerviosos y temerosos por no saber cómo estará desarrollado el examen.

Por otra parte la labor del docente se ve incrementada notablemente ya sea en la preparación de las pruebas y en las revisiones de las mismas, lo cual hace que la problemática sea doble por una parte el alumno y por otra parte la labor del docente.

Por esta razón hemos visto la necesidad de manifestar una solución viable la misma que será un portal web, para que los niños den sus exámenes de una manera rápida y sobretodo divertido para ellos con lo cual se tendrá mayor interés en el aprendizaje y mejor desempeño en todas las materias que se dicten a los pequeños. En cuanto a la labor del docente se facilitaría por el motivo que será menor el tiempo en realizar el

cuestionario de evaluaciones y las revisiones de los mismos luego de efectuadas dichas pruebas por los alumnos.

Por lo tanto el maestro tendría mayor tiempo para preparar sus siguientes clases y solucionar con esto una de las grandes problemáticas de la educación en el sector público. Por otro lado la utilización de un software libre significa que no se incurrirá en gastos de licenciamiento tanto para la escuela ni los padres de familia, ya que en las escuelas fiscales se educan en su mayor parte niños de clase humilde.

CAPITULO II

MARCOTEORICO

2.1 ANTECEDENTES INVESTIGATIVOS.

Partiendo del hecho, de que lá evaluación es un aspecto fundamental dentro del proceso educativo, se ha considerado pertinente lá realización de este trabajo investigativo relacionado específicamente con la evaluación que se realiza a los niños.

En una investigación realizada en la biblioteca de UNIANDES se ha podido encontrar algunas tesis relacionados con el proceso educativo en general pero no con el proceso evaluatorio, así tenemos el trabajo realizado por Ing. Francisco Lalama, con su tema “Aplicación multimedia para el proceso enseñanza aprendizaje de los alumnos de primer año de básica del colegio La Sallé y también la tesis realizada por Ing. Luis

Figuroa con su tema “Aplicación multimedia para la enseñanza de geografía política”, de estos trabajos se puede concluir que el proceso educativo se ve fortalecido por el apoyo de la informática.

Cabe mencionar que no se ha encontrado ninguna tesis relacionada con la evaluación académica, lo cual implica que este trabajo investigativo está dotado de originalidad y pertinencia. En cuanto a los trabajos encontrados nos servirán como antecedente para desarrollar la presente tesis.

2.2 FUNDAMENTACION CIENTIFICA.

Este trabajo investigativo se fundamentará científicamente en los siguientes temas:

2.2.1 Software libre

El software libre (en inglés free software, esta denominación también se confunde a veces con gratis por el doble sentido del inglés free en castellano) es la denominación del software que respeta la libertad de los usuarios sobre su producto adquirido y, por tanto, una vez obtenido puede ser usado, copiado, estudiado, cambiado y redistribuido libremente. Según la Free Software Foundation, el software libre se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, modificar el software y distribuirlo modificado.

El software libre suele estar disponible gratuitamente, o al precio de costo de la distribución a través de otros medios; sin embargo no es obligatorio que sea así, por lo tanto no hay que asociar software libre a "software gratuito" (denominado usualmente

freeware), ya que, conservando su carácter de libre, puede ser distribuido comercialmente ("software comercial"). Análogamente, el "software gratis" o "gratuito" incluye en ocasiones el código fuente; no obstante, este tipo de software no es libre en el mismo sentido que el software libre, a menos que se garanticen los derechos de modificación y redistribución de dichas versiones modificadas del programa. (Software Libre - Carlos Gradin 2004)

Tampoco debe confundirse software libre con "software de dominio público". Éste último es aquel software que no requiere de licencia, pues sus derechos de explotación son para toda la humanidad, porque pertenece a todos por igual. Cualquiera puede hacer uso de él, siempre con fines legales y consignando su autoría original. Este software sería aquel cuyo autor lo dona a la humanidad o cuyos derechos de autor han expirado, tras un plazo contado desde la muerte de este, habitualmente 70 años. Si un autor condiciona su uso bajo una licencia, por muy débil que sea, ya no es del dominio público.

Historia

Entre los años 60 y 70 del Siglo XX, el software no era considerado un producto sino un añadido que los vendedores de las grandes computadoras de la época² (las mainframes) aportaban a sus clientes para que éstos pudieran usarlos. En dicha cultura, era común que los programadores y desarrolladores de software compartieran libremente sus programas unos con otros.

Allá por el 1971, cuando la informática todavía no había sufrido su gran boom, las personas que hacían uso de ella, en ámbitos universitarios y empresariales, creaban y compartían el software sin ningún tipo de restricciones.

Este comportamiento era particularmente habitual en algunos de los mayores grupos de usuarios de la época, como DECUS (grupo de usuarios de computadoras DEC). A finales de los 70, las compañías iniciaron el hábito de imponer restricciones a los usuarios, con el uso de acuerdos de licencia. (Luis Lucas Ibáñez)

Con la llegada de los años 80 la situación empezó a cambiar. Las computadoras más modernas comenzaban a utilizar sistemas privativos, forzando a los usuarios a aceptar condiciones restrictivas que impedían realizar modificaciones a dicho software.

En caso de que algún usuario o programador encontrase algún error en la aplicación, lo único que podía hacer era darlo a conocer a la empresa desarrolladora para que esta lo solucionara. Aunque el programador estuviese capacitado para solucionar el problema y lo deseara hacer sin pedir nada a cambio, el contrato le impedía que mejorase el software.

La pérdida de tiempo era constante, ya que en ocasiones, los trabajadores enviaban por red sus trabajos a imprimir y al ir a buscarlos se encontraban la impresora atascada y una cola enorme de trabajos pendientes. ³ Richard Stallman decidió arreglar el problema, e implementar el envío de un aviso por red cuando la impresora se bloqueara. Para ello necesitaba tener acceso al código fuente de los controladores de la impresora. Pidió a la empresa propietaria de la impresora lo que necesitaba, comentando, sin pedir

nada a cambio, qué era lo que pretendía realizar. La empresa se negó a entregarle el código fuente.

Richard Stallman explica que por aquellos años, en el laboratorio donde trabajaba, habían recibido una impresora donada por una empresa externa. El dispositivo, que era utilizado en red por todos los trabajadores, parecía no funcionar a la perfección, dado que cada cierto tiempo el papel se atascaba. Como agravante, no se generaba ningún aviso que se enviase por red e informase a los usuarios de la situación. (Carlos Gradin)

En ese preciso instante, Richard Stallman se vio en una encrucijada: debía elegir entre aceptar el nuevo ⁴software privativo firmando acuerdos de no revelación y acabar desarrollando más software privativo con licencias restrictivas, que a su vez deberían ser más adelante aceptadas por sus propios colegas.

Con este antecedente, en 1984, Richard Stallman comenzó a trabajar en el proyecto GNU, y un año más tarde fundó la Free Software Foundation (FSF). Stallman introdujo la definición de free software y el concepto de "copyleft", que desarrolló para otorgar libertad a los usuarios y para restringir las posibilidades de apropiación del software.

Libertades del software libre.- De acuerdo con tal definición, el software es "libre" si garantiza las siguientes libertades:

El software privativo (también llamado propietario, de código cerrado o software no libre) es cualquier programa informático en el que el usuario tiene limitaciones para usarlo, modificarlo o redistribuirlo y

Libertad	Descripción
0	La libertad de usar el programa, con cualquier propósito.

1	La libertad de estudiar cómo funciona el programa y modificarlo, adaptándolo a tus necesidades.
2	La libertad de distribuir copias del programa, con lo cual puedes ayudar a tu prójimo.
3	La libertad de mejorar el programa y hacer públicas esas mejoras a los demás, de modo que toda la comunidad se beneficie.
Las libertades 1 y 3 requieren acceso al código fuente porque estudiar y modificar software sin su código fuente es muy poco viable.	

El último punto justifica parcialmente las limitaciones impuestas por la licencia GNU GPL frente a otras licencias de software libre. El sentido original es más libre, abierto y menos restrictivo que el que le otorga la propia situación de incompatibilidad, que podría ser resuelta en la próxima versión 3.0 de la licencia GNU GPL causa en estos momentos graves perjuicios a la comunidad de programadores de software libre, que muchas veces no pueden reutilizar o mezclar códigos de dos licencias distintas.

Código Fuente: la remezcla es un anagrama, porque todos los elementos existen en una relación simultánea, por ello en su interior nada es primero y nada es posterior, nada es futuro o pasado, nada es viejo y nada es nuevo (Los Sistemas De Información Y Su Influencia En Los Estándares De Calidad de Escalera Izquierdo, Gregorio y García Rojo, Elena)

El término ⁶software no libre se emplea para referirse al software distribuido bajo una licencia de software más restrictiva que no garantiza estas cuatro libertades. Las leyes de la propiedad intelectual, reservan la mayoría de los derechos de modificación, duplicación y redistribución para el dueño del copyright; el software dispuesto bajo una licencia de software libre rescinde específicamente la mayoría de estos derechos reservados.

Para evitar la confusión, algunas personas utilizan los términos "libre" (software libre) y "gratis" (⁶software gratis) para evitar la ambigüedad de la palabra inglesa "free". Sin embargo, estos términos alternativos son usados únicamente dentro del movimiento del software libre, aunque están extendiéndose lentamente hacia el resto del mundo. Otros defienden el uso del término open source software (software de código abierto). La principal diferencia entre los términos "open source" y "free software" es que éste último tiene en cuenta los aspectos éticos y filosóficos de la libertad, mientras que el "open source" se basa únicamente en los aspectos técnicos.

En un intento por unir los mencionados términos que se refieren a conceptos semejantes, se está extendiendo el uso de la palabra "FLOSS" con el significado de free/libre and open source software e, indirectamente, también a la comunidad que lo produce y apoya.

Para entender la situación de este tipo de software con respecto a su uso en seguridad informática es imprescindible describir, en primer lugar, a qué se refiere este documento cuando hace referencia a "software libre". (Software Libre y No Libre - Octavio Kulesz) (Software Libre - Carlos Gradin 2004) El software gratis puede incluir restricciones que no se adaptan a la definición de software libre, es decir puede no incluir el código

fuente, puede prohibir explícitamente a los distribuidores recibir una compensación a cambio. (Universidad Trashumante-Territorios, redes, lenguajes)

Tipos de licencias

Una licencia es aquella autorización formal con carácter contractual que un autor de un software da a un interesado para ejercer "actos de explotación legales". Pueden existir tantas licencias como acuerdos concretos se den entre el autor y el licenciatarario. Desde el punto de vista del software libre, existen distintas variantes del concepto o grupos de licencias:

Licencias GPL

Una de las más utilizadas es la Licencia Pública General de GNU (GNU GPL). El autor conserva los derechos de autor (copyright), y permite la redistribución y modificación bajo términos diseñados para asegurarse de que todas las versiones modificadas del software permanecen bajo los términos más restrictivos de la propia GNU GPL.

La definición de software libre no contempla el asunto del precio; un eslogan frecuentemente usado es "libre como en libertad, no como en cerveza gratis" o en inglés "Free as in freedom, not as in free beer" (aludiendo a la ambigüedad del término inglés "free"), y es habitual ver a la venta CD de software libre como distribuciones Linux. Sin embargo, en esta situación, el comprador del CD tiene el derecho de copiarlo y redistribuirlo.

En la práctica esto hace que las licencias de software libre se dividan en dos grandes grupos, aquellas que pueden ser mezcladas con código licenciado bajo GNU GPL (y que inevitablemente desaparecerán en el proceso, al ser el código resultante licenciado

bajo GNU GPL) y las que no lo permiten al incluir mayores u otros requisitos que no contemplan ni admiten la GNU GPL y que por lo tanto no pueden ser enlazadas ni mezcladas con código gobernado por la licencia GNU GPL.

En el sitio web oficial de GNU hay una lista de licencias que cumplen las condiciones impuestas por la GNU GPL y otras que no.

Aproximadamente el 60% del software licenciado como software libre emplea una licencia GPL.

Es decir, la licencia GNU GPL posibilita la modificación y redistribución del software, pero únicamente bajo esa misma licencia. Y añade que si se reutiliza en un mismo programa código "A" licenciado bajo licencia GNU GPL y código "B" licenciado bajo otro tipo de licencia libre, el código final "C", independientemente de la cantidad y calidad de cada uno de los códigos "A" y "B", debe estar bajo la licencia - GNU GPL

D'Sagan Carl

Licencias AGPL

La Licencia Pública General de Affero (en inglés Affero General PublicLicense, también Affero GPL o AGPL) es una licencia copyleft derivada de la Licencia Pública General de GNU diseñada específicamente para asegurar la cooperación con la comunidad en el caso de software que corra en servidores de red.

La Affero GPL es íntegramente una GNU GPL con una cláusula nueva que añade la obligación de distribuir el software si éste se ejecuta para ofrecer servicios a través de una red de ordenadores.

La Free Software Foundation recomienda que el uso de la GNU AGPLv3 sea considerado para cualquier software que usualmente corra sobre una red.

Licencias estilo BSD

Llamadas así porque se utilizan en gran cantidad de software distribuido junto a los sistemas operativos BSD. El autor, bajo tales licencias, mantiene la protección de BSD copyright únicamente para la renuncia de garantía y para requerir la adecuada atribución de la autoría en trabajos derivados, pero permite la libre redistribución y modificación, incluso si dichos trabajos tienen propietario.

Son muy permisivas, tanto que son fácilmente absorbidas al ser mezcladas con la licencia GNU GPL con quienes son compatibles. Puede argumentarse que esta licencia asegura “verdadero” software libre, en el sentido que el usuario tiene libertad ilimitada con respecto al software, y que puede decidir incluso redistribuirlo como no libre. Otras opiniones están orientadas a destacar que este tipo de licencia no contribuye al desarrollo de más software libre (normalmente utilizando la siguiente analogía: "una licencia BSD es más libre que una GPL si y sólo si se opina también que un país que permita la esclavitud es más libre que otro que no la permite").

Licencias estilo MPL y derivadas

Esta licencia es de Software Libre y tiene un gran valor porque fue el instrumento que empleó Netscape Communications Corp. para liberar su Netscape Communicator 4.0 y empezar ese proyecto tan importante para el mundo del Software Libre: Mozilla. Se

utilizan en gran cantidad de productos de software libre de uso cotidiano en todo tipo de sistemas operativos.

La licencia pública de Mozilla (en inglés Mozilla Public License o MPL) es una licencia de código abierto y de software libre. Fue desarrollada originalmente por Netscape Communications Corporation (una división de la empresa América Online), y más tarde su control fue traspasado a la Fundación Mozilla. (MPL- Nietzsche Friedrich)

La MPL es Software Libre y promueve eficazmente la colaboración evitando el efecto "viral" de la GPL. Desde un punto de vista del desarrollador la GPL presenta un inconveniente en este punto, y lamentablemente mucha gente se cierra en banda ante el uso de dicho código. No obstante la MPL no es tan excesivamente permisiva como las licencias tipo BSD. Estas licencias son denominadas de copyleft débil. La NPL fue la primera licencia nueva después de muchos años, que se encargaba de algunos puntos que no fueron tenidos en cuenta por las licencias BSD y GNU. En el espectro de las licencias de software libre se la puede considerar adyacente a la licencia estilo BSD, pero perfeccionada.

Comparación con el software de código abierto

Aunque en la práctica el software de ¹¹código abierto y el software libre comparten muchas de sus licencias, la FSF opina que el movimiento de código abierto es filosóficamente diferente del movimiento del software libre. Apareció en 1998 con un grupo de personas, entre los que cabe destacar a Eric S. Raymond y Bruce Perens, que formaron la Open Source Initiative (OSI). Ellos buscaban darle mayor relevancia a los beneficios prácticos del compartir el código fuente, e interesar a las principales casas de software y otras empresas de la industria de la alta tecnología en el concepto

Código abierto es el término con el que se conoce al software distribuido y desarrollado libremente. (Código Abierto- Machen Arthur)

Mientras que la FSF y Richard Stallman, prefieren plantear el asunto en términos éticos empleando el término Software Libre.

Estos defensores ven que el término "código abierto", en inglés open source, evita la ambigüedad del término en ese idioma que es free en free software. El término "código abierto" fue acuñado por Christine Peterson del thinktankForesightInstitute y se registró para actuar como marca registrada el término en inglés para los productos de software libre.

Mucha gente reconoce el beneficio cualitativo del proceso de desarrollo de software cuando los desarrolladores pueden usar, modificar y redistribuir el código fuente de un programa. El movimiento del software libre hace especial énfasis en los aspectos morales o éticos del software, viendo la excelencia técnica como un producto secundario deseable de su estándar ético. El movimiento de ¹²código abierto ve la excelencia técnica como el objetivo prioritario, siendo la compartición del código fuente un medio para dicho fin. Por dicho motivo, la FSF se distancia tanto del movimiento de código abierto como del término "Código Abierto" (en inglés Open Source).

Puesto que la OSI sólo aprueba las licencias que se ajustan a la OSD (Open SourceDefinition, Definición de Código Abierto), la mayoría de la gente lo interpreta como un esquema de distribución, e intercambia libremente "código abierto" con "software libre". Aun cuando existen importantes diferencias filosóficas entre ambos términos, especialmente en términos de las motivaciones para el desarrollo y el uso de tal software, raramente suelen tener impacto en el proceso de colaboración.

Aunque el término "código abierto" elimina la ambigüedad de Libertad frente a Precio, introduce una nueva: entre los programas que se ajustan a la definición de Código Abierto, que dan a los usuarios la libertad de mejorarlos, y los programas que simplemente tiene el código fuente disponible, posiblemente con fuertes restricciones sobre el uso de dicho código fuente. Mucha gente cree que cualquier software que tenga el código fuente disponible es de código abierto, puesto que lo pueden manipular (un ejemplo de este tipo de software sería el popular paquete de software gratuito Graphviz, inicialmente no libre pero que incluía el código fuente, aunque luego AT&T le cambió la licencia. Sin embargo, mucho de este software no da a sus usuarios la libertad de distribuir sus modificaciones, restringe el uso comercial, o en general restringe los derechos de los usuarios.

La Corporación AT&T (siglas de su antiguo nombre, American Telephone and Telegraph; NYSE: AT&T) es una compañía estadounidense de telecomunicaciones. Provee servicios de voz, video, datos, e internet a negocios, clientes y agencias del gobierno. Durante su larga historia, AT&T ha sido, en ocasiones, la compañía telefónica más grande del mundo, el operador de televisión por cable más grande de los Estados Unidos, y hasta fue clasificado como un monopolio. Durante el año 2005 se une con su competidora SBC Communications (Marcela Landres - Digital How)

Modelo de negocio

El negocio detrás del software libre se caracteriza por la oferta de servicios adicionales al software como: la personalización y/o instalación del mismo, soporte técnico,

donaciones, patrocinios; en contraposición al modelo de negocio basado en licencias predominante en el software de código cerrado.

Implicaciones económico-políticas

Una vez que un producto de software libre ha empezado a circular, rápidamente está disponible a un costo muy bajo. Al mismo tiempo, su utilidad no decrece. El software, en general, podría ser considerado un bien de uso inagotable, tomando en cuenta que su costo marginal es pequeñísimo y que no es un bien sujeto a rivalidad (la posesión del bien por un agente económico no impide que otro lo posea).

Puesto que el software libre permite el libre uso, modificación y redistribución, a menudo encuentra un hogar entre usuarios para los cuales el coste del software no libre es a veces prohibitivo, o como alternativa a la picantería. También es sencillo modificarlo localmente, lo que permite que sean posibles los esfuerzos de traducción a idiomas que no son necesariamente rentables comercialmente.

En informática un programa es de código cerrado cuando el código fuente que no se encuentra disponible para cualquier usuario, es decir no se hace público. Se le llama así en contraposición al código abierto. (Código Abierto - Machen Arthur)

La mayoría del software libre se produce por equipos internacionales que cooperan a través de la libre asociación. Existen muchas posturas acerca de la relación entre el software libre y el actual sistema político-económico:

- Algunos consideran el software libre como un competidor contra el centralismo en empresas y gobiernos, una forma de orden espontáneo o de anarquismo práctico.
- Algunos consideran el software libre como una forma de trabajo colaborativo en un modelo de mercado, tal como se había planteado el cooperativismo.
- Algunos comparan el software libre a una economía del regalo, donde el valor de una persona está basado en lo que ésta da a los demás, sin que incurra valor monetario formal de por medio.
- Grupos como Oekonux e Hipatia consideran que todo debería producirse de esta forma y que este modelo de producción no se limita a reemplazar el modelo no libre de desarrollo del software. La cooperación basada en la libre asociación puede usarse y se usa para otros propósitos (tales como escribir enciclopedias, por ejemplo).
- Hay proyectos de desarrollo con impulso gubernamental que utilizan software libre, así como en proyectos de voluntariado en países del tercer mundo.

Los equipos están típicamente compuestos por individuos con una amplia variedad de motivaciones, y pueden provenir tanto del sector privado, del sector voluntario o del sector público (Código Abierto- Machen Arthur)

Las implicaciones políticas y económicas del software libre, o su afinidad con el antiautoritarismo, son discutidas. Mientras para unos estas implicaciones son notorias y representan un factor importante a tomarse en cuenta, para otros si bien podría existir una leve relación, no tiene suficiente relevancia.

Seguridad relativa

Existe una cierta controversia sobre la seguridad del software libre frente al software no libre. Un método usado de forma habitual para determinar la seguridad relativa de los productos es determinar cuántos fallos de seguridad no parcheados existen en cada uno de los productos involucrados. Por lo general los usuarios de este método recomiendan que cuando un producto no proporcione un método de parchear los fallos de seguridad, no se use dicho producto, al menos hasta que no esté disponible un arreglo.

Software libre en la administración pública

Existe una serie de países en los cuales, sus administraciones públicas, han mostrado apoyo al software libre, sea migrando total o parcialmente sus servidores y sistemas de escritorio, sea subvencionándolo.

Como ejemplos de ello se tiene a Alemania, Argentina, Brasil, Argentina, Brasil, Cuba, Chile, China, Ecuador, España, Francia, México, República Dominicana y Venezuela.

Además de lo anterior, la ¹⁵Administración Pública tiene una cierta función de escaparate y/o guía de la industria que la hace tener un gran impacto, que debería dirigirse a la creación de un tejido tecnológico generador de riqueza nacional. Ésta puede crearse fomentando empresas, cuyo negocio sea en parte el desarrollo de nuevo software libre para la administración, el mantenimiento y la adaptación del existente asimismo auditar el software existente.

Motivaciones del software libre

- La motivación ética, abanderada por la Free Software Foundation, heredera de la cultura hacker, y partidaria del apelativo libre, que argumenta que el software es conocimiento y debe poderse difundir sin trabas. Su ocultación es una actitud antisocial y la posibilidad de modificar programas es una forma de libertad de expresión.

La definición de software libre no contempla el asunto del precio; un eslogan frecuentemente usado es "libre como en libertad, no como en cerveza gratis" o en inglés "Free as in freedom, not as in free beer" (aludiendo a la ambigüedad del término inglés "free"), y es habitual ver a la venta CD de software libre como distribuciones Linux. Sin embargo, en esta situación, el comprador del CD tiene el derecho de copiarlo y redistribuirlo. El software gratis puede incluir restricciones que no se adaptan a la definición de software libre, es decir puede no incluir el código fuente, puede prohibir explícitamente a los distribuidores recibir una compensación a cambio, etc. (Codigo Abierto- Machen Arthur)

- La motivación pragmática, abanderada por la Open Source Initiative y partidaria del apelativo abierto, que argumenta ventajas técnicas y económicas, con respecto a evitar una tragedia de los anticomunes mejorando los incentivos.

Aparte de estas dos grandes motivaciones, la gente que trabaja en software libre suele hacerlo por muchas otras razones, que van desde la diversión a la mera retribución económica, que es posible debido a modelos de negocio sustentables.

Ventajas del software libre

- **Bajo costo de adquisición:** Se trata de un software económico ya que permite un ahorro de grandes cantidades en la adquisición de las licencias.
- **Innovación tecnológica:** Esto se debe a que cada usuario puede aportar sus conocimientos y su experiencia y así decidir de manera conjunta hacia donde se debe dirigir la evolución y el desarrollo del software. Este es un gran avance en la tecnología mundial.
- **Independencia del proveedor:** Al disponer del código fuente, se garantiza una independencia del proveedor que hace que cada empresa o particular pueda seguir contribuyendo al desarrollo y los servicios del software.
- **Escrutinio público:** Esto hace que la corrección de errores y la mejora del producto se lleven a cabo de manera rápida y eficaz por cada uno de los usuarios que lleguen a utilizar el producto.
- **Adaptación del software:** Esta cualidad resulta de gran utilidad para empresas e industrias específicas que necesitan un software personalizado para realizar un trabajo específico y con el software libre se puede realizar y con costes mucho más razonables.
- **Lenguas:** Aunque el software se cree y salga al mercado en una sola lengua, el hecho de ser software libre facilita en gran medida su traducción y localización para que usuarios de diferentes partes del mundo puedan aprovechar estos beneficios.

2.2.2 Linux

Una distribución Linux o distribución GNU/Linux (con lo cualmente llamadas diestros) es una distribución de software basada en el núcleo Linux que incluye determinados paquetes de software para satisfacer las necesidades de un grupo específico de usuarios, dando así origen a ediciones domésticas, empresariales y para servidores. Por lo general están compuestas, total o mayoritariamente, de software libre, aunque a menudo incorporan aplicaciones o controladores propietarios.

Además del núcleo Linux, las distribuciones incluyen habitualmente las bibliotecas y herramientas del proyecto GNU y el sistema de ventanas X WindowSystem. Dependiendo del tipo de usuarios a los que la distribución esté dirigida se incluye también otro tipo de software como procesadores de texto, hoja de cálculo, reproductores multimedia, herramientas administrativas, etcétera. (Linux - JuanCarlos Espinoza)

Existen distribuciones que están soportadas comercialmente, como Fedora (Red Hat), openSUSE (Novell), Ubuntu (Canonical Ltd.), Mandriva, y distribuciones mantenidas por la comunidad como Debian y Gentoo. Aunque hay otras distribuciones que no están relacionadas con alguna empresa o comunidad, como es el caso de Slackware.

Antes de que surgieran las primeras distribuciones, un usuario de Linux debía ser algo experto en Unix; no sólo debía conocer qué bibliotecas y ejecutables necesitaba para iniciar el sistema y que funcionase, sino también los detalles importantes que se requieren en la instalación y configuración de los archivos en el sistema.

Entre las distribuciones más antiguas se incluían:

- Dos discos denominados H J Lu's "Boot-root" con el núcleo y un mínimo de herramientas para utilizar.
- MCC Interim Linux, que se podía descargar en un servidor público FTP de la Universidad de Mánchester en febrero de 1992.
- TAMU, creado por entusiastas de la Universidad de Texas A&M al mismo tiempo que SLS
- SLS (Softlanding Linux System).
- Yggdrasil Linux creó el primer CD-ROM de una distribución GNU/Linux.

Las distribuciones GNU/Linux comenzaron a surgir poco después de que el núcleo Linux fuera utilizado por otros programadores además de los creadores originales. (Linux - JuanCarlos Espinoza)

Existía mayor interés en desarrollar un sistema operativo que en desarrollar aplicaciones, interfaces para los usuarios o un paquete de software conveniente.

- SLS no estuvo bien mantenida; así pues, Patrick Volkerding lanzó una distribución basada en SLS a la que llamó Slackware; lanzada el 16 de julio de 1993. Esta es la distribución más antigua que está en desarrollo activo.
- Los usuarios vieron en Linux una alternativa a los sistemas operativos DOS, Microsoft Windows en la plataforma PC, Mac OS en Apple Macintosh y las versiones de uso bajo licencia (de pago) de UNIX. La mayoría de estos primeros usuarios se habían familiarizado con el entorno UNIX en sus trabajos o centros de estudios. Estos adoptaron GNU/Linux¹⁹ por su estabilidad, reducido (o nulo) coste y por la disponibilidad del código fuente del software incluido.

Componentes

El escritorio típico de una distribución Linux contiene un núcleo, herramientas y librerías, software adicional, documentación, un sistema de ventanas, un administrador de ventanas y un entorno de escritorio, este suele ser GNOME o KDE. Gran parte del software incluido es de fuente abierta o software libre y distribuido por sus desarrolladores tanto en binario compilado como en forma de código fuente, permitiendo a sus usuarios modificar o compilar el código fuente original si lo desean. Muchas distribuciones incorporan software privativo, no disponible en forma de código fuente.

Las distribuciones eran originalmente una cuestión de comodidad para el usuario medio, evitándole la instalación (y en muchos casos compilación) por separado de paquetes de uso común, pero hoy se han popularizado incluso entre los expertos en éste tipo de sistemas operativos (UNIX / Linux). Si bien, históricamente, Linux estuvo mejor posicionado en el mercado de los servidores, distribuciones centradas en la facilidad de instalación y uso, tales como Fedora, Mandriva, Opensuse, Knoppix y Ubuntu, entre otras, han logrado una mayor aceptación en el mercado doméstico. (Linux - Juan Carlos Espinoza)

Muchas distribuciones proveen un sistema de instalación gráfica como lo hacen otros sistemas modernos. Distribuciones independientes como Gentoo Linux, T2 y Linux FromScratch proveen el código fuente de todo el software y solo incluyen los binarios del núcleo, herramientas de compilación y de un instalador; el instalador compila todo el software para el CPU específico de la PC del usuario.

Gestión de paquetes

Las distribuciones están divididas en “paquetes”. Cada paquete contiene una aplicación específica o un servicio. Ejemplos de paquetes son una librería para manejar el formato de imagen PNG, una colección de tipografías o un navegador web.

El paquete es generalmente distribuido en su versión compilada y la instalación y desinstalación de los paquetes es controlada por un sistema de gestión de paquetes en lugar de un simple gestor de archivos. Cada paquete elaborado para ese sistema de ¹⁹paquetes contiene meta-información tal como fecha de creación, descripción del paquete y sus dependencias.

El sistema de paquetes analiza esta información para permitir la búsqueda de paquetes, actualizar las librerías y aplicaciones instaladas, revisar que todas las dependencias se cumplan y obtenerlas si no se cuenta con ellas de manera automática. (Linux – Juan Carlos Espinoza)

Perezoso Algunos de los sistemas de paquetes más usados son:

- RPM, creado por Red Hat y usado por un gran número de distribuciones de Linux, es el formato de paquetes del Linux Standard Base.

- Deb, paquetes Debian, originalmente introducidos por Debian, pero también utilizados por otros como Knoppix y Ubuntu.
- Tgz, usado por Slackware, empaqueta el software usando tar y gzip. Pero, además, hay algunas herramientas de más alto nivel para tratar con este formato: slapt-get, slackpkg y swaret.
- Ebuilds, archivo que contiene información acerca de cómo obtener, compilar e instalar un paquete en el sistema Portage de Gentoo Linux con el comando emerge. Generalmente, estas instalaciones se basan en la compilación de fuentes, aunque algunos paquetes binarios se pueden instalar de esta manera.
- Pacman, para Arch Linux usa binarios precompilados distribuidos en un fichero .pkg.tar.gz ó .pkg.tar.xz.

Aunque las distribuciones casi siempre vienen con mucha mayor cantidad de software que los sistemas propietarios, en ocasiones algunos usuarios pueden instalar software que no fue incluido en la distribución. Si el software es distribuido sólo en forma de código fuente, requerirá ser compilado por el ordenador. Sin embargo, si el programa es compilado, el paquete no será registrado por el gestor de paquetes y por lo tanto no podrá ser controlado por él. Esto significa que el administrador del equipo tendrá que tomar medidas adicionales para mantener el software actualizado. El gestor de paquetes no lo podrá hacer automáticamente.

La mayor parte de las distribuciones instalan los paquetes, incluyendo el núcleo Linux y otras piezas fundamentales del sistema operativo con una configuración preestablecida. Esto hace la instalación más sencilla, especialmente para los usuarios nuevos, pero no es siempre aceptable, pues hay programas que deben de ser cuidadosamente configurados

para que sean funcionales, para que operen correctamente con otra aplicación o para que su seguridad sea robusta. Los administradores se ven obligados a invertir tiempo reconfigurando y revisando software soportado por la distribución.

Es también posible armar un sistema a la medida en su totalidad, descartando incluso el uso de una distribución. Lo primero que hay que hacer es generar un sistema base que permita conseguir, compilar, configurar e instalar el código fuente. Generar los binarios de este sistema base requerirá de otra máquina que sea capaz de generar los binarios para el dispositivo deseado, esto puede ser alcanzado por medio de una compilación cruzada.

En otras distribuciones la instalación puede llegar a ser muy lenta, pues es posible ajustar y configurar la mayor parte o la totalidad del software incluido en la distribución. (Linux – Juan Carlos Espinoza)

Tipos y tendencias

En general, las distribuciones Linux pueden ser:

- Comerciales o no comerciales.
- Ser completamente libres o incluir software privativo.
- Diseñadas para uso en el hogar o en las empresas.
- Diseñadas para servidores, escritorios o dispositivos empotrados.
- Orientadas a usuarios regulares o usuarios avanzados.

- De uso general o para dispositivos altamente especializados, como un cortafuegos, un enrutador o un clúster computacional.
- Diseñadas e incluso certificadas para un hardware o arquitectura específicos.
- Orientadas hacia grupos en específico
- Configuradas especialmente para ser más seguras, completas, portables o fáciles de usar.
- Soportadas bajo distintos tipos de hardware.

La diversidad de las distribuciones Linux es debido a cuestiones técnicas, de organización y de puntos de vista diferentes entre usuarios y proveedores. El modo de licenciamiento del software libre permite que cualquier usuario con los conocimientos e interés suficiente pueda adaptar o diseñar una distribución de acuerdo a sus necesidades.
(Linux - Juan Carlos Espinoza)

Distribuciones que no requieren instalación (Live CD)

Una distribución live o Live CD o Live DVD, más genéricamente Live Distro, (traducido en ocasiones como CD vivo o CD autónomo), es una distribución almacenada en un medio extraíble, tradicionalmente un CD o un DVD (de ahí sus nombres), que puede ejecutarse desde éste sin necesidad de instalarlo en el disco duro de una computadora, para lo cual usa la memoria RAM como disco duro virtual y el propio medio como sistema de archivos.

Cuando el sistema operativo es ejecutado por un dispositivo de sólo lectura como un CD o DVD, el usuario necesita utilizar una memoria USB o un disco duro instalado en

la máquina para conservar su información entre sesiones. La información del sistema operativo es usualmente cargada en la memoria RAM.

La portabilidad de este tipo de distribuciones las hace ideales para ser utilizadas en demostraciones, operaciones de recuperación, cuando se utiliza una máquina ajena o como medio de instalación para una distribución estándar. Actualmente, casi todas las distribuciones tienen una versión CD/DVD autónomo o "vivo".

Desarrollo

La mayoría de las distribuciones están, en mayor o menor medida, desarrolladas y dirigidas por sus comunidades de desarrolladores y usuarios. En algunos casos están dirigidas y financiadas completamente por la comunidad. Como ocurre con Debian GNU/Linux, mientras que otras mantienen una distribución comercial y una versión de la comunidad, como hace RedHat con Fedora, o SuSE con OpenSuSE.

En muchas ciudades y regiones, asociaciones locales conocidas como grupos de usuarios promueven este sistema operativo y el software libre. Suelen ofrecer conferencias, talleres o soporte técnico de forma gratuita o introducción a la instalación de GNU/Linux para nuevos usuarios.

En las distribuciones y otros proyectos de software libre y código abierto son muy comunes las salas de chat IRC y grupos de noticias. Los foros también son comunes, sobre todo en el soporte a usuarios, y las listas de correo suelen ser el medio principal para discutir sobre el desarrollo, aunque también se utilizan como medio de soporte al usuario.

Escala de desarrollo

Un estudio sobre la distribución Red Hat 7.1 reveló que ésta en particular posee más de 30 millones de líneas de código real. Utilizando el modelo de cálculo de costos COCOMO, puede estimarse que esta distribución requeriría 8.000 programadores por año para su desarrollo. De haber sido desarrollado por medios convencionales de código cerrado, hubiera costado más de mil millones de dólares en los Estados Unidos.

La mayor parte de su código (71%) pertenecía al lenguaje C, pero fueron utilizados muchos otros lenguajes para su desarrollo, incluyendo C++, Bash, Lisp, Ensamblador, Perl, Fortran y Python.

Además, la licencia predominante en alrededor de la mitad de su código total (contado en líneas de código) fue la GPL en su versión 2.

El núcleo Linux contenía entonces 2,4 millones de líneas de código, lo que representaba el 8% del total.

En un estudio posterior se realizó el mismo análisis para Debian GNU/Linux versión 2.2. Esta distribución contenía más de 55 millones de líneas de código fuente, y habría

costado 1.900 millones de dólares (año 2000) el desarrollo por medios convencionales (no libres); el núcleo Linux en octubre de 2003 tiene unas 5,5 millones de líneas más.

Distribuciones populares

Entre las distribuciones Linux más populares se incluyen:

- Arch Linux, una distribución basada en el principio KISS con un sistema de desarrollo continuo entre cada versión (no es necesario volver a instalar todo el sistema para actualizarlo).
- CentOS, una distribución creada a partir del mismo código del sistema Red Hat pero mantenida por una comunidad de desarrolladores voluntarios.
- Debian, una distribución mantenida por una red de desarrolladores voluntarios con un gran compromiso por los principios del software libre.
- Fedora, una distribución lanzada por Red Hat para la comunidad.
- Gentoo, una distribución orientada a usuarios avanzados, conocida por la similitud en su sistema de paquetes con el FreeBSDPorts, un sistema que automatiza la compilación de aplicaciones desde su código fuente.
- gOS, una distribución basada en Ubuntu para netbooks.
- Knoppix, la primera distribución live en correr completamente desde un medio extraíble. Está basada en Debian.
- Kubuntu, la versión en KDE de Ubuntu.
- Linux Mint, una popular distribución derivada de Ubuntu.
- Mandriva, mantenida por la compañía francesa del mismo nombre, es un sistema popular en Francia y Brasil. Está basada en Red Hat.

- openSUSE, originalmente basada en Slackware es patrocinada actualmente por la compañía Novell.
- PCLinuxOS, derivada de Mandriva, paso de ser un pequeño proyecto a una popular distribución con una gran comunidad de desarrolladores.
- PuppyLinux, versión para pc's antiguas o con pocos recursos que pesa 130 mb.
- Red Hat Enterprise Linux, derivada de Fedora, es mantenida y soportada comercialmente por Red Hat.
- Slackware, una de las primeras distribuciones Linux y la más antigua en funcionamiento. Fue fundada en 1993 y desde entonces ha sido mantenida activamente por Patrick J. Volkerding.
- Slax, es un sistema Linux pequeño, moderno, rápido y portable orientado a la modularidad. Está basado en Slackware.
- Ubuntu, una popular distribución para escritorio basada en Debian y mantenida por Canonical.
- Dragora y Trisquel GNU/Linux, que van adquiriendo importancia entre las distribuciones que sólo contienen software libre.

El sitio web DistroWatch ofrece una lista de las distribuciones más populares; la lista está basada principalmente en el número de visitas, por lo que no ofrece resultados muy confiables acerca de la popularidad de las distribuciones.

Distribuciones especializadas

Otras distribuciones se especializan en grupos específicos:

- OpenWrt, diseñada para ser empotrada en dispositivos enrutadores.

- Edubuntu, un sistema del proyecto Ubuntu diseñado para entornos educativos.
- MythTV, orientada para equipos multimedia o grabadores de vídeo digital.
- Musix, una distribución de Argentina destinada a los músicos.
- mkLinux, YellowDog Linux o Black Lab Linux, orientadas a usuarios de Macintosh y de la plataforma PowerPC.
- 64 Studio, una distribución basada en Debian diseñada para la edición multimedia.
- ABC GNU/Linux, distribución para la construcción de clusters Beowulf desarrollado por Iker Castaños Chavarri, Universidad del País Vasco (España) .

2.2.3 Php

- Introducción, definición y evolución de PHP Actualmente la red de Internet, compuesta por páginas web, sirve de soporte a una gran cantidad de sistemas de información y comunicaciones que engloban áreas tan importantes como la investigación, el comercio electrónico, la visualización de información, el correo electrónico..
- Las páginas web son textos ASCII escritos en HTML (HypertextMarkupLanguage), que se transfieren entre los servidores de WWW y los navegadores mediante el protocolo HTTP (Hypertext Transfer Protocol).

- Cuando comenzó la World Wide Web, los sitios web ofrecían páginas estáticas, es decir, que a efectos del usuario, el único proceso realizado era el de visualización de sus contenidos por parte del navegador del cliente.
- A medida que pasaron los años, aumentó la afluencia de público y surge la necesidad de reunir y procesar las peticiones del cliente con la finalidad de ofrecerle informaciones mejor dirigidas, escogidas y elaboradas.
- Es entonces, cuando surgen dos alternativas de diseño web:
- Las llamadas páginas en el lado del cliente, programadas en HTML dinámico (DHTML), se ejecutan en el navegador y son capaces de cambiar su apariencia. Las páginas del lado del servidor, permiten modificar los contenidos antes de enviarlos al cliente. Las primeras fueron CGI (Common Gateway Interface) que implementan páginas web activas en el servidor, capaces de leer lo que le enviaba un cliente al servidor a través de un formulario, o escribir en la página del cliente.
- PHP (HypertextPreprocessor) es un lenguaje script (no se compila para conseguir códigos máquina si no que existe un intérprete que lee el código y se encarga de ejecutar las instrucciones que contiene éste código), para el desarrollo de páginas web dinámicas del lado del servidor, cuyos fragmentos de código se intercalan fácilmente en páginas HTML, debido a esto, y a que es de Open Source (código abierto), es el más popular y extendido en la web.
- PHP es capaz de realizar determinadas acciones de una forma fácil y eficaz sin tener que generar programas programados en un lenguaje distinto al HTML. Esto se debe a que PHP ofrece un extenso conjunto de funciones para la explotación de bases de datos sin complicaciones. Es por esto, que levanta un mayor interés con respecto a los lenguajes pensados para los CGI.

- PHP fue desarrollado originalmente por RasmusLedford en 1994 como un CGI escrito en Perl que permitía la interpretación de un número limitado de comandos. El sistema fue denominado Personal Home Page Tools y consiguió relativo éxito gracias a que otras personas pidieron a Rasmus que les permitiese utilizar sus programas en sus propias páginas.

Utilizando DHTML podemos conseguir que los objetos aparezcan y desaparezcan, que varíen su tamaño, que se les asocie posiciones absolutas en pantalla. En definitiva, DHTML aumenta las posibilidades de visualización de las instrucciones Html. (Php- Egon Schmid & Programación Desarrollo Web Con Php Y Mysql - Welling, Luke Y Thomson, Laura)

- Cuando Rasmus tuvo la necesidad de crear páginas dinámicas que trabajasen con formularios, creó una serie de etiquetas a las que denominó “FormInterpreters”, y lo sacó al público con el nombre de PHP/FI en 1995. Luego salió la versión mejorada, llamada PHP/FI 2.0.
- ZeevSuraski y AndiGutmans programaron el analizador sintáctico incluyendo nuevas funcionalidades como el soporte a nuevos protocolos de Internet y el soporte a la gran mayoría de las bases de datos comerciales, como MySQL y Postgre SQL, así como un módulo para Apache. Con estas mejoras surgió PHP3 en 1997. Este analizador define la sintaxis y semántica de la versión PHP3 y la siguiente: PHP4.
- PHP3 carecía del uso de sesiones, algo muy común en las páginas web de cierta complejidad. En el año 2000, PHP3 evolucionó a PHP4, que utiliza el motor Zend (desarrollado por Zeev y Andi encargado de interpretar el código fuente de los scripts de PHP), desarrollado para cubrir las necesidades actuales y

solucionar algunos inconvenientes de la anterior versión. Algunas mejoras de esta nueva versión son su mayor independencia del servidor web y su rapidez, ya que primero se compila y luego se ejecuta, mientras que antes se ejecutaba a la vez que se interpretaba el código.

- La última versión es PHP5, que utiliza el motor Zend-2 y presenta mejoras significativas y un entorno de programación orientado a objetos mucho más completo, que permite que el PHP proporcione un alto rendimiento a las aplicaciones Web empresariales a nivel de las plataformas J2EE y .NET. Otro lenguaje de scripting para la generación dinámica de contenidos en el servidor es ASP. Aunque se parece a PHP en cuanto a potencia y dificultad, su sintaxis llega a diferir notablemente.
- Una diferencia sensible es que PHP ha sido desarrollado inicialmente para entornos UNIX y es en este sistema operativo donde se aprovechan mejor sus prestaciones y consigue un mayor rendimiento. ASP, que es una tecnología Microsoft, está orientado a sistemas Windows, especialmente NT.

2.2.4 Mysql

MySQL es un sistema de gestión de bases de datos relacional, licenciado bajo la GPL de la GNU. Su diseño multihilo le permite soportar una gran carga de forma muy eficiente. MySQL fue creada por la empresa sueca MySQL AB, que mantiene el copyright del código fuente del servidor SQL, así como también de la marca.

Aunque MySQL es software libre, MySQL AB distribuye una versión comercial de MySQL, que no se diferencia de la versión libre más que en el soporte técnico que se ofrece, y la posibilidad de integrar este gestor en un software propietario, ya que de no ser así, se vulneraría la licencia GPL.

Este gestor de bases de datos es, probablemente, el gestor más usado en el mundo del software libre, debido a su gran rapidez y facilidad de uso. Esta gran aceptación es debida, en parte, a que existen infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación, además de su fácil instalación y configuración.

Historia de MySQL

MySQL surgió como un intento de conectar el gestor mSQL a las tablas propias de MySQL AB, usando sus propias rutinas a bajo nivel. Tras unas primeras pruebas, vieron que mSQL no era lo bastante flexible para lo que necesitaban, por lo que tuvieron que desarrollar nuevas funciones. Esto resultó en una interfaz SQL a su base de datos, con una interfaz totalmente compatible a mSQL.

Por sus librerías han llevado el prefijo 'my' durante los diez últimos años. Por otro lado, la hija de uno de los desarrolladores se llama My. No saben cuál de estas dos causas (aunque bien podrían tratarse de la misma), han dado lugar al nombre de este conocido gestor de bases de datos.

Características de MySQL

Las principales características de este gestor de bases de datos son las siguientes:

1. Aprovecha la potencia de sistemas multiprocesador, gracias a su implementación multihilo.

2. Soporta gran cantidad de tipos de datos para las columnas.
3. Dispone de API's en gran cantidad de lenguajes (C, C++, Java, PHP, etc).
4. Gran portabilidad entre sistemas.
5. Soporta hasta 32 índices por tabla.
6. Gestión de usuarios y passwords, manteniendo un muy buen nivel de seguridad en los datos.

¿Qué es lo que le falta?

MySQL se incluye en el grupo de sistemas de bases de datos relacionales, carece de algunas de sus principales características:

1. Subconsultas: tal vez ésta sea una de las características que más se echan en falta, aunque gran parte de las veces que se necesitan, es posible reescribirlas de manera que no sean necesarias.
2. SELECT INTO TABLE: Esta característica propia de Oracle, todavía no está implementada.
3. Triggers y Procedures: Se tiene pensado incluir el uso de procedures almacenados en la base de datos, pero no el de triggers, ya que los triggers reducen de forma significativa el rendimiento de la base de datos, incluso en aquellas consultas que no los activan.
4. Transacciones: a partir de las últimas versiones ya hay soporte para transacciones, se ha de activar un modo especial.
5. Integridad referencial: aunque sí que admite la declaración de claves ajenas en la creación tablas, internamente no las trata de forma diferente al resto de campos.

Los desarrolladores comentan en la documentación que todas estas carencias no les resultaba un problema, ya que era lo que ellos necesitaban. De hecho, MySQL fue diseñada con estas características, debido a que lo que buscaban era un gestor de bases de datos con una gran rapidez de respuesta. Pero ha sido con la distribución de MySQL por Internet, cuando más y más gente les está pidiendo estas funcionalidades, por lo que serán incluidas en futuras versiones del gestor.

Opinión Personal

Tras haber probado la PostgreSQL, y viendo las carencias que poseía MySQL, pensé que no merecería la pena ni tan siquiera probarlo, aunque por otro lado, creía que algo debía tener para que hubiera tanta gente que lo use, cuando está a merced de cada uno elegir la base de datos que quiere usar. La verdad es tras haber hecho unas pocas pruebas, mi impresión sobre este gestor mejoró considerablemente.

MySQL surgió como una necesidad de un grupo de personas sobre un gestor de bases de datos rápido, por lo que sus desarrolladores fueron implementando únicamente lo que precisaban, intentando hacerlo funcionar de forma óptima. (Base de Datos - Jhon Roso, Norelis Rivas, Mercedes Ramírez)

Para comenzar, el shell de comandos muestra una interfaz más amena y los comandos para gestionar la base de datos son más intuitivos, siendo muchos de ellos sentencias SQL (hay que decir que no dispone de ayuda en línea sobre las palabras clave de SQL). Por otro lado, la API de PHP para acceder a MySQL era muchísimo más sencilla de usar, teniendo un estilo mucho más natural.

2.2.5 Apache

El servidor HTTP (Hypertext Transfer Protocol) Apache, es un software libre y de código abierto para las plataformas Windows, Mac OS X y UNIX (GNU, BSD, etc), en la cual se hace realidad y se implementa el protocolo HTTP 1.1 y la noción de sitios virtuales.

La verdad es que aunque estas diferencias son agradables, no llegan a tener una importancia suficiente como para cambiar el gestor que habitualmente solemos usar. Este tipo de cambios deberían estar basados en diferencias en el rendimiento que se nos ofrece, que es lo que se tratará en el siguiente apartado.

Apache, el servidor web escogido por excelencia, calidad de servicios, robustez y estabilidad hacen que día a día usuarios y servidores reiteren su confianza y renueven la elección a este servicio. Programacion Desarrollo Web Con Php Y Mysql - Welling, Luke Y Thomson, Laura

Historia

Para comenzar con la historia de Apache, nos remontamos a febrero del 1995, y fue en esta cronología en donde el grupo Apache comienza su desarrollo, el cual inicialmente se basó en el código del popular NCSA HTTPd 1.3. Fueron Brian Behlendorf y Cliff Skolnick quienes a través de una lista de correo coordinaron el trabajo, establecieron y consolidaron el grupo de desarrollo de Apache, y es lo que hoy conocemos como la Apache Software Foundation.

Hoy en día podemos mencionar a Apache como uno de los mayores éxitos alcanzados por un grupo comunitario de desarrollo, adoptando una filosofía libre de conocimiento y distribución, y porque no decir que muchas veces de la mano de Apache y su poderoso servidor web grandes, medianas y pequeñas empresas optaron por escoger servidores GNU/Linux en sus oficinas, es por ello que este logro también es muy reconocido, ya que va de la mano junto a su filosofía de comunidad y colaboración.

En cuanto a las características que posee Apache y que la llevo al éxito en la inserción y utilización en ámbitos empresariales, tecnológicos y educativos:

Cabe aclarar que el desarrollo del Servidor Web Apache inicialmente fue para las plataformas UNIX, pero como gran desafío se comenzó a desarrollar un proyecto de migración a las diferentes plataformas que hoy conocemos, gracias al trabajo

inmensurable de toda una comunidad de desarrolladores que aportaron sus parches, reportando errores y haciendo el esfuerzo de corregirlos. (Unix - Sergio Luján Mora)

1. Fundamentalmente corre sobre una multitud de plataformas y Sistemas Operativos.
2. Apache ofrece tecnología libre y de código abierto, otorgándole una transparencia y dando la posibilidad de conocer que es lo que realmente estamos instalando.
3. Apache es un servidor Web altamente configurable y de diseño modular, capaz de ampliar su funcionalidad y calidad de servicios.
4. Apache trabaja en conjunto con gran cantidad de Lenguajes de Programación interpretados como PHP (PHP Hypertext Pre-processor), Perl, soporte con CGI (Common Gateway Interface), Java, JSP (Java Server Pages) y otros lenguajes de script, el complemento ideal para los sitios web dinámicos que vemos en la actualidad.
5. Es posible configurar y personalizar cada uno de los mensajes de error que se pueden producir por la utilización del servidor.
6. Contar con los archivos Log, en donde registra gran cantidad de información global del sistema, errores producidos en un determinado tiempo, en la cual estos archivos son de gran importancia para los administradores de sistemas y pueden influenciar de alguna manera las políticas de seguridad debido a la gran cantidad de información que contiene.
7. Otra particularidad propia de Apache y que está muy ligada a su pensamiento y filosofía libre, es que al ser tan popular y utilizado, es posible encontrar gran cantidad de documentos.

Tal y como dijimos anteriormente Apache fue desarrollado altamente modular por tal motivos contamos con una gran cantidad de módulos escritos y que aportan al complemento y la expansión funcional de este Servidor. Entre los módulos más conocidos tenemos:

- **mod_ssl** – Brindando comunicaciones seguras vía SSL (Secure Sockets Layer) y TLS (TransportLayer Security).
- **mod_rewrite** – Conocido también como reescritura de direcciones o URL, sirven para reescribir URL dinámicas y transformarlas en estáticas.
- **mod_dav** – Con soporte para el protocolo WebDav.
- **mod_auth_ldap** – Permitiendo autenticar usuarios con la ayuda de LDAP (LightweightDirectory Access Protocol)
- **mod_perl** – Soporte para el lenguaje de programación Perl.
- **mod_php** – Soporte para el lenguaje de programación PHP.
- **mod_python** – Soporte para los sitios dinámicos realizados en Python.
- **mod_ruby** – Soporte para el lenguaje de programación Ruby.
- **mod_mono** – Soporte para el proyecto MONO, la implementación .NET en GNU/Linux.

Como todo gran proyecto de software y en especial como es el de Apache, cuenta con un sitio Web Oficial del proyecto, en donde podemos encontrar por sobre todas las cosas una gran cantidad de información y documentación traducida en varios idiomas. Entre tantas cosas podemos ver, como instalar, configurar y dejar totalmente funcional nuestro servidor web, conocer más acerca de los módulos, etc.

Los requerimientos mínimos del sistema que debemos respetar por lo menos para esta última versión estable 2.2.2, con respecto al espacio en Disco necesitamos tener

disponible al menos 50 MB, si bien Apache ocupa aproximadamente de 10 MB necesitamos reservar y ser precavidos para futuras expansiones de funcionalidades. Si hablamos de Apache en la plataforma GNU/Linux necesitamos tener pre instalado el compilador ANSI-C conocido como GCC (Compilador GNU C) entre las tantas opciones que podemos tener para poder compilar el código fuente. Otra manera de instalar Apache, es dependiendo de la distribución GNU/Linux y los diferentes archivos pre compilados que existan, de la misma manera es para la plataforma Windows. Con esto estas características podremos tener instalado en pocos minutos nuestro servidor Web Apache.

Como bien se sabe, en la parte de la configuración siempre se lleva a cabo a través de archivos de textos planos totalmente documentados y comentados, y obviamente al momento de la instalación la configuración que reside en la misma es considerada y llamada por defecto, con algunos parámetros deshabilitados y la suficiente configuración para que el servidor ya esté en condiciones de ser activado.

Al ser un servicio de sistema realmente muy importante, solo los usuarios autorizados son los únicos capaces de controlar el funcionamiento del mismo y la calidad de sus servicios, las visitas a los archivos de configuración y de seguridad. Estos son los usuarios que estarán autorizados para comenzar, reiniciar o apagar el servicio Web.

2.2.6 Software educativo

Sánchez J. (1999), en su Libro "Construyendo y Aprendiendo con el Computador", define el concepto genérico de Software Educativo como cualquier programa computacional cuyas características estructurales y funcionales sirvan de apoyo al proceso de enseñar, aprender y administrar. Un concepto más restringido de Software Educativo lo define como aquel material de aprendizaje especialmente diseñado para ser utilizado con una computadora en los procesos de enseñar y aprender.

Según Rguez Lamas (2000), es una aplicación informática, que soportada sobre una bien definida estrategia pedagógica, apoya directamente el proceso de enseñanza aprendizaje constituyendo un efectivo instrumento para el desarrollo educacional del hombre del próximo siglo.

El Software Educativo se puede considerar como el conjunto de recursos informáticos diseñados con la intención de ser utilizados en el contexto del proceso de enseñanza – aprendizaje.

Se caracterizan por ser altamente interactivos, a partir del empleo de recursos multimedia, como videos, sonidos, fotografías, diccionarios especializados, explicaciones de experimentados profesores, ejercicios y juegos instructivos que apoyan las funciones de evaluación y diagnóstico.

Los software educativos pueden tratar las diferentes materias (Matemática, Idiomas, Geografía, Dibujo), de formas muy diversas (a partir de cuestionarios, facilitando una información estructurada a los alumnos, mediante la simulación de fenómenos) y

ofrecer un entorno de trabajo más o menos sensible a las circunstancias de los alumnos y más o menos rico en posibilidades de interacción; pero todos comparten las siguientes características:

- Permite la interactividad con los estudiantes, retroalimentándolos y evaluando lo aprendido.
- Facilita las representaciones animadas.
- Incide en el desarrollo de las habilidades a través de la ejercitación.
- Permite simular procesos complejos.
- Reduce el tiempo de que se dispone para impartir gran cantidad de conocimientos facilitando un trabajo diferenciado, introduciendo al estudiante en el trabajo con los medios computarizados.
- Facilita el trabajo independiente y a la vez un tratamiento individual de las diferencias.
- Permite al usuario (estudiante) introducirse en las técnicas más avanzadas.

El uso del software educativo en el proceso de enseñanza - aprendizaje puede ser:

Por parte del alumno.

Se evidencia cuando el estudiante opera directamente el software educativo, pero en este caso es de vital importancia la acción dirigida por el profesor.

Por parte del profesor.

Se manifiesta cuando el profesor opera directamente con el software y el estudiante actúa como receptor del sistema de información. La generalidad plantea que este no es el caso más productivo para el aprendizaje.

El uso del software por parte del docente proporciona numerosas ventajas, entre ellas:

- Enriquece el campo de la Pedagogía al incorporar la tecnología de punta que revoluciona los métodos de enseñanza - aprendizaje.
- Constituyen una nueva, atractiva, dinámica y rica fuente de conocimientos.
- Pueden adaptar el software a las características y necesidades de su grupo teniendo en cuenta el diagnóstico en el proceso de enseñanza - aprendizaje.
- Permiten elevar la calidad del proceso docente - educativo.
- Permiten controlar las tareas docentes de forma individual o colectiva.
- Muestran la interdisciplinariedad de las asignaturas.
- Marca las posibilidades para una nueva clase más desarrolladora.

Los software educativos a pesar de tener unos rasgos esenciales básicos y una estructura general común se presentan con unas características muy diversas: unos aparentan ser un laboratorio o una biblioteca, otros se limitan a ofrecer una función instrumental del tipo máquina de escribir o calculadora, otros se presentan como un juego o como un libro, bastantes tienen vocación de examen, unos pocos se creen expertos y la mayoría participan en mayor o menor medida de algunas de estas peculiaridades.

Para poner orden a esta disparidad, se elaboraron múltiples tipologías que los clasifican a partir de diferentes criterios.

Hasta el año 2003, según los polos en los cuales se ha movido la educación, existían dos tipos de software educativos:

1.- Algorítmicos, donde predomina el aprendizaje vía transmisión del conocimiento, pues el rol del alumno es asimilar el máximo de lo que se le transmite.

Considerando la función educativa se pueden clasificar en:

Sistemas Tutoriales

Sistema basado en el diálogo con el estudiante, adecuado para presentar información objetiva, tiene en cuenta las características del alumno, siguiendo una estrategia pedagógica para la transmisión de conocimientos.

Sistemas Entrenadores

Se parte de que los estudiantes cuentan con los conceptos y destrezas que van a practicar, por lo que su propósito es contribuir al desarrollo de una determinada habilidad, intelectual, manual o motora, profundizando en las dos fases finales del aprendizaje: aplicación y retroalimentación.

Libros Electrónicos

Su objetivo es presentar información al estudiante a partir del uso de texto, gráficos, animaciones, videos, etc., pero con un nivel de interactividad y motivación que le facilite las acciones que realiza.

2.- Heurísticos, donde el estudiante descubre el conocimiento interactuando con el ambiente de aprendizaje que le permita llegar a él.

Considerando la función educativa se pueden clasificar en:

Simuladores

Su objetivo es apoyar el proceso de enseñanza – aprendizaje, semejando la realidad de forma entretenida.

Juegos Educativos

Su objetivo es llegar a situaciones excitantes y entretenidas, sin dejar en ocasiones de simular la realidad.

Sistemas Expertos

Programa de conocimientos intensivo que resuelve problemas que normalmente requieren de la pericia humana. Ejecuta muchas funciones secundarias de manera

análoga a un experto, por ejemplo, preguntar aspectos importantes y explicar razonamientos.

Sistemas Tutoriales Inteligentes de enseñanza

Despiertan mayor interés y motivación, puesto que pueden detectar errores, clasificarlos, y explicar por qué se producen, favoreciendo así el proceso de retroalimentación del estudiante.

A partir del 2004 surge una nueva tendencia, que es la de integrar en un mismo producto, todas o algunas de estas tipologías de software educativos. A este nuevo modelo de software se le ha denominado HIPERENTORNO EDUCATIVO o HIPERENTORNO DE APRENDIZAJE, lo cual no es más que un sistema informático basado en tecnología hipertexto que contiene una mezcla de elementos representativos de diversas tipologías de software educativo.

2.2.7 Hot-potatoes

Hot Potatoes es un conjunto de seis herramientas que pueden ayudar a educadores, profesores, y escritores a publicar material educativo en la Web.

Las pequeñas herramientas de Hot Potatoes te permiten crear multi-elección interactiva, respuesta corta, rellenar en el espacio, crucigramas, y ejercicios de desordenación de una frase usando HTML y JavaScript sin necesidad de tener ningún conocimiento ni de HTML ni de JavaScript.

JBC te permite crear exámenes "tipo test" donde cada pregunta puede tener hasta cinco respuestas, y donde cada número de ellas pueden ser correctas o no. Cada estudiante recibe un porcentaje de acierto después de cada pregunta acertada.

JQuiz te facilita el diseño de preguntas con la posibilidad de que el estudiante rellene con palabras o frases como respuesta.

JCloze crea ejercicios para rellenar en el espacio. Hasta 100 respuestas correctas pueden especificarse para cada espacio en blanco, además la posibilidad de incluir una pequeña pista para cada espacio.

JCross diseña crucigramas que se pueden rellenar on-line. Puedes usar parrillas de hasta 20x20 letras.

JMix te permite crear ejercicios de ordenación de frases. Puedes especificar hasta 100 respuestas correctas distintas.

JMatch crea ejercicios de emparejamiento u ordenación. Una lista de objetos fijos aparecen a la izquierda (pueden ser imágenes o texto), y una lista de objetos desordenados a la derecha.

Hot Potatoes soporta el uso de acentos y te facilita la opción de acceder al código de las páginas para cualquier modificación.

2.2.8 Educación infantil

La educación infantil constituye la primera etapa del sistema educativo. Su extensión temporal alcanza hasta los 6 años, momento que señala el acceso a la escolaridad obligatoria. Se articula en dos ciclos: el primero se extiende hasta los 3 años y el segundo desde los 3 hasta los 6 años. No es una etapa obligatoria, de hecho, no existe ningún país europeo en que la obligatoriedad de la enseñanza esté fijada antes de los 5 o 6 años de edad. Las especiales características de la educación infantil, junto con el papel todavía preponderante que en esta etapa educativa desempeña la familia, desaconsejan la obligatoriedad. No obstante, la Administración Central coordina la oferta de puestos escolares de educación infantil en las distintas administraciones públicas, reconociendo de esta manera el carácter educativo de este período, cuya finalidad es contribuir al desarrollo físico, personal y regulando las condiciones que deben cumplir los centros que implanten la etapa infantil.

2.2.8 Evaluación

“Dar una nota es evaluar, hacer una prueba es evaluar, el registro de las notas se denomina evaluación. Al mismo tiempo varios significados son atribuidos al término: análisis de desempeño, valoración de resultados, medida de capacidad, apreciación del “todo” del alumno” (Hoffman, 1999)

En el lenguaje cotidiano, el concepto de evaluación es polisémico porque éste se impone o no en la práctica según las necesidades mismas de la evaluación y en función de las diferentes formas de concebirla. En efecto, puede significar tanto estimar y calcular como valorar o apreciar. Quizá en este sentido, conviene no olvidar tampoco

desde la dimensión pedagógica las implicancias polivalentes del término: la evaluación hace referencia a un proceso por medio del cual alguna o varias características de un alumno, de un grupo de estudiantes o un ambiente educativo, objetivos, materiales, profesores, programas, etc, reciben la atención de quien evalúa, se analizan y se valoran sus características y condiciones en función de parámetros de referencia para emitir un juicio que sea relevante para la educación.

Así pues, la evaluación, en términos generales, supone una instancia de valoración. En los términos particulares de la evaluación educativa es posible distinguir varios objetos de evaluación cuyas relaciones implícitas son evidentes. Entre otros, es posible valorar: el sistema educativo, las instituciones, el profesorado, los materiales de la enseñanza, los proyectos educativos y los aprendizajes.

En el caso particular de la evaluación de los aprendizajes de los alumnos, evaluar supone conocer qué y para qué evaluar, para lo cual es requisito esencial recoger información, formular un juicio de valor y tomar decisiones con vista al futuro.

Como señala Imbernón, fue Tyler en los años cuarenta quien al plantear la educación como un proceso definió a la evaluación como una instancia en que debía establecer en qué medida se había alcanzado los objetivos establecidos inicialmente. Dos décadas después, aparecerían nuevas preocupaciones éticas y sociales respecto a la problemática evaluativa desarrollarían en la década del setenta, otros enfoques como las alternativas cualitativas. Y así, podemos llegar a concebir a la evaluación educativa ya no como una finalidad de la enseñanza sino como un medio de perfeccionamiento y mejora constante.

En virtud de este proceso histórico es posible reconocer diferentes concepciones de evaluación. Desde el paradigma cuantitativo ésta puede ser entendida como objetiva, neutral y predictiva, de manera tal que centra en la eficiencia y la eficacia. Lo que se evalúa es pues, los productos observables.

Pero desde una perspectiva cualitativa, por el contrario, la evaluación se centra en reconocer lo que está sucediendo y comprender qué significado esta tiene para las diferentes personas, en este caso no solo se evalúa el producto sino también el proceso.

Para un tercer paradigma, el crítico, la evaluación no solo se centra en recoger información sino que también implica diálogo y autorreflexión.

Otra dimensión de análisis respecto a las evaluaciones es considerar que éstas pueden ser sumativas o formativas. Mientras que las primeras se centran en el producto final, las segundas se orientan más hacia los procesos.

Funciones de la evaluación ¿Para qué evaluar?

Desde el punto de vista social, es posible identificar diversas funciones como:

a. **La selección social:** históricamente, y aún en la actualidad, la evaluación ha cumplido la función de dirigir mecanismos de selección y control social. Tal como señala Foucault: “El examen combina las técnicas de la jerarquía que vigila y las de la sanción que normaliza. Es una mirada normalizadora, una vigilancia que permite calificar, clasificar y castigar. Establece sobre los individuos una visibilidad a través de

la cual se los diferencia y se los sanciona.” (Foucault, 1993). Y en este sentido, “Es el examen, (según Díaz Barriga) el instrumento que permite invertir los problemas sociales en pedagógicos” (Litwin, 2003)

b. Medir la calidad del sistema educativo, control del sistema (3): La tendencia actual entiende a la evaluación como una actividad política y administrativa, y es una parcela de las políticas sociales y de administración pública en tanto el conjunto mismo de las políticas y los servicios públicos se han vuelto objeto de evaluación. Las causas de este fenómeno se deben a:

Razones económicas y presupuestarias: contención del gasto público y prioridades en la asignación de recursos

Presión de la opinión pública (rendición de cuentas): Exigencia de transparencia, en armonía con los principios de una gestión democrática.

Consecuentemente, se produce un cambio en la administración pública, del modelo del control vía legalidad se pasa a otro racional-burocrático, en donde se miden los resultados de las políticas propuestas.

c. Promoción, acreditación y certificación: estas funciones, aun cuando pueden ser analizadas desde un punto de vista pedagógico, poseen también claras dimensiones sociales en tanto suponen, entre otros aspectos, la legitimación de competencias profesionales frente al resto de la sociedad.

Desde el punto de vista estrictamente pedagógico, es posible identificar, siguiendo la enumeración que realiza De Ketele, las siguientes funciones para la evaluación:

a. **Certificación:** Supone un balance respecto a objetivos terminales, los macro-objetivos que integran un número relativamente significativo de objetivos intermedios. Lo relevante en términos de certificación se refiere particularmente a la integración. Por otra parte, la certificación es una decisión dicotómica en tanto ésta se otorga o no.

b. **Clasificación en el interior de una población:** Esta función se refiere a la realización de un balance de objetivos de perfeccionamiento.

c. **Balance de objetivos intermedios:** Esto supone promediar resultados intermedios para obtener una valoración global, la cual debe pronunciarse respecto al éxito o el fracaso de todo el proceso.

d. **Diagnóstico:** El objetivo es poder describir una situación para tomar decisiones de ajuste.

e. **Clasificación:** El objetivo es establecer niveles para tomar decisiones respecto a la organización de la población en subgrupos. Estos subgrupos pueden responder a criterios de selección homogénea o heterogénea según el caso.

f. **Selección:** Supone un criterio de nivel mínimo requerido, y representa el típico sistema del “examen de ingreso”.

g. **Predicción:** Fundamentada en investigaciones previas postula la estabilidad de las condiciones en las que se ha realizado la observación valorativa,

h. **Jerarquización:** Supone el orden en que deben abordarse diferentes objetivos pedagógicos.

Instrumentos de evaluación ¿Cómo evaluar?

La evaluación, como hemos mencionado, puede ser utilizada como un instrumento de control social que fija parámetros y legitima niveles de acreditación. Sin embargo, una mirada reflexiva sobre las prácticas pedagógicas, implica concebirlas como herramientas que permiten identificar el modo en que el alumno construye su conocimiento. Esto implica:

1.-Comprender el significado de las respuestas elaboradas por el alumno.

2.-Considerar el tipo de información relevada por el docente en relación al proceso de aprendizaje y al proceso de producción.

¿Qué tipo de requisitos deben tenerse en cuenta al tomar decisiones respecto a la evaluación?

Los instrumentos de evaluación, no pueden plantearse al margen de los criterios de validez, confiabilidad, practicidad y utilidad que mencionaremos a continuación:

* Validez: se refiere al grado de precisión con que se mide lo que se desea medir. En este sentido es absolutamente relevante la muestra sobre la cual se ejecuta la medición. Porque no se trata de determinar si el instrumento es o no válido. La validez se refiere siempre a los resultados, para lo cual deben considerarse el uso que se hará de éstos. “Cuando se requiere determinar si un instrumento es válido se requiere, entonces, información acerca de los criterios que han presidido su construcción y administración. Los criterios son entonces, externos a la evaluación misma” (Camilioni, 2003)

Existen por otra parte, siguiendo a Camilioni, diferentes clases de validez, entre las que podemos identificar:

- **Validez de contenido:** deberá representar una muestra significativa del universo de contenido cubierto por el curso o la unidad didáctica, en todos los casos, deberá referir a los contenidos más importantes según sea su significatividad en el aprendizaje de los alumnos.
- **Validez predictiva:** se refiere a la correlación existente entre los resultados obtenidos en una o varias pruebas combinadas y el desempeño posterior del alumno en aspectos que corresponden al área evaluada por esas pruebas.
- **Validez de construcción:** supone la coherencia del instrumento respecto al marco teórico que sustenta el proyecto pedagógico.

- **Validez de convergencia:** establece la relación que existe entre unos programas de evaluación o un instrumento y otros programas o instrumentos de validez ya conocida.
 - **Validez manifiesta:** se refiere al modo en que los instrumentos aparecen frente al público externo. La razonabilidad debe ser visible y explicable a fin de que no sólo sea válida sino que también parezca serlo.
 - **Validez de significado:** se refiere a la relación que se establece desde la perspectiva de los estudiantes entre el programa y los instrumentos de evaluación, por un lado y los procesos de enseñanza y aprendizaje por el otro.
 - **Validez de retroacción:** cuando la evaluación tiene un efecto normativo sobre los contenidos de la enseñanza, esto es, la evaluación establece lo que se debe enseñar.
-
- **Confiabilidad:** se refiere al grado de exactitud con que se mide un determinado rasgo. La confiabilidad debe ser estable y objetiva, independientemente de quien utiliza un programa o un instrumento de evaluación. “Un instrumento confiable permite aislar los aspectos que mide de otros que para el caso se consideran irrelevantes” (Camilioni: 2003)
 - **Practicidad:** se refiere a la viabilidad de la construcción, administración y análisis de resultados.

- Utilidad: refiere a la medida en que una evaluación resulta útil para la orientación tanto de los alumnos como de los docentes, la escuela o a los sectores interesados en la calidad de la educación.

Tipos de instrumentos de evaluación

En la carpeta de trabajo se mencionan diversos instrumentos de evaluación y se clasifican según estén orientadas hacia los procesos o hacia los productos. En el primer grupo se mencionan los portafolios, las entrevistas, las observaciones documentadas, los registros de aprendizajes diarios, la auto evaluación, los informes de entrevistas sobre proyectos, productos y muestras del alumnos y la expresión en voz alta del pensamiento del alumno. Más orientadas a la evaluación de productos encontramos instrumentos tales como las pruebas de ensayo (que pueden ser de respuesta extensa o limitada), los proyectos, las carpetas de trabajos, Muestras, Investigaciones, Productos de expresión artística y test estandarizados del tipo de “Multiplechoice”.

De Ketele, describe una serie de estrategias de evaluación que, en nuestra opinión, conforman una buena orientación al momento de definir cuál sería el instrumento de evaluación más adecuado:

Evaluación clásica puntual empírica: en esta categoría ubica los exámenes escritos tradicionales y la “lección oral” tradicional que denomina “Entrevista libre”. Al respecto de este tipo de evaluaciones señala que a menudo no están establecidos claramente los criterios de evaluación porque muchas veces se trata de criterios implícitos.

Evaluación centrada en los objetivos: incluye aquí otros instrumentos de evaluación que se diseñan conforme a parámetros más precisos, como los tests o entrevistas centradas en objetivos, así como el análisis de contenidos. Estos instrumentos se caracterizan por poseer criterios explícitos aunque varíe su nivel de puntualización.

Evaluación durante el proceso: aquí se encuentran diferentes alternativas de observación destinadas a establecer valoraciones a lo largo del proceso educativo. De Ketele denomina a estos instrumentos “observación libre, sistemática y provocada”.

Notas

1. Imbernón, Francisco (1993) “Reflexiones sobre la evaluación en el proceso de enseñanza-aprendizaje. De la medida a la evaluación”, en Revista Aula de Innovación Educativa Nro 20, Año II, Depto de Didáctica y Organización Escolar de la Universidad de Málaga
2. TYLER, RALPH (1973), “INTRODUCCIÓN Cap. 1”, en: Principios básicos del currículo, Troquel, Buenos Aires.
3. Cano García E.(1998), Evaluación de la calidad educativa, capítulo IV, ”La evaluación de la calidad de los sistemas educativos”. Editorial La Muralla, Madrid

4. DE KETELE, J. M. (1984), "Cap. 2: Evaluar para educar: ¿por qué?, ¿qué?, ¿quién?, ¿cómo?", en: Observar para educar, Visor, Madrid, pp. 13 a 27 y pp. 29 a 32.

2.3 IDEA A DEFENDER Y VARIABLES

“Con la utilización de evaluaciones académicas automatizadas en software libre, se mejorará el proceso evaluativo en los niños de los primeros años de básica en la escuela Fiscal Mixta Manuel de Jesús Calle”

Variable Independiente: **Software libre**

Variable Dependiente: **Evaluación académica en los primeros años de básica**

CAPITULO III

MARCO METODOLÓGICO

3.1 MODALIDAD DE LA INVESTIGACIÓN

Una investigación es un proceso sistemático, organizado y objetivo, cuya finalidad es responder a una pregunta o hipótesis y así aumentar el conocimiento y la información sobre algo desconocido.

Podemos manifestar que para este trabajo investigativo aplicaremos dos modalidades de investigación las cuales son:

3.1.1 Investigación Cuantitativa. Esta modalidad de investigación trata de describir, contextualizar o explicar con técnicas estadísticas el objeto de estudio. Parte de las técnicas inductivas (razonamientos que se desarrollan de los casos particulares hasta la generalización). En nuestro caso las muestras han sido representativas con relación al Universo y han permitido la evaluación del proceso.

3.1.2 Investigación Cualitativa.- En cambio esta investigación trata de comprender y explicar argumentativamente el objeto de estudio, considerando su contexto histórico, tecnológico y socioeconómico. Se aplicó directamente esta modalidad para evaluar aspectos de comportamiento humano, es decir se investigó los procesos operativos de la Institución, en el momento de evaluar a los niños.

3.2 TIPOS DE INVESTIGACIÓN

Se pueden mencionar los siguientes tipos de investigación:

Bibliográfica.- Este tipo de investigación ha permitido fundamentar científicamente las variables de la presente tesis y se utiliza para elaborar el marco teórico.

De Campo.- En cambio con este tipo de investigación se ha podido valor la problemática y ratificarla y luego nos orienta a una propuesta de solución.

3.3 POBLACION Y MUESTRA

La población o universo de nuestra investigación está compuesta de la siguiente forma

TIPO DE FUNCION	N° DE PERSONAS
Director	1
Profesores	20
Padres de Familia	150
TOTAL	171

3.3.1 Muestra de la Investigación:

Partiendo del hecho que la muestra es un porcentaje del universo, la hemos calculado en base a la siguiente fórmula:

$$N = \frac{\text{Población}}{(\text{Población} - 1) * \text{Error}^2 + 1}$$

Donde:

N: Tamaño de la muestra

Error.= 0,05 ó 5%

Reemplazando tenemos:

$$N = 171 / ((170) * 0,05^2 + 1)$$

$$N = 171 / ((170 * 0,0025) + 1)$$

$$N = 171 / (1,425)$$

$$N = 120$$

3.4 METODOS, TECNICAS E INSTRUMENTOS

Con el fin de obtener los mejores resultados posibles se ha seleccionado el método inductivo – deductivo para efectos investigativos.

Este método permite mediante un conjunto de preguntas adecuadas inducir las respuestas sobre la ratificación de la existencia del problema, una vez que esto a sucedido se deduce las posibles soluciones al problema.

- **Técnicas**

Entre las técnicas para recopilar información dentro de este proceso investigativo podemos mencionar: **La encuesta:** que consiste en investigar lo sintomatología del problema en base a una serie de preguntas relacionadas con las dificultades que atraviesa la Institución. **La entrevista** es una indagación directa con una persona directamente relacionada con el problema, en nuestro caso lo podremos hacer con el Director. Finalmente **la observación** que permitirá evaluar los procesos de manera visual.

- **Instrumentos**

Se utilizaran los siguientes instrumentos en las técnicas mencionadas anteriormente:

- **Cuestionarios** (Se utilizaran para las encuestas respectivas)
- **Guía de entrevista** (Entre vista directa de necesidades que hemos encontrado en la manera de evaluación a los niños de los primeros años)
- **Libreta de notas** (Registrará los sucesos especiales en la observación de los procesos)

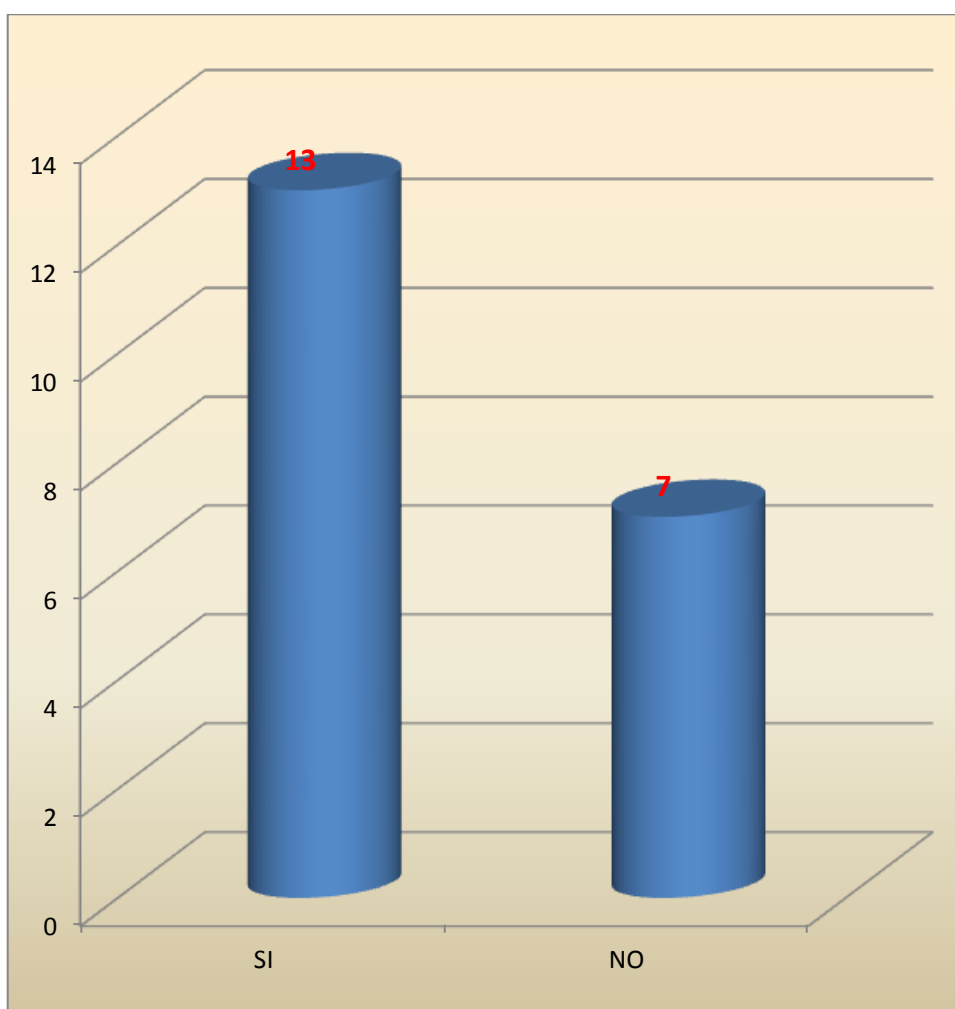
3.5 INTERPRETACIÓN DE DATOS

Para la interpretación de datos vamos a tabular y graficar los resultados de la encuesta, dicha realización lo haremos pregunta por pregunta, iniciamos con el cuestionario dedicado al personal docente de la Escuela Fiscal Mixta Manuel de Jesús Calle.

1.- ¿Las evaluaciones que realizan los maestros son repetitivas cada año escolar?

	FRECUENCIA	PORCENTAJE
SI	13	65%
NO	7	35%

TOTAL	20	100%
-------	----	------

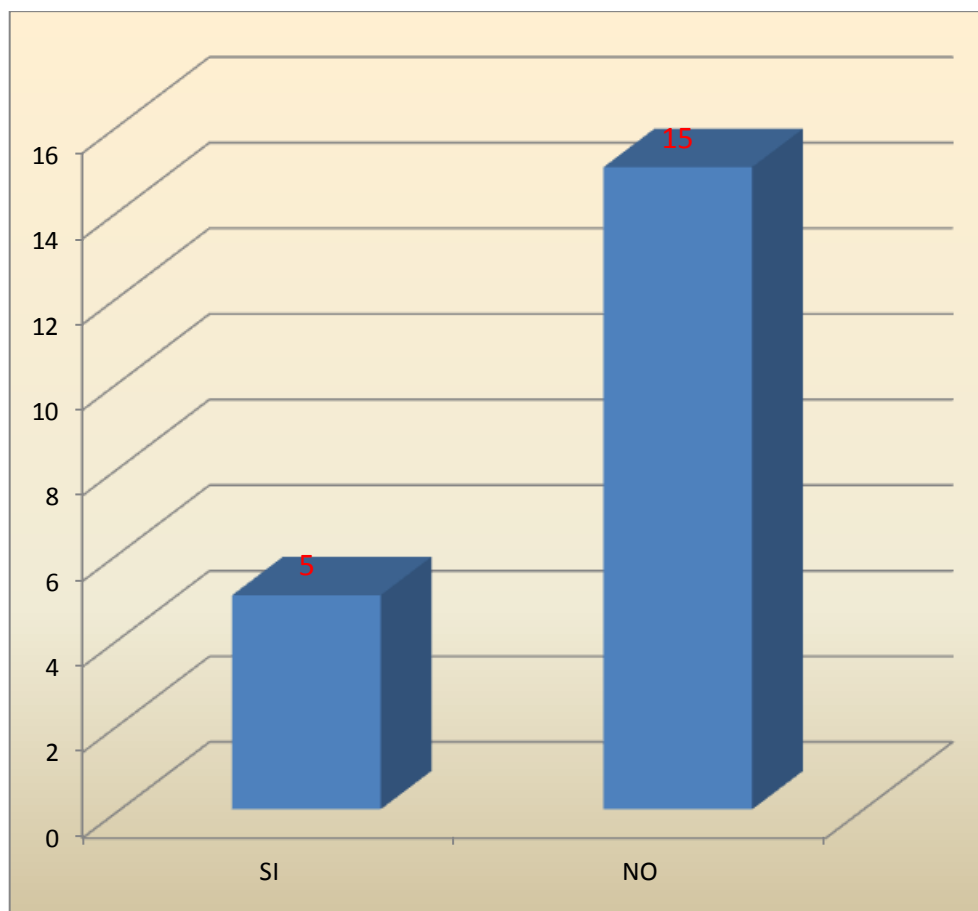


Se puede apreciar de un 65% considera que las evaluaciones son repetitivas cada año escolar.

2.- ¿Cree usted que los maestros tiene el apoyo de las nuevas tecnologías (tics) en el proceso evaluatorio a los niños?

	FRECUENCIA	PORCENTAJE
SI	5	25%
NO	15	75%

TOTAL	20	100%
-------	----	------

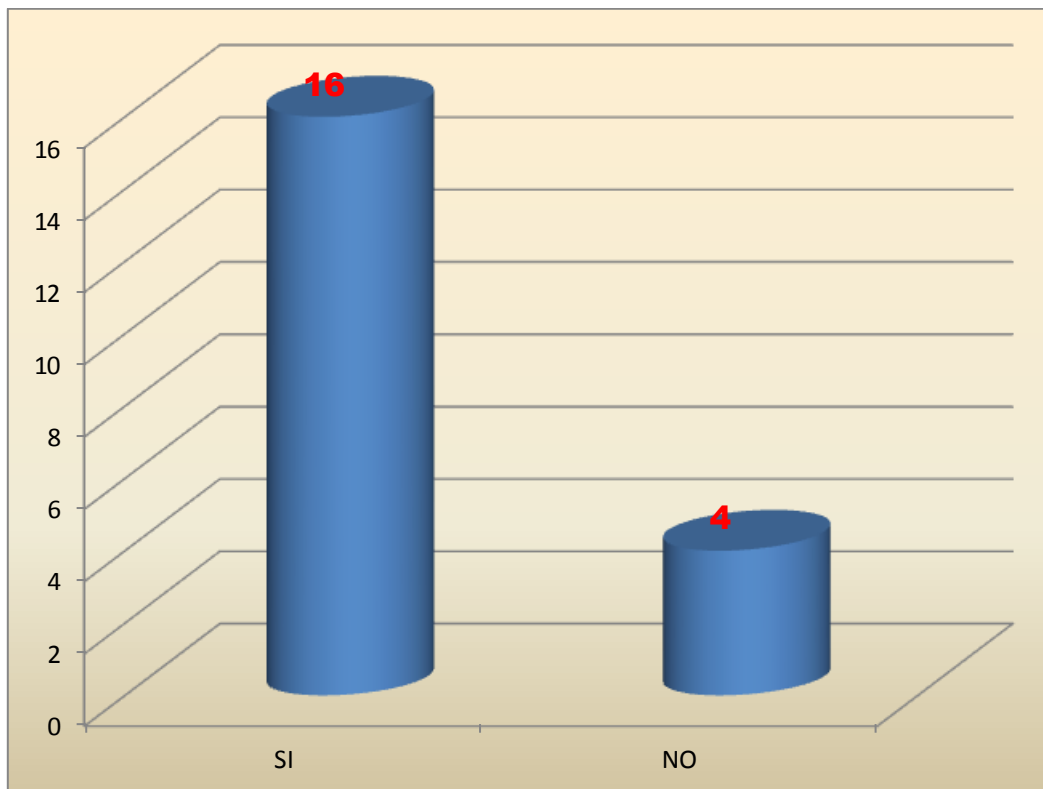


El 75% de los maestros señalaron que no utilizan las nuevas tecnologías en el proceso evaluatorio a los niños de la escuela.

3.- ¿Usted cree que los niños llegan temerosos y nerviosos al proceso evaluatorio tradicional?

	FRECUENCIA	PORCENTAJE
SI	16	80%
NO	4	20%

TOTAL	20	100%
--------------	----	------

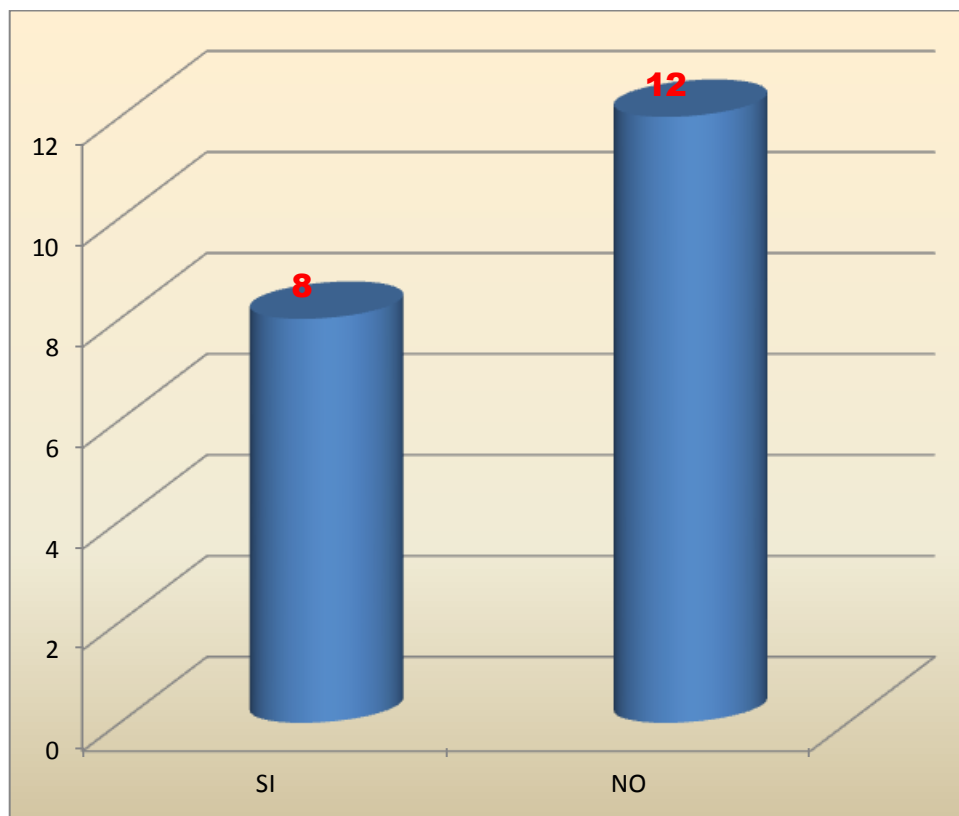


Para los maestros es claramente notorio ya que el 80% de ellos han dicho que sus alumnos llegan al proceso evaluatorio nerviosos y temerosos al momento de las pruebas.

4.- ¿Considera usted que en el proceso de evaluación, muchos de los niños se distraen fácilmente?

	FRECUENCIA	PORCENTAJE
SI	8	40%
NO	12	60%

TOTAL	20	100%
-------	----	------

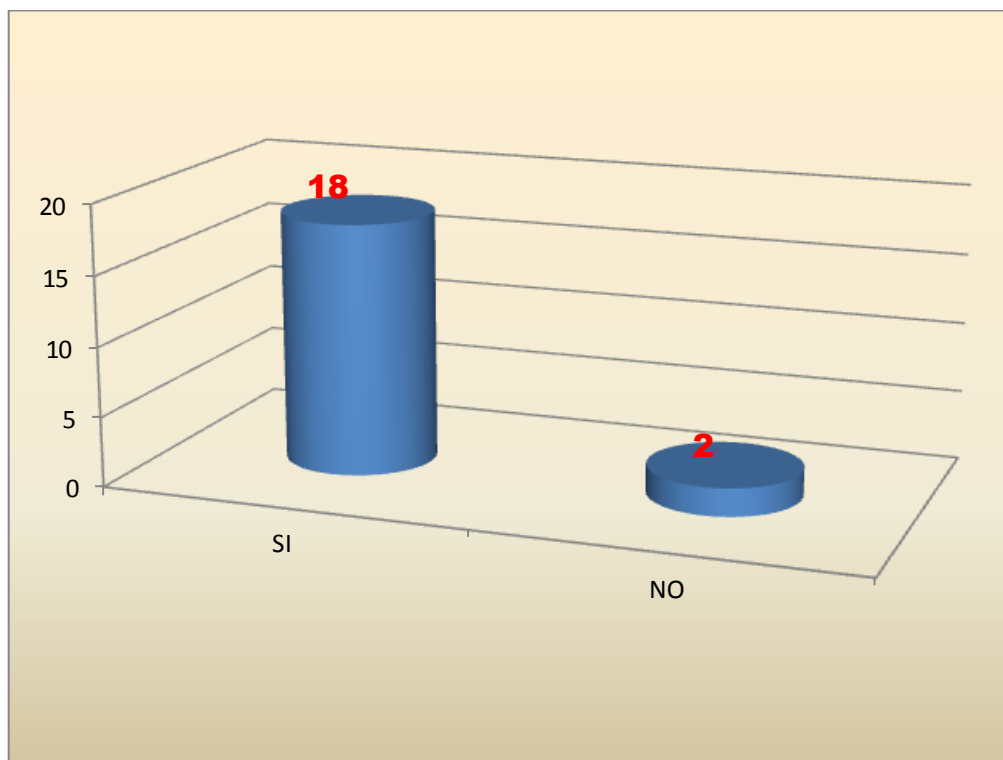


El 40% de los maestros encuestados han notado que sus alumnos se distraen fácilmente al momento de las evaluaciones.

5.- ¿Considera usted que el trabajo del profesor se aumenta al momento de volver a evaluar a los niños con bajo aprovechamiento?

	FRECUENCIA	PORCENTAJE
SI	18	90%
NO	2	10%

TOTAL	20	100%
-------	----	------

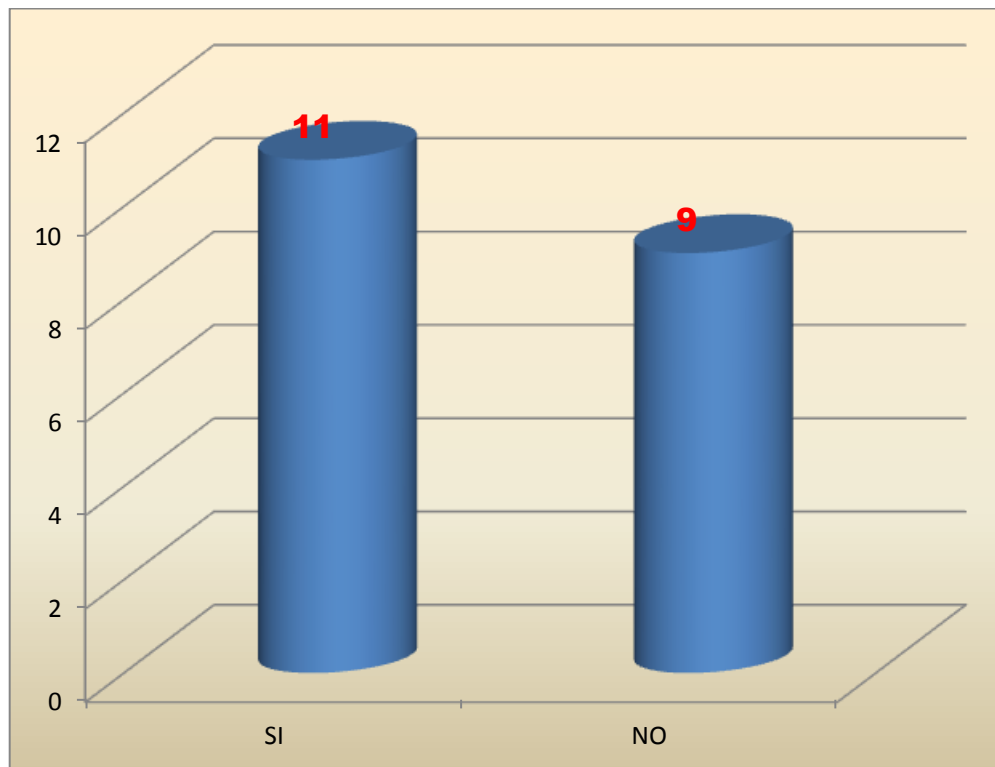


El 90% de los maestros han afirma que su trabajo se incrementa el momento de volver a evaluar a los alumnos que tiene un aprovechamiento muy bajo.

6.- ¿Cree usted que con la implementación de un sistema para la evaluación de los alumnos se mejorara el rango de promedios en la escuela?

	FRECUENCIA	PORCENTAJE
SI	11	55%
NO	9	45%

TOTAL	20	100%
--------------	----	------



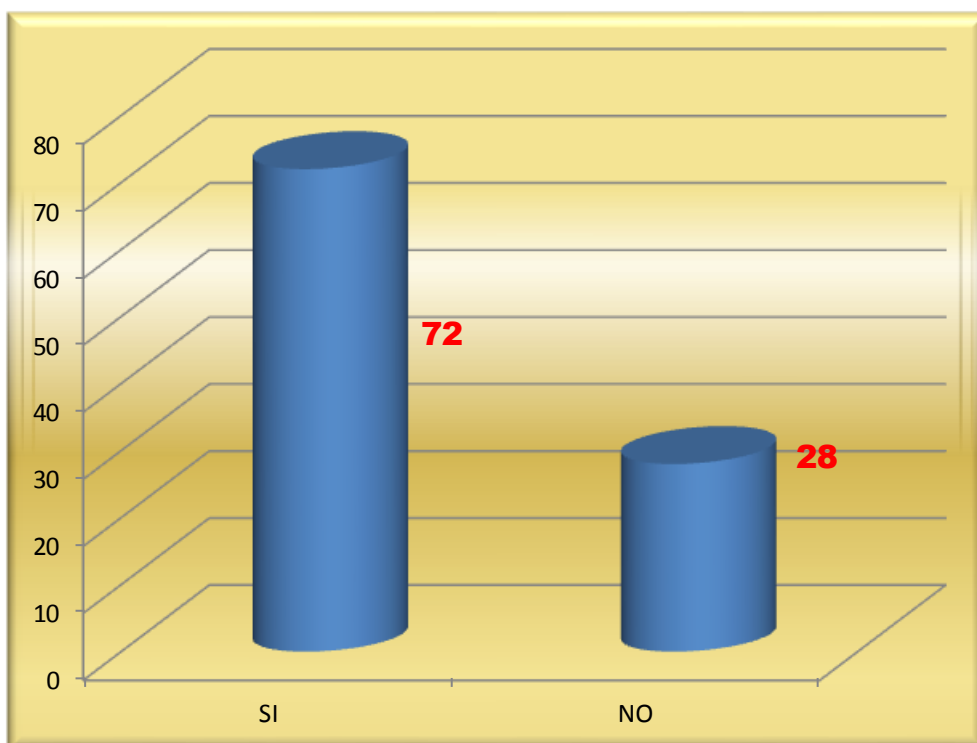
El 55% de los encuestados afirma la implementación de un sistema para la evaluación de los niños, y se mejorara el promedio global de los niños en la escuela.

Encuesta realizada a los padres de familia de la Escuela Manuel de Jesús Calle.

1.- ¿Considera usted que se debería mejorar la manera de evaluación a los niños en la Escuela?

	FRECUENCIA	PORCENTAJE
SI	72	72%
NO	28	28%

TOTAL	100	100%
--------------	-----	------

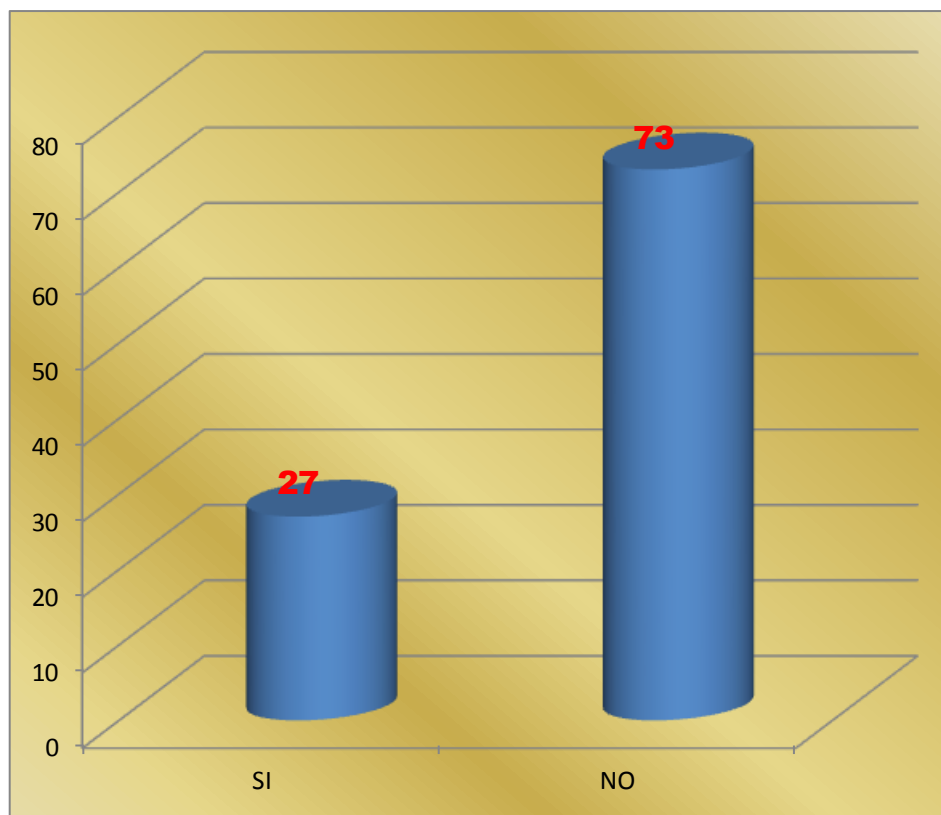


El 72% de los padres de familia dijeron que se debe mejorar el sistema de evolución de sus niños.

2.- ¿Cree usted que los niños se distraen fácilmente durante el proceso evaluatorio tradicional y eso produce una imagen de bajo rendimiento?

	FRECUENCIA	PORCENTAJE
SI	27	27%
NO	73	73%

TOTAL	100	100%
-------	-----	------

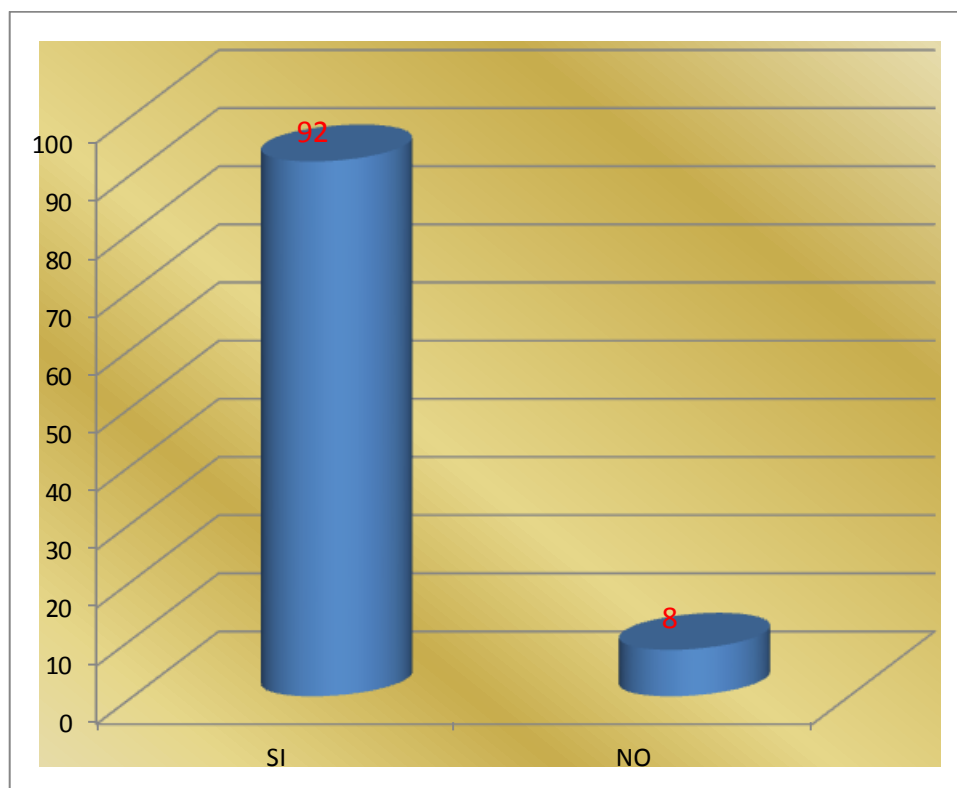


Se puede apreciar que el 73% de padres de familias consideran que sus hijos no se distraen en el proceso de evaluaciones.

3.- ¿Cree usted que se debe mejorar el sistema de evaluación y calificación de los niños de la Escuela?

	FRECUENCIA	PORCENTAJE
SI	92	92%
NO	8	8%

TOTAL	100	100%
--------------	-----	------

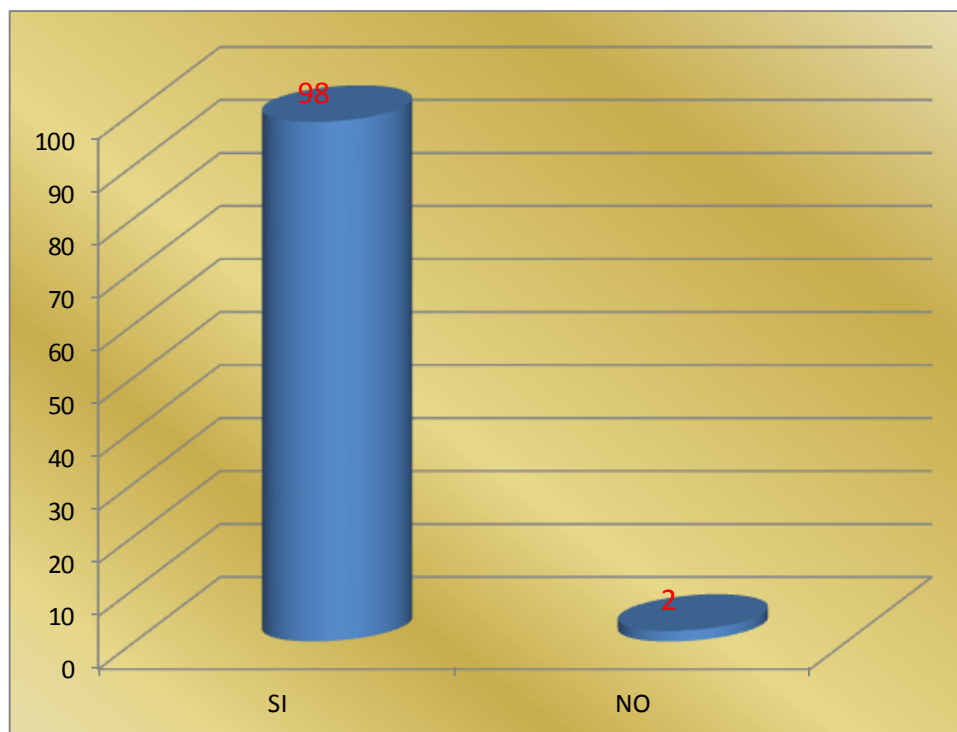


El 92% dicen que se debe mejorar el sistema de evaluación y calificación de los niños en la escuela.

4.-¿Cree usted que la implementación de nuevas tecnologías deben ser parte del proceso de mejoramiento en la educación fiscal?

	FRECUENCIA	PORCENTAJE
SI	98	98%
NO	2	2%

TOTAL	100	100%
--------------	-----	------

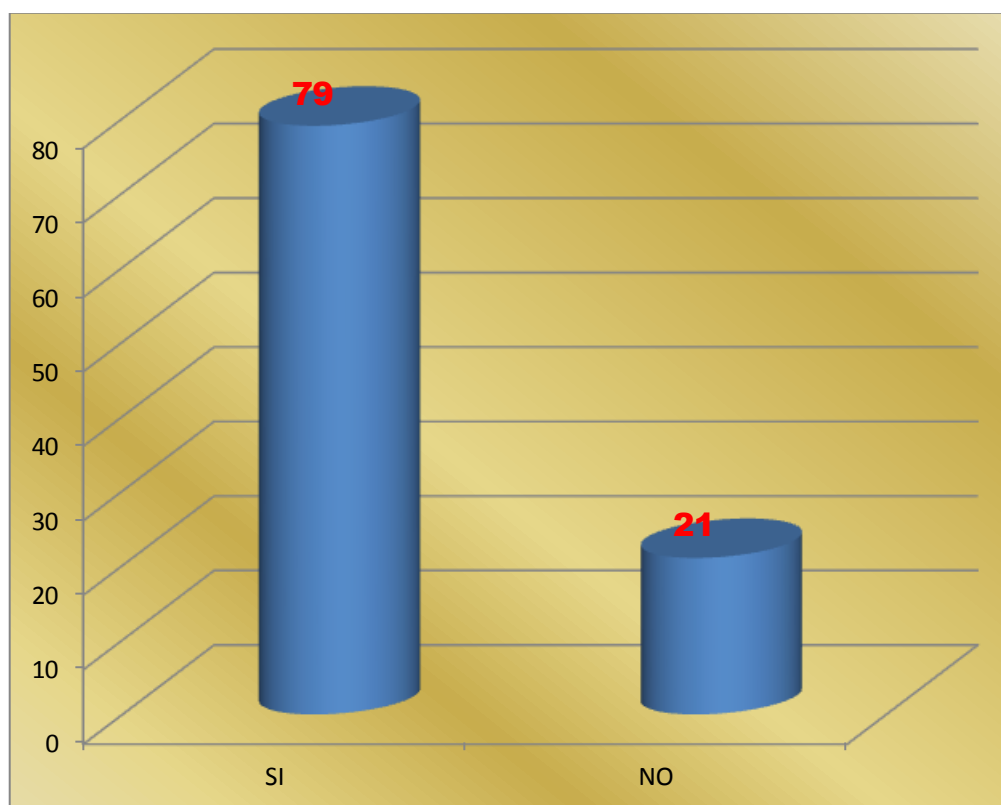


Es muy notorio que los padres de familia consideran que se debe implementar nuevas tecnologías en la educación de los niños.

5.- ¿Cree usted que la implementación de nuevas tecnologías permitirán mejorar el proceso enseñanza – aprendizaje y por ende la evaluación del mismo?

	FRECUENCIA	PORCENTAJE
SI	79	79%
NO	21	21%

TOTAL	100	100%
-------	-----	------



El 79% considera que las nuevas tecnologías se constituyen en un apoyo al proceso enseñanza-aprendizaje y por ende a la evaluación.

3.7 CONCLUSIONES

Las conclusiones que se emiten a continuación, son obtenidas de la investigación de campo realizada y tenemos:

- Se tiene evaluaciones repetitivas cada año escolar.
- Los maestros no se apoyan en nuevas tecnologías para el proceso evaluatorio
- Se considera que la mayoría de los niños llegan nervioso y temeros al proceso evaluatorio con sus maestros.
- Los niños se distraen fácilmente durante el proceso de evolución.
- El trabajo del maestro se aumenta más en el momento de evaluar a alumnos con menor promedio.
- Se considera que la implantación de un sistema mejorara mucho el promedio de los niños en el plantel.

3.8 RECOMENDACIONES

De las conclusiones obtenidas surgen las siguientes recomendaciones:

- Implementar el uso de las nuevas tecnologías para el proceso evaluatorio. Estas tecnologías deben estar basadas en software libre.
- Implementar un portal web en el cual se incorporen las diferentes evaluaciones de cada una de las materias.
- Capacitar al personal docente en el uso de las nuevas tecnologías.
- Facilitar y mejorar el nivel de conocimiento de los niños implementando el uso de las computadoras en red.

CAPITULO IV

MARCO PROPOSITIVO.

4.1 Tema: APLICACIÓN INFORMÁTICA EN SOFTWARE LIBRE PARA MEJORAR EL PROCESO DE EVALUACIÓN ACADEMICA A NIÑOS DE LOS PRIMEROS AÑOS DE EDUCACIÓN BÁSICA DE LA ESCUELA FISCAL MIXTA “MANUEL DE JESÚS CALLE”

4.2. FUNDAMENTACIÓN.

El presente trabajo investigativo está fundamentado en algunos temas de tipo informático así como de tipo educativo.

En el aspecto informático deberemos señalar que se han utilizado algunas herramientas relacionadas con el diseño Web, en primer lugar mencionaremos a Dreamweaver de Macromedia como el entorno de elaboración de las páginas en el entorno administrativo. En lo que se refiere al lenguaje que permite la comunicación con la base de datos, señalaremos que se trabajó con PHP. La base de datos que contiene toda la información de la Escuela es el motor conocido como MySQL. Se debe mencionar también que en la elaboración de la aplicación web, se ha intervenido algunas herramientas adicionales tales como Flash Macromedia, Java script, hotpotatoes y por supuesto editores gráficos como el Photoshop y Corel Draw.

Para el funcionamiento de la aplicación en su fase de diseño, se utilizó como servidor de páginas web al APACHE, que viene incorporado en el Software Wampserver.

4.3. OBJETIVOS DE LA PROPUESTA

4.3.1 Objetivo General

Mejorar el nivel evaluativo de los niños de los primeros años de la Escuela Fiscal Mixta “**Manuel J. Calle**” en base al apoyo de una aplicación Web desarrollada con herramientas del software libre

4.3.2 Objetivos Específicos.

- Diseñar una base de datos que almacene información relativa de la escuela.
- Diseñar todas las opciones que van a constituir tanto el modulo administrativo como el modulo del usuario en el portal web.
- Desarrollar las páginas web dinámicas y estáticas que corresponden a cada opción de los módulos.
- Difundir el uso de la aplicación Web entre los maestros, padres de familia y alumnos de la escuela, para que se constituya en estrategia importante dentro de la imagen de la escuela.
- Lograr que la aplicación Web se convierta en un nexo directo entre los maestros sector estudiantil y todos los que deseen ingresar a la página a nivel mundial.

4.4. DESCRIPCION DE LA PROPUESTA

La propuesta planteada como solución al problema descrito en el capítulo I consiste en una **Aplicación Web de tipo dinámica**, la cual permitirá que los alumnos y padres de familia accedan directamente a varias de los procesos que se encuentran en dicha página.

Esta aplicación ha sido elaborada en las siguientes etapas:

4.4.1. METODOLOGIA DE DESARROLLO.

Muchas veces es necesario inclinarse por una metodología de desarrollo de software cuando se trata de elaborar sistemas o aplicaciones de diferente tipo, pero ¿cuál de las estudiadas o conocidas se adapta mejor a mi campo?, pues es muy difícil adoptar una se adapta plenamente a los requerimientos que el usuario tiene. Por ello me he inclinado por una metodología de desarrollo de software libre, dentro de la cual la metodología lineal o en cascada es la primordial. A esta metodología se le agrega una realimentación de tipo espiral para tratar de lograr un mejor alcanzar de mejor manera captar los requerimientos que tiene el usuario.

Por demás esta mencionar que la metodología lineal involucra algunos pasos en serie como son: Análisis, diseño, desarrollo, pruebas e implementación y documentación del software elaborado.

4.4.2. ANALISIS PREVIO.

Procedemos a describir algunos de los procesos que se realizan en la escuela y que nos servirán de base para el diseño de la aplicación web. Estos procesos tienen relación directa con la parte educativa y la parte administrativa.

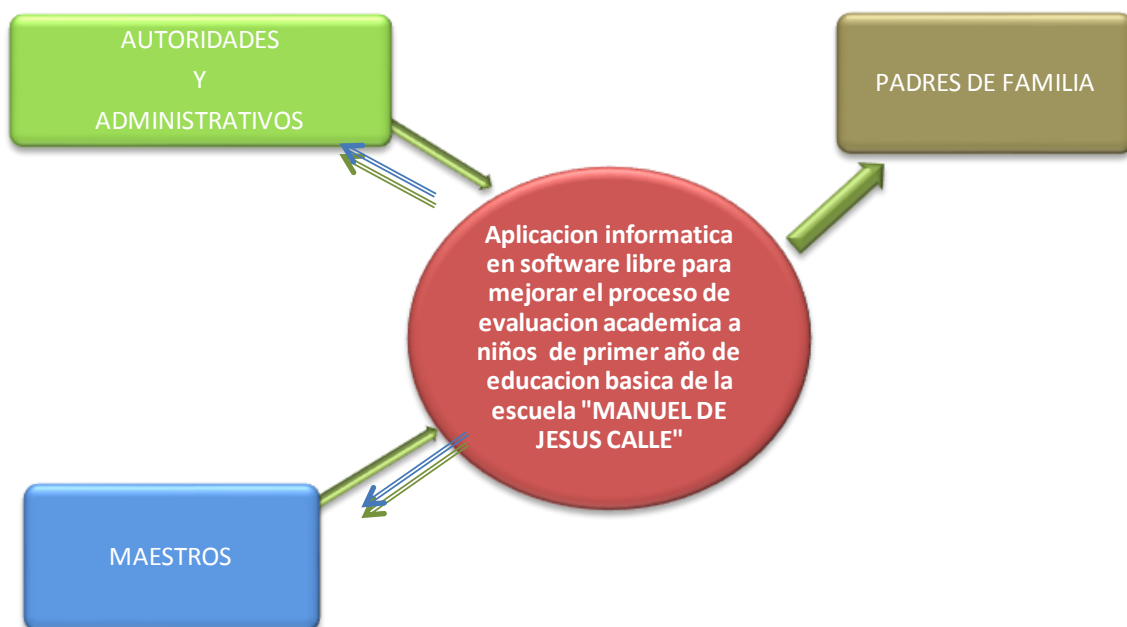
En la parte educativa se realizaron las siguientes actividades:

- Visitar a los docentes en sus labores cotidianas para obtener información como:
 - Temática tratada.
 - Libros y textos utilizados como apoyo.
 - Tipos y formas de evaluación a sus alumnos.
 - Frecuencia de evaluación.

- Se realizó una prueba a los niños para ver el grado de conocimiento sobre el uso de las computadoras.

- En la parte administrativa se realizaron las siguientes actividades:
 - Visitar a las autoridades principales del plantel “Rector Encargado” Lcdo. Piño Vaca, para solicitar información relacionada con el plantel, entre ellas:
 - Historia y Evolución.
 - Ubicación.
 - Listado de personal docente y administrativo.

FLUJO DE INFORMACIÓN



4.4.3. DISEÑO.

4.4.3.1. DISEÑO DE LAS OPCIONES PRINCIPALES DE LA APLICACIÓN.

Del análisis previo realizado, se deduce que deberemos disponer de algunas opciones informativas y que deberá también existir una parte dinámica que maneje todo lo referente a evaluaciones. Algunas de las opciones de la parte informativa estarán vinculadas a la base de datos con la finalidad de que la aplicación tenga mejor estructuración.

Según lo manifestado anteriormente se tendrán dos partes que son:

4.4.3.2. MODO USUARIO

Las opciones de este modo corresponden a aquellas que el usuario mira al cargar la página Web, entre las opciones tenemos:

INICIO. Se constituirá en la página inicial de la aplicación.

LA ESCUELA.- Esta opción involucra aspectos relativos a la historia de la escuela acompañadas con imágenes de la misma.

PERSONAL.- En cambio en esta parte se puede visualizar información relacionada con el personal administrativo y docente que labora en la institución, también se puede mirar el listado de las autoridades.

ACTIVIDADES- Se podrá observar el listado de acciones educativas para con la colectividad, éstas están divididas en curriculares y extracurriculares

EVALUACIONES- Aquí se podrá observar el listado de cuestionarios que corresponden a cada evaluación mensual de los niños de primer y segundo año.

CONTACTOS.- Opción de comunicación, posibilita la recepción de correos, de nuestra ubicación y nuestros teléfonos.

4.4.3.3 MODO ADMINISTRATIVO

En este modo se puede administrar la parte dinámica de la aplicación, su acceso está restringido al administrador del sistema y a los usuario que el considere conveniente. Todas las opciones tiene características dinámicas es decir involucran el manejo de información que se almacena en la base de datos. Entre los módulos tenemos:

DOCENTES.- Manipula toda la información relacionada a los profesores que laboran en la institución.

ADMINISTRATIVOS.- En esta opción podemos manejar el listado de docentes y todo el personal que labora en la institución.

ACTIVIDADES.- Podemos controlar toda la información relacionada con las acciones educativas que se dan en la institución.

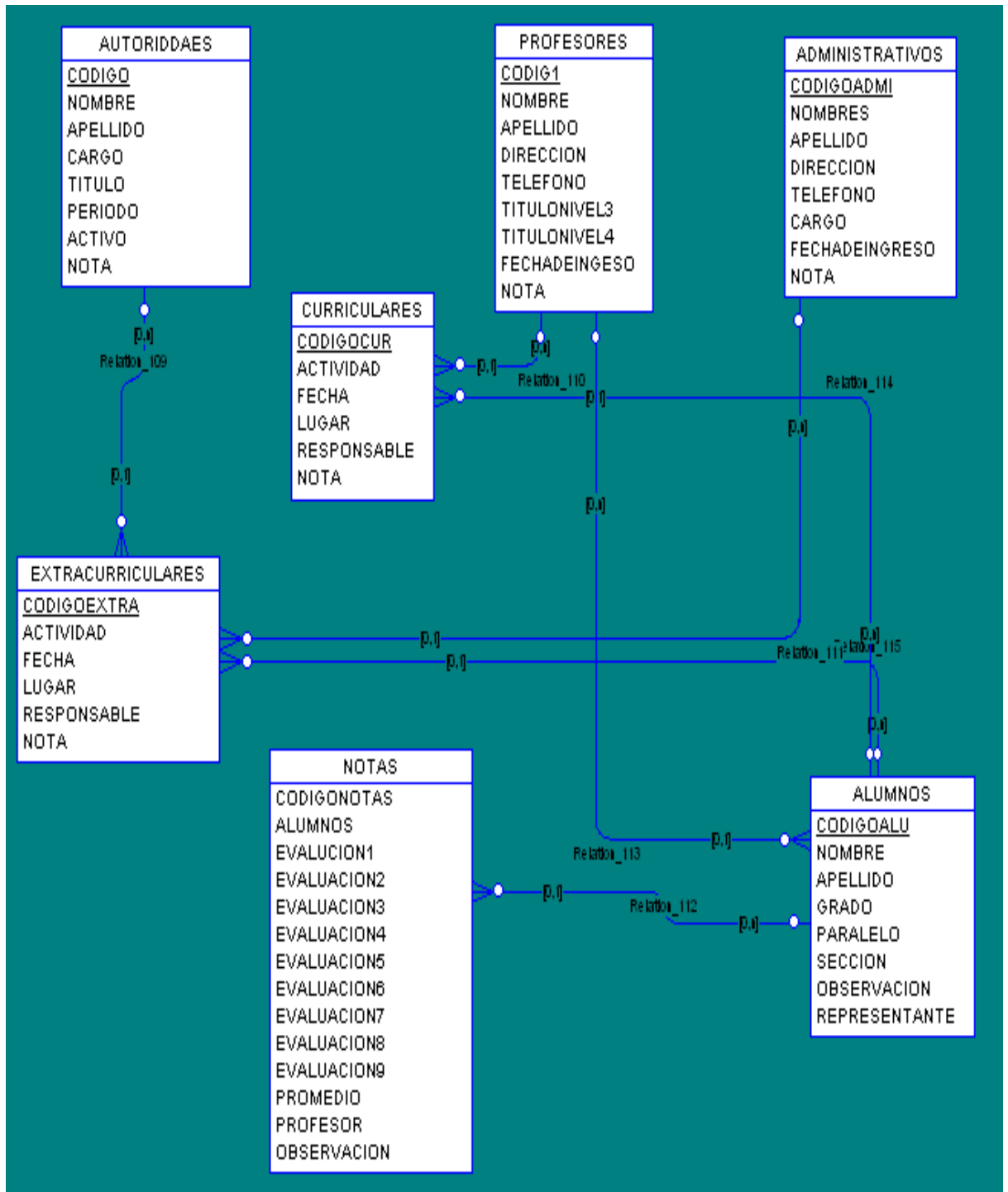
AUTORIDADES.- Podemos manipular el listado del personal administrativo de la institución.

NOTAS.- En esta opción se puede elaborar el ingreso y registro de las calificaciones correspondientes a los alumnos de los primeros años.

SALIR.- Nos permite abandonar del modo administrativo.

4.4.3.4. DISEÑO DE LA BASE DE DATOS.

A continuación se expone el modelo de la base de datos, se debe mencionar que la modelación se la ha realizado en Power Designer para web, luego se ha generado el strip de la base y se ha lo ha exportado al MySQL.



4.4.4. MODELO LOGICO DE LA BASE DE DATOS

4.4.4.1. SCRIP DE LA BASE DE DATOS

```
/* ===== */
/* Database name: Basemjc */
/* DBMS name: Microsoft SQL Server 7.x */
/* ===== */
/* ===== */
/* Table: ADMINISTRATIVO */
/* ===== */
```

```
CREATE TABLE `administrativo` (
  `codigo` int(11) NOT NULL auto_increment,
  `nombre` varchar(30) NOT NULL,
  `apellido` varchar(30) NOT NULL,
  `direccion` varchar(40) NOT NULL,
  `titulo` varchar(50) NOT NULL,
  `cargo` varchar(50) NOT NULL,
  `fecha ingreso` datetime NOT NULL,
  `nota` varchar(150) default NULL,
  PRIMARY KEY (`codigo`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=6 ;
```

-- Volcar la base de datos para la tabla `administrativo`

```
INSERT INTO `administrativo` (`codigo`, `nombre`, `apellido`, `direccion`, `titulo`,
`cargo`, `fecha ingreso`, `nota`) VALUES
(1, 'Anita', 'Morales', 'San miguel', 'Ingeniera', 'Secretaria', '0000-00-00 00:00:00',
NULL),
(2, 'Rosa', 'Escobar', 'balsapamba', 'Iecenciada', 'conserje', '0000-00-00 00:00:00',
NULL),
(3, 'mary', 'pacheco', 'quito', 'licenciada', 'asistente', '0000-00-00 00:00:00', 'xxx'),
(4, 'luis', 'rios', 'ambato', 'ingeniero', 'colector', '0000-00-00 00:00:00', 'yyyy'),
(5, 'Marcos', 'Aguirre', 'Quevedo', 'Lcdo en Educacion primaria', 'Inspector General',
'0000-00-00 00:00:00', 'Ninguna');
```

```
/* ===== */
/* Table: AUTORIDADES */
/* ===== */
```

```
CREATE TABLE `autoridades` (
`codigo` int(11) NOT NULL auto_increment,
`nombre` varchar(30) NOT NULL,
`apellido` varchar(30) NOT NULL,
`cargo` varchar(50) NOT NULL,
`titulo` varchar(50) NOT NULL,
`periodo` varchar(50) NOT NULL,
`activo` varchar(40) default NULL,
```

```

`nota` varchar(120) default NULL,

PRIMARY KEY (`codigo`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=4 ;

-- Volcar la base de datos para la tabla `autoridades`

INSERT INTO `autoridades` (`codigo`, `nombre`, `apellido`, `cargo`, `titulo`, `periodo`,
`activo`, `nota`) VALUES

(1, 'Pliño', 'Vaca', 'Rector', 'Licenciado', 'desde 1999 hasta la actualidad', 'si', 'solo es
rector encargado'),

(2, 'Gloria', 'Gaibor', 'Vicerectora', 'profesora', 'desde 1990 hasta la actualidad', 'si',
'aparte de ser maestra es vicerectora'),

(3, 'Gloria', 'urr', 'insp', 'lcda', '1999-2011', 'si', 'xxx');

/* ===== */
/*Table: CURRICULARES */
/* ===== */

CREATE TABLE `curriculares` (

`codigo` int(11) NOT NULL auto_increment,

`actividad` varchar(100) default NULL,

`fecha` datetime NOT NULL,

`lugar` varchar(100) default NULL,

`responsable` varchar(50) NOT NULL,

`nota` varchar(150) default NULL,

```

```

PRIMARY KEY (`codigo`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=4 ;

-- Volcar la base de datos para la tabla `curriculares`

INSERT INTO `curriculares` (`codigo`, `actividad`, `fecha`, `lugar`, `responsable`,
`nota`) VALUES

(1, 'libro leído', '0000-00-00 00:00:00', 'Patios de la escuela', 'Lcdo. Jorge', 'Asistencia
de padres de familia'),

(2, 'Concurso de Dibujo', '0000-00-00 00:00:00', 'Rectorado', 'Lcda Juanita Jaramillo',
'Se darà media beca al mejor'),

(3, 'oratoria', '0000-00-00 00:00:00', 'patio', 'valentin', 'xxx');

/* ===== */
/* Table: EXTRACURRICULARES */
/* ===== */

CREATE TABLE `extracurriculares` (
`codigo` int(11) NOT NULL auto_increment,
`actividad` varchar(100) default NULL,
`fecha` datetime NOT NULL,
`lugar` varchar(100) default NULL,
`responsable` varchar(50) NOT NULL,
`nota` varchar(150) default NULL,
PRIMARY KEY (`codigo`)

```

```
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=7 ;
```

```
-- Volcar la base de datos para la tabla `extracurriculares`
```

```
INSERT INTO `extracurriculares` (`codigo`, `actividad`, `fecha`, `lugar`, `responsable`,  
`nota`) VALUES
```

```
(1, 'deportiva', '0000-00-00 00:00:00', 'canchas deportivas', 'Lcdo Valentin', 'Solo  
participan estudiantes de primaria'),
```

```
(2, 'deportivas de representantes', '0000-00-00 00:00:00', 'Canchas deportivas', 'Lcdo  
Valentin', 'Solo participan los padres de los primeros años'),
```

```
(3, 'intercolegial', '0000-00-00 00:00:00', 'caluma', 'Lcda Ulloa', 'Solo participan los  
niños d 7 años'),
```

```
(4, 'Campeonato femenino de basquet', '0000-00-00 00:00:00', 'Patio de la institucion',  
'Lcda Rocio Coloma', 'Solo participan los estudiantes de los primeros años'),
```

```
(5, 'Campeonato amistoso', '0000-00-00 00:00:00', 'Patios de la escuela', 'Lcdo. Granja',  
'Campeonato amistoso entre el paralelo A y B'),
```

```
(6, 'Juego Amistoso de Indor Futbol', '0000-00-00 00:00:00', 'Babahoyo', 'Lcdo.  
Arreaga', 'Colbaoracion de padres de familia');
```

```
/* ===== */
```

```
/*Table: NOTAS */
```

```
/* ===== */
```

```
CREATE TABLE `notas` (
```

```
  `codigo` int(11) NOT NULL auto_increment,
```

```

`alumno` varchar(50) NOT NULL,
`evaluacion 1` decimal(10,0) NOT NULL,
`evaluacion 2` decimal(10,0) NOT NULL,
`evaluacion 3` decimal(10,0) NOT NULL,
`evaluacion 4` decimal(10,0) NOT NULL,
`evaluacion 5` decimal(10,0) NOT NULL,
`evaluacion 6` decimal(10,0) NOT NULL,
`evaluacion 7` decimal(10,0) NOT NULL,
`evaluacion 8` decimal(10,0) NOT NULL,
`evaluacion 9` decimal(10,0) NOT NULL,
`promedio` decimal(10,0) NOT NULL,
`profesor` varchar(50) NOT NULL,
`observacion` varchar(150) default NULL,
PRIMARY KEY (`codigo`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=4 ;

```

-- Volcar la base de datos para la tabla `notas`

```

INSERT INTO `notas` (`codigo`, `alumno`, `evaluacion 1`, `evaluacion 2`, `evaluacion
3`, `evaluacion 4`, `evaluacion 5`, `evaluacion 6`, `evaluacion 7`, `evaluacion 8`,
`evaluacion 9`, `promedio`, `profesor`, `observacion`) VALUES
(1, 'Nayeli Garcia', '19', '20', '19', '20', '19', '19', '20', '29', '19', '19', 'Gloria Gaibor', 'es
promovida al siguiente año'),
(2, 'Geovanna angulo', '17', '18', '17', '17', '17', '17', '17', '17', '17', '17', 'Gloria Lanos',
'pasa al siguiente nivel'),

```



```
(3, 'jose', '14', '0', '0', '0', '0', '0', '0', '0', '0', '0', 'Lcda Mary', NULL);
```

```
/* ===== */  
/* Table: PROFESOR */  
/* ===== */
```

```
CREATE TABLE `profesor` (  
  `codigo` int(11) NOT NULL auto_increment,  
  `nombre` varchar(30) NOT NULL,  
  `apellido` varchar(30) NOT NULL,  
  `direccion` varchar(50) NOT NULL,  
  `telefono` varchar(9) default NULL,  
  `titulo 3 nivel` varchar(30) default NULL,  
  `titulo 4 nivel` varchar(30) default NULL,  
  `fecha ingreso` datetime NOT NULL,  
  `nota` varchar(150) default NULL,  
  PRIMARY KEY (`codigo`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=8 ;
```

```
-- Volcar la base de datos para la tabla `profesor`
```

```
INSERT INTO `profesor` (`codigo`, `nombre`, `apellido`, `direccion`, `telefono`, `titulo  
3 nivel`, `titulo 4 nivel`, `fecha ingreso`, `nota`) VALUES  
(1, 'Gladis', 'Llanos', 'Balsapamba', '032456789', 'licenciada ', '', '0000-00-00 00:00:00',  
NULL),
```

(2, 'Hugo', 'Nuñez', 'Balsapamba', '986747896', 'profesor', '', '0000-00-00 00:00:00', NULL),

(3, 'Noralisa', 'Chiluisa', 'Catarama', '052567843', 'Licenciada', NULL, '0000-00-00 00:00:00', NULL),

(4, 'Carlos', 'andrade', 'balsapamba', '0987654', 'licenciado', NULL, '0000-00-00 00:00:00', NULL),

(5, 'Ernersto', 'mora', 'baba', '213456886', NULL, NULL, '0000-00-00 00:00:00', NULL),

(6, 'juan', 'pio', 'la calle', '123456789', 'licenciado', NULL, '0000-00-00 00:00:00', NULL),

(7, 'Juana', 'Jaramillo', 'Babahoyo', '052733754', 'Lcda en Educacion Primaria', 'Ninguno', '0000-00-00 00:00:00', 'Ninguna');

```

/* ===== */
/* Table: USUARIOS */
/* ===== */

```

```

CREATE TABLE `usuarios` (
  `codigo` int(11) NOT NULL auto_increment,
  `usuario` varchar(15) NOT NULL,
  `clave` varchar(15) NOT NULL,
  `nombre` varchar(20) default NULL,
  `apellido` varchar(20) default NULL,
  `nota` varchar(100) default NULL,
  PRIMARY KEY (`codigo`)

```

```
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=3 ;
```

```
-- Volcar la base de datos para la tabla `usuarios`
```

```
INSERT INTO `usuarios` (`codigo`, `usuario`, `clave`, `nombre`, `apellido`, `nota`)
```

```
VALUES
```

```
(1, 'Gaby', '123', 'Gabrilea', 'Moran', 'xxxx'),
```

```
(2, 'Marian', '234', 'Marian', 'Andrade', 'sistemas');
```

4.4.4.2. DICCIONARIO DE DATOS

TABLA ADMINISTRATIVO

	Campo	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra
<input type="checkbox"/>	codigo	int(11)			No		auto_increment
<input type="checkbox"/>	nombre	varchar(30)	latin1_swedish_ci		No		
<input type="checkbox"/>	apellido	varchar(30)	latin1_swedish_ci		No		
<input type="checkbox"/>	direccion	varchar(40)	latin1_swedish_ci		No		
<input type="checkbox"/>	titulo	varchar(50)	latin1_swedish_ci		No		
<input type="checkbox"/>	cargo	varchar(50)	latin1_swedish_ci		No		
<input type="checkbox"/>	fecha ingreso	datetime			No		
<input type="checkbox"/>	nota	varchar(150)	latin1_swedish_ci		Sí	NULL	

TABLA AUTORIDADES

	Campo	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra
<input type="checkbox"/>	<u>codigo</u>	int(11)			No		auto_increment
<input type="checkbox"/>	nombre	varchar(30)	latin1_swedish_ci		No		
<input type="checkbox"/>	apellido	varchar(30)	latin1_swedish_ci		No		
<input type="checkbox"/>	cargo	varchar(50)	latin1_swedish_ci		No		
<input type="checkbox"/>	titulo	varchar(50)	latin1_swedish_ci		No		
<input type="checkbox"/>	periodo	varchar(50)	latin1_swedish_ci		No		
<input type="checkbox"/>	activo	varchar(40)	latin1_swedish_ci		Sí	NULL	
<input type="checkbox"/>	nota	varchar(120)	latin1_swedish_ci		Sí	NULL	

TABLA CURRICULARES

	Campo	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra
<input type="checkbox"/>	<u>codigo</u>	int(11)			No		auto_increment
<input type="checkbox"/>	actividad	varchar(100)	latin1_swedish_ci		Sí	NULL	
<input type="checkbox"/>	fecha	datetime			No		
<input type="checkbox"/>	lugar	varchar(100)	latin1_swedish_ci		Sí	NULL	
<input type="checkbox"/>	responsable	varchar(50)	latin1_swedish_ci		No		
<input type="checkbox"/>	nota	varchar(150)	latin1_swedish_ci		Sí	NULL	

TABLA EXTRACURRICUALRES

	Campo	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra
<input type="checkbox"/>	<u>codigo</u>	int(11)			No		auto_increment
<input type="checkbox"/>	actividad	varchar(100)	latin1_swedish_ci		Sí	NULL	
<input type="checkbox"/>	fecha	datetime			No		
<input type="checkbox"/>	lugar	varchar(100)	latin1_swedish_ci		Sí	NULL	
<input type="checkbox"/>	responsable	varchar(50)	latin1_swedish_ci		No		
<input type="checkbox"/>	nota	varchar(150)	latin1_swedish_ci		Sí	NULL	

TABLA NOTAS

	Campo	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra
<input type="checkbox"/>	codigo	int(11)			No		auto_increment
<input type="checkbox"/>	alumno	varchar(50)	latin1_swedish_ci		No		
<input type="checkbox"/>	evaluacion 1	decimal(10,0)			No		
<input type="checkbox"/>	evaluacion 2	decimal(10,0)			No		
<input type="checkbox"/>	evaluacion 3	decimal(10,0)			No		
<input type="checkbox"/>	evaluacion 4	decimal(10,0)			No		
<input type="checkbox"/>	evaluacion 5	decimal(10,0)			No		
<input type="checkbox"/>	evaluacion 6	decimal(10,0)			No		
<input type="checkbox"/>	evaluacion 7	decimal(10,0)			No		
<input type="checkbox"/>	evaluacion 8	decimal(10,0)			No		
<input type="checkbox"/>	evaluacion 9	decimal(10,0)			No		
<input type="checkbox"/>	promedio	decimal(10,0)			No		
<input type="checkbox"/>	profesor	varchar(50)	latin1_swedish_ci		No		
<input type="checkbox"/>	observacion	varchar(150)	latin1_swedish_ci		Sí	NULL	

TABLA PROFESORES

	Campo	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra
<input type="checkbox"/>	codigo	int(11)			No		auto_increment
<input type="checkbox"/>	nombre	varchar(30)	latin1_swedish_ci		No		
<input type="checkbox"/>	apellido	varchar(30)	latin1_swedish_ci		No		
<input type="checkbox"/>	direccion	varchar(50)	latin1_swedish_ci		No		
<input type="checkbox"/>	telefono	varchar(9)	latin1_swedish_ci		Sí	NULL	
<input type="checkbox"/>	titulo 3 nivel	varchar(30)	latin1_swedish_ci		Sí	NULL	
<input type="checkbox"/>	titulo 4 nivel	varchar(30)	latin1_swedish_ci		Sí	NULL	
<input type="checkbox"/>	fecha ingreso	datetime			No		
<input type="checkbox"/>	nota	varchar(150)	latin1_swedish_ci		Sí	NULL	

TABLA USUARIOS

	Campo	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra
<input type="checkbox"/>	<u>codigo</u>	int(11)			No		auto_increment
<input type="checkbox"/>	usuario	varchar(15)	latin1_swedish_ci		No		
<input type="checkbox"/>	clave	varchar(15)	latin1_swedish_ci		No		
<input type="checkbox"/>	nombre	varchar(20)	latin1_swedish_ci		Sí	NULL	
<input type="checkbox"/>	apellido	varchar(20)	latin1_swedish_ci		Sí	NULL	
<input type="checkbox"/>	nota	varchar(100)	latin1_swedish_ci		Sí	NULL	

4.5. DESARROLLO.

En esta sección exponemos parte de la programación de las páginas dinámicas que contiene la aplicación:

```
if(typeof(loc)=="undefined"||loc==""){ var
loc="";if(document.body&&document.body.innerHTML){ var
tt=document.body.innerHTML.toLowerCase();var
last=tt.indexOf("menuadministrativo.js");if(last>0){ var
first=tt.lastIndexOf("\",last);if(first>0&&first<last)loc=document.body.innerHTML.su
bstr(first+1,last-first-1);} } }

var bd=0

document.write("<style type=\"text/css\">");
document.write("\n<!--\n");

document.write(".menuadministrativo_menu{ border-color:black;border-
style:solid;border-width:"+bd+"px0px"+bd+"px0px;background-
color:#5d5454;position:absolute;left:0px;top:0px;visibility:hidden;}");
```

```

document.write("a.menuadministrativo_plain:link,a.menuadministrativo_plain:visited{t
ext-align:left;background-color:#5d5454;color:#ffffff;text-decoration:none;border-
color:black;border-style:solid;border-width:0px "+bd+"px 0px "+bd+"px;padding:2px
0px 2px 0px;cursor:hand;display:block;font-size:10pt;font-family:Arial, Helvetica,
sans-serif;}");

document.write("a.menuadministrativo_plain:hover,a.menuadministrativo_plain:active{
background-color:#a7dbb9;color:#000000;text-decoration:none;border-
color:black;border-style:solid;border-width:0px "+bd+"px 0px "+bd+"px;padding:2px
0px 2px 0px;cursor:hand;display:block;font-size:10pt;font-family:Arial, Helvetica,
sans-serif;}");

document.write("a.menuadministrativo_1:link,a.menuadministrativo_1:visited{ textalign:
left;background:#5d5454url("+loc+"menuadministrativo_1.gif)no-repeat
right;;color:#ffffff;text-decoration:none;border-color:black;border-style:solid;border-
width:0px"+bd+"px0px"+bd+"px;padding:2px0px2px0px;cursor:hand;display:block;fo
nt-size:10pt;font-family:Arial, Helvetica, sans-serif;}");

document.write("a.menuadministrativo_1:hover,a.menuadministrativo_1:active{ backgro
und:#a7dbb9url("+loc+"menuadministrativo_12.gif)no-repeat right;color: #000000;text-
decoration:none;border-color:black;border-style:solid;border-width:0px "+bd+"px 0px
"+bd+"px;padding:2px 0px 2px 0px;cursor:hand;display:block;font-size:10pt;font-
family:Arial, Helvetica, sans-serif;}");

document.write("\n-->\n");

document.write("</style>");

var fc=0x000000;

var bc=0xa7dbb9;

if(typeof(frames)== "undefined"){ var frames=0;}

```

```

startMainMenu("menuadministrativo_left.gif",31,10,2,0,0)

mainMenuItem("menuadministrativo_b1",".gif",31,147,"javascript:;", "", "DOCENTES",
2,2,"menuadministrativo_plain");

mainMenuItem("menuadministrativo_b2",".gif",31,147,"javascript:;", "", "ADMINISTR
ATIVOS",2,2,"menuadministrativo_plain");

mainMenuItem("menuadministrativo_b3",".gif",31,147,"javascript:;", "", "ACTIVIDAD
ES",2,2,"menuadministrativo_plain");

mainMenuItem("menuadministrativo_b4",".gif",31,147,"javascript:;", "", "AUTORIDAD
ES",2,2,"menuadministrativo_plain");

mainMenuItem("menuadministrativo_b5",".gif",31,147,"javascript:;", "", "NOTAS",2,2,"
menuadministrativo_plain");

mainMenuItem("menuadministrativo_b6",".gif",31,147,loc+"index.html", "", "SALIR",2,
2,"menuadministrativo_plain");

endMainMenu("menuadministrativo_right.gif",31,10)

startSubmenu("menuadministrativo_b5","menuadministrativo_menu",147);

submenuItem("CONSULTAR",loc+"notas1.php", "", "menuadministrativo_plain");

submenuItem("AÑADIR",loc+"notas2.php", "", "menuadministrativo_plain");

submenuItem("MODIFICAR",loc+"notas3.php", "", "menuadministrativo_plain");

endSubmenu("menuadministrativo_b5");

startSubmenu("menuadministrativo_b4","menuadministrativo_menu",147);

submenuItem("LISTAR",loc+"autoridades1.php", "", "menuadministrativo_plain");

submenuItem("AGREGAR",loc+"autoriddaes2.php", "", "menuadministrativo_plain");

submenuItem("MODIFICAR",loc+"autoridades3.php", "", "menuadministrativo_plain");

```



```
submenuItem("ELIMINAR",loc+"autoridades4.php","", "menuadministrativo_plain");
endSubmenu("menuadministrativo_b4");
```

```
startSubmenu("menuadministrativo_b3_2","menuadministrativo_menu",79);
submenuItem("LISTAR",loc+"actividades21.php","", "menuadministrativo_plain");
submenuItem("AGREGAR",loc+"actividades22.php","", "menuadministrativo_plain");
submenuItem("EDITAR",loc+"actividades23.php","", "menuadministrativo_plain");
submenuItem("ELIMINAR",loc+"actividades24.php","", "menuadministrativo_plain");
endSubmenu("menuadministrativo_b3_2");
```

```
startSubmenu("menuadministrativo_b3_1","menuadministrativo_menu",85);
submenuItem("LISTAR",loc+"actividades11.php","", "menuadministrativo_plain");
submenuItem("AÑADIR",loc+"actividades12.php","", "menuadministrativo_plain");
submenuItem("MODIFICAR",loc+"actividades13.php","", "menuadministrativo_plain")
;
submenuItem("ELIMINAR",loc+"actividades14.php","", "menuadministrativo_plain");
endSubmenu("menuadministrativo_b3_1");
```

```
startSubmenu("menuadministrativo_b3","menuadministrativo_menu",166);
mainMenuItem("menuadministrativo_b3_1","CURRICULARES",0,0,"javascript:;", "", "
",1,1,"menuadministrativo_1");
mainMenuItem("menuadministrativo_b3_2","EXTRACURRICULARES",0,0,"javascript
:;", "", "",1,1,"menuadministrativo_1");
endSubmenu("menuadministrativo_b3");
```

```
startSubmenu("menuadministrativo_b2","menuadministrativo_menu",147);
submenuItem("LISTAR",loc+"administra1.php","", "menuadministrativo_plain");
submenuItem("NUEVOS",loc+"administra2.php","", "menuadministrativo_plain");
submenuItem("EDITAR",loc+"administra.php","", "menuadministrativo_plain");
endSubmenu("menuadministrativo_b2");
```

```
startSubmenu("menuadministrativo_b1","menuadministrativo_menu",147);
submenuItem("LISTAR",loc+"docentes1.php","", "menuadministrativo_plain");
submenuItem("AÑADIR",loc+"docentes2.php","", "menuadministrativo_plain");
submenuItem("MODIFICAR",loc+"docentes3.php","", "menuadministrativo_plain");
submenuItem("ELIMINAR",loc+"docentes4.php","", "menuadministrativo_plain");
endSubmenu("menuadministrativo_b1");
```

```
loc="";
```

4.6. IMPLEMENTACION.-

Para un mejor funcionamiento de la aplicación Web la Escuela debería considerar la adquisición de un dominio como por ejemplo www.escuelmjcalle.edu.ec.

El hosting a contratar debe estar bajo el Sistema Operativo Linux, para poder disponer de los servicios como apache MySql y PHP.

4.7. MANUAL DE USUARIO.

A continuación están detalladas las pantallas del sistema, las cuales se pueden observar mediante nuestro local host.

Las opciones principales de la aplicación web son:

La escuela



Esta es la página inicial del portal, la presentación, en la cual podemos observar una breve historia y fotos de la escuela Manuel de J Calle, de esta misma ventana podemos acceder a los diferentes módulos, la cual nos permitirá ver las opciones que posee esta aplicación como son:

PAGINA INICIAL

Esta imagen es una breve muestra de la aportación que se realiza para la institución.

Introducción - Ubicación - Misión - Visión



Fig. #1

Mediante el servidor local, se logra el acceso a la página inicial, la misma que permite al usuario el acceso a la línea de banner, ya sean estos LA ESCUELA, PERSONAL, ACTIVIDADES, EVALUACIONES, ó CONTACTOS. De la misma manera permite al personal autorizado, ingresar a las opciones de ADMINISTRACION mediante un botón que se encuentra en la parte inferior-derecha de la Figura #1.

La escuela



Fig. #2

La Fig. #2 muestra mediante imágenes el entorno educativo evaluativo y recreativo, y a su vez una breve historia sobre la escuela Manuel de J Calle.

Personal



Fig. #3

Mediante el acceso a la pestaña del módulo PERSONAL, se puede observar las diferentes aplicaciones, tales como: AUTORIDADES, PERSONAL ADMINISTRATIVO Y PERSONAL DOCENTE. Los cuales nos mostraran el listado de los mismos, como lo indica la Fig. #3

Autoridades



Fig. #4

La Fig. #4 permite visualizar los diferentes campos que se deben rellenar con los datos específicos de las autoridades pertenecientes a la institución.

Personal Administrativo



Fig. #5

La Fig. #5 permite visualizar los diferentes campos que se deben rellenar con los datos específicos, correspondiente al personal administrativo de la institución.

Listado General de Docentes



Fig. #6

La Fig. #6 permite visualizar los datos específicos del listado general de docentes que pertenecen a la institución.

Actividades



Fig. #7

Las Fig. #7 permite elegir una de las opciones correspondientes a las actividades que realiza la institución.

Actividades Curriculares & Extracurriculares



Fig. #8



Fig. #9

La Fig. #8 & Fig. #9 permiten visualizar el listado de las actividades ya sean estas educativas, deportivas ó artísticas.

Evaluaciones



Fig. #10

La Fig. #10 nos permitirá ver el modelo de exámenes que los maestros tienen para evaluar a los niños de los primeros años de educación básica, los mismos que están separados por año escolar, con sus respectivos cuestionarios correspondientes a las nueve unidades evaluativas establecidas por la institución.

Ejemplo de Evaluación para niños del Primer Año Escolar

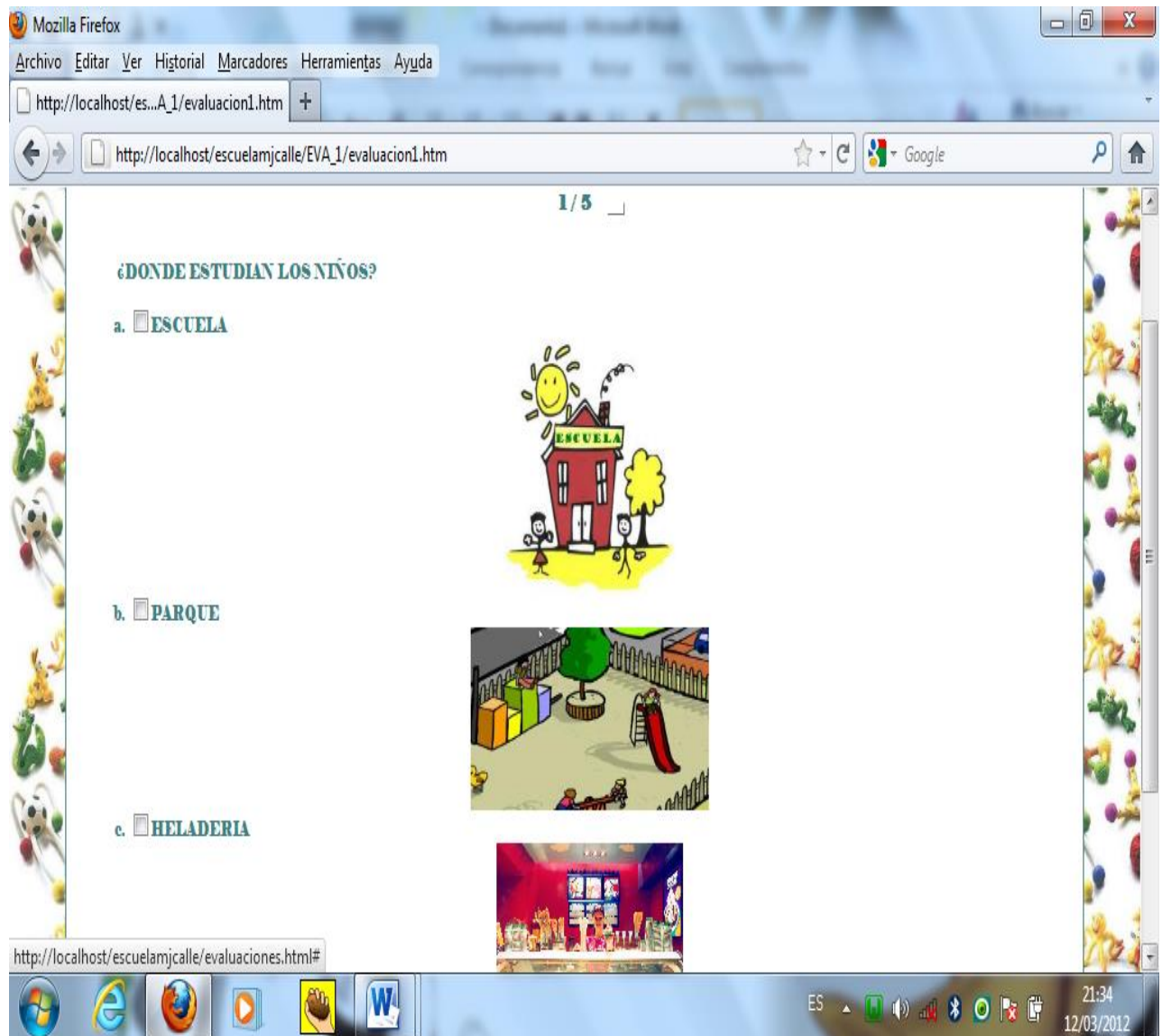


Fig. #11

Como se puede observar la Fig. #11 pertenece a un modelo de evaluación, para poder apreciar el nivel de conocimientos adquiridos en los niños de los dos primeros años de Educación Básica, mediante la culminación de cada unidad conformada por cuestionarios, según lo establecido por el libro escolar de la institución.

Contactos

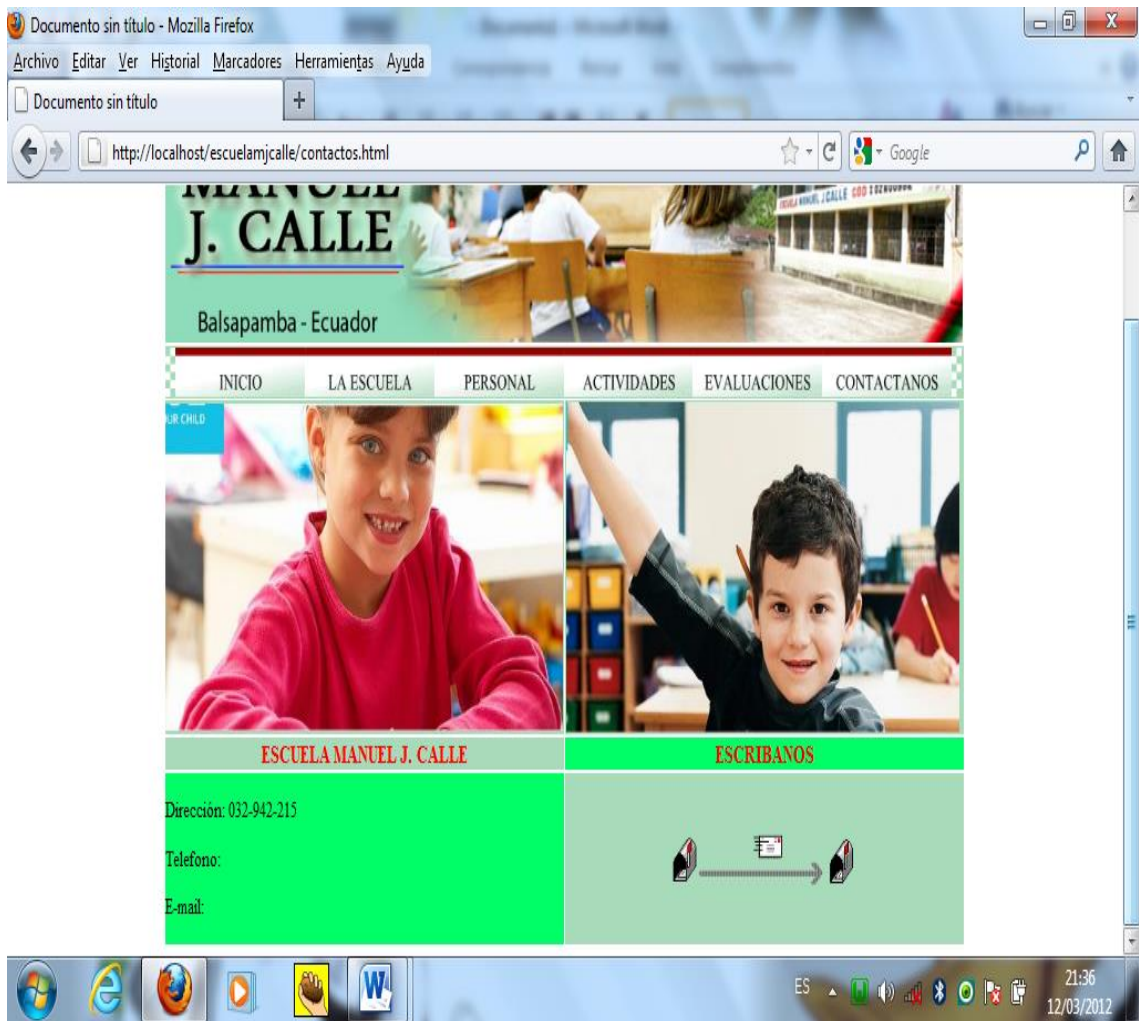


Fig. # 12

Esta Fig. #12 nos permite observar la forma de comunicación y obtener mayor información para/con la institución.

4.10.1 MENU DE USUARIO ADMINISTRATIVO



ESCUELA FISCAL MIXTA
MANUEL DE J. CALLE
Balsapamba - Ecuador

Escuela M. J. Calle

USUARIO

CLAVE

Ingresar Cancelar

INICIO

Fig. #13

En la Fig. #13 debemos resaltar que mediante este menú permitirá ingresar al modo administrativo, el cual aprueba el acceso mediante el ingreso del respectivo usuario y su contraseña.

MENU DE ADMINISTRACION



Fig. #14

La Fig. #14 permite visualizar la bienvenida al modo administrativo. La misma q nos permitirá el ingreso a cualquiera de las opciones como las siguientes:

MENU DE DOCENTES



Fig. #15

La Fig. #15 muestra el conjunto menús y submenús correspondientes a la pestaña de Docentes.

AÑADIR REGISTROS EN DOCENTES

The image shows a web application interface for adding new teacher records. The header is the same as in Fig. #15. Below the header, the title 'INGRESO DE NUEVOS DOCENTES' is displayed. The form contains the following fields and values:

Nombre:	Juara
Apellido:	Jaramillo
Direccion:	Babahoyo
Telefono:	052733754
Titulo 3 nivel:	Loda en Educacion Primaria
Titulo 4 nivel:	Ninguno
Fecha ingreso:	02-03-1990
Nota:	Ninguna

At the bottom of the form, there are two buttons: 'Insertar registro' and 'Cancelar'. Below the form, there is a yellow circular button with a black arrow pointing left.

Fig. #16

Esta fig. #16 permite de los datos correspondientes a los docentes



Fig. #17

La Fig. #17 muestra que los datos han sido ingresados de forma correcta.

MENU ADMINISTRATIVO

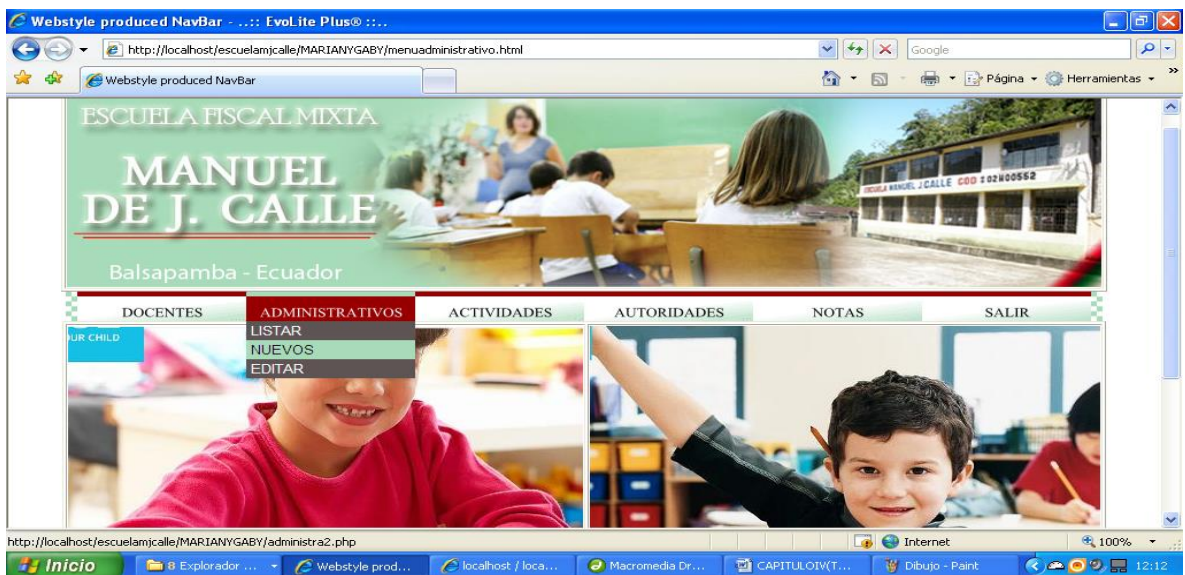
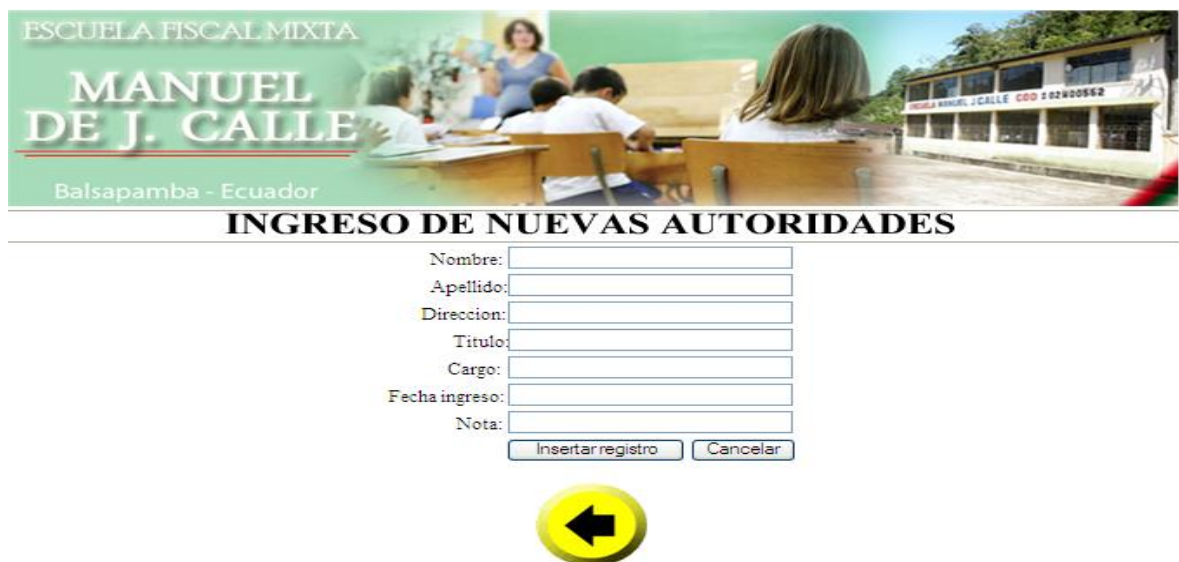


Fig. #18

Esta ventana nos muestra la Fig. #18 la cual permite visualizar el conjunto de menús y sus diferentes opciones dentro del menú administrativo.

INGRESAR REGISTROS EN ADMINISTRATIVOS (AUTORIDADES)



ESCUELA FISCAL MIXTA
MANUEL DE J. CALLE
Balsapamba - Ecuador

INGRESO DE NUEVAS AUTORIDADES

Nombre:
Apellido:
Direccion:
Titulo:
Cargo:
Fecha ingreso:
Nota:




Fig. #19

Esta fig. #19 permite el ingreso de los datos correspondientes a las autoridades del plantel.



ESCUELA FISCAL MIXTA
MANUEL DE J. CALLE
Balsapamba - Ecuador

DATOS GRABADOS CORRECTAMENTE



Fig. #20

La Fig. #20 muestra que los datos han sido ingresados de forma correcta.

MENU DE ACTIVIDADES



Fig. #21

Esta ventana nos muestra la Fig. #21 la cual permite visualizar el conjunto de menús y sus diferentes opciones dentro del menú Actividades.

INGRESAR ACTIVIDADES CURRICULARES

ESCUELA FISCAL MIXTA
MANUEL DE J. CALLE
Balsapamba - Ecuador

INGRESAR DE NUEVAS ACTIVIDADES CURRICULARES

Actividad:
Fecha:
Lugar:
Responsable:
Nota:

←

Fig. #22

Esta fig. #22 permite el ingreso de los datos correspondientes a las actividades curriculares del plantel.



Fig. #23

La Fig. #23 muestra que los datos han sido ingresados de forma correcta.

INGRESAR ACTIVIDADES EXTRACURRICULARES

The image shows a software interface for a school system. At the top, there is a header with the text "ESCUELA FISCAL MIXTA" and "MANUEL DE J. CALLE" in large, bold letters. Below this, it says "Balsapamba - Ecuador". The main content area features the text "INGRESO DE NUEVAS ACTIVIDADES EXTRACURRICULARES" in bold letters. Below this text is a form with the following fields: "Actividad:", "Fecha:", "Lugar:", "Responsable:", and "Nota:". Below the form are two buttons: "Insertar registro" and "Cancelar". Below the buttons is a yellow circular arrow icon pointing to the left.

Fig. #24

Esta Fig. #24 permite el ingreso de los datos correspondientes a las actividades extracurriculares del plantel.



Fig. # 25

La Fig. #25 muestra que los datos han sido ingresados de forma correcta.

MENU DE AUTORIDADES



Fig. #26

Esta ventana nos muestra la Fig. #26 la cual permite visualizar el conjunto de menús y sus diferentes opciones dentro del menú Autoridades.

INGRESO DE AUTORIDADES



ESCUELA FISCAL MIXTA
MANUEL DE J. CALLE
Balsapamba - Ecuador

INGRESO DE NUEVAS AUTORIDADES

Nombre:
Apellido:
Direccion:
Titulo:
Cargo:
Fecha ingreso:
Nota:



Fig. # 27

Esta fig. #27 permite el ingreso de los datos correspondientes a las autoridades del plantel.



ESCUELA FISCAL MIXTA
MANUEL DE J. CALLE
Balsapamba - Ecuador

DATOS GRABADOS CORRECTAMENTE



Fig. # 28

La Fig. #28 muestra que los datos han sido ingresados de forma correcta.

MENU DE NOTAS



Fig. #29

Esta ventana nos muestra la Fig. #29 la cual permite visualizar el conjunto de menús y sus diferentes opciones dentro del menú Notas.

4.11. REQUERIMIENTOS

Para el funcionamiento tenemos que tomar en cuenta los siguientes requerimientos:

4.11.1. Requerimientos de hardware:

- Servidor Intel Xeon Core Duo 1,7 o superior
- 2 Gb de Ram
- 1 Gb libre para instalación y 3 Gb libres para su correcto funcionamiento

4.11.2. Requerimientos de software:

- Servidor web apache Wamp Server v.2.0 o superior
- Mozilla Firefox 4.0 o superior (plugin de flash incluido)

4.11.3 Instalar WampServer

- Dar clic en el icono para empezar la instalación(ver Figura 4-9)

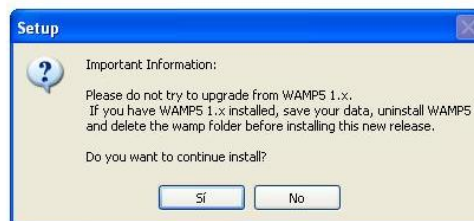


Figura 4-9: Ejecutable del wamp

- La instalación se basa en un asistente normal que nos solicitará varios datos típicos de instalaciones, como que aceptemos los términos de la licencia. Luego nos saldrá la ventana para acabar que marcaremos que ejecute Wamp Server
- (ver Figura 4-10) inmediatamente.



Figura 4-10: Instalación completada del wamp

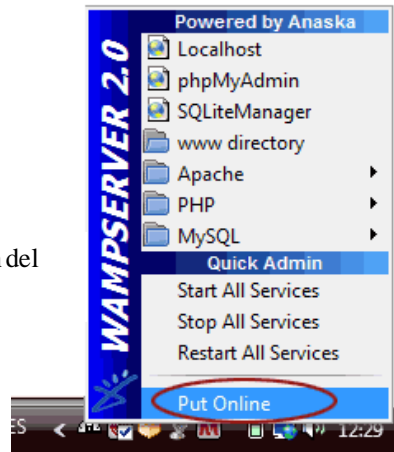
- Si todo ha funcionado, en 1 minuto mas podremos comprobar si PHP está funcionando en nuestro computador. Sólo tendríamos que encender los servicios. Para ello Wamp Server tiene un panel de control que se accede desde un icono de programa residente de la barra de tareas (ver figura 4-11):



Figura 4-11: Icono del wamp

- Pulsamos el icono con el ratón (botón izquierdo, clic normal) y veremos abajo del todo una instrucción que pone “Put Online”, que pondrá todos los servicios en funcionamiento. ver figura 4-12.

Figura 4-12: Ejecución del wamp



Ahora, para comprobar que los servicios funcionan sólo nos queda abrir un navegador. Vamos a escribir la siguiente dirección URL en la barra de direcciones:

`http://localhost`

Entonces nos tiene que salir la página de inicio del servidor Apache con PHP 2.0, personalizada por Wamp, que es algo como esto (ver Figura 4-13):

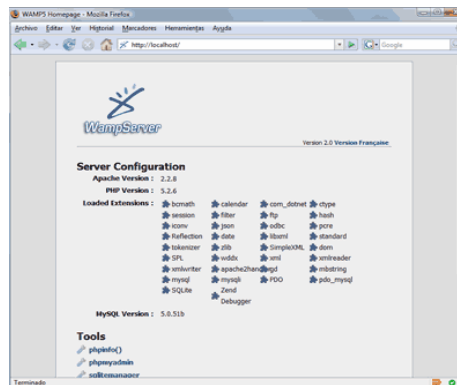


Figura 4-13: Pruebas del wamp

Si no sale nada puede que haya habido un problema o un error al iniciar los servicios, generalmente el Apache, que utiliza el puerto 80 que a veces está ocupado por otro programa.

Ahora podemos colocar en nuestro servidor todas las páginas PHP que tenemos. El directorio donde generalmente se localiza la raíz de publicación es: `C:\wamp\www` aquí ponemos nuestra carpeta denominada proyecto.

4.12. CONCLUSIONES

Como conclusiones del uso y desarrollo de la aplicación Web podemos señalar:

- Las aplicaciones Web son una estrategia fundamental dentro del proceso enseñanza aprendizaje del nivel escolar
- El proceso evaluativo mediante la aplicación web implica la capacitación del docente así como la dedicación de un tiempo adicional para el diseño de las evaluaciones.
- Las evaluaciones mediante la aplicación se han convertido en una experiencia agradable y no tan traumante como en otros periodos.
- Las herramientas del software libre son de muy fácil utilización y existe una gran cantidad de ellas en internet.

4.13. RECOMENDACIONES

Entre las cosas a recomendar señalaremos:

- Capacitar a los profesores en el manejo del programa hotpotatoes para la elaboración de los cuestionarios.

- Gestionar antes las autoridades gubernamentales la dotación del internet así como el número mayor de equipos.
- Concientizar al nivel de padres de familia sobre lo útil y provechoso que resulta el internet en el área educativa.

BIBLIOGRAFIA.

- SANCHEZ Segundo, “*Linux, guía práctica* “, Editorial Ra-ma, 2007, Madrid-España.
- GARCIA Rafael, “*El libro de Linux*”, Editorial Ra-ma, Madrid-España, 2007
- JONO Baton, JESSE Jhonatan, “*El libro oficial de Ubuntu*”, Prentice-Hall, Madrid-España, 2008.
- DEL PESO Emilio, PIATTINE Mario, “*Auditoría Informática, un enfoque práctico*”, Editorial Ra-ma, Madrid-España, Segunda edición, 2007
- GLABER Lothar, “*Open source, Software*”, McGraw-Hill, 2007
- THOMPSON Lara, “*Desarrollo Web con PHP y MySql*”, Editorial Agapea, Madrid-España, 2009. SANMARTI Neo, “*Ideas claves, Evaluar para aprender*”, Prentice Hall, México-México, 2007.
- GARCIA José, “*Bases pedagógicas de la evaluación*”, Prentice-Hall, México-México, 2007. GARCIA José, “*Evaluación en el primer ciclo infantil*”, Prentice-Hall, México-México, 2007.
- COGESHALL Jhon, “*PHP 5*”, Editorial Agapea, Madrid-España, 2009.