



UNIVERSIDAD NACIONAL  
DE  
CATAMARCA



FACULTAD DE TECNOLOGÍA Y CIENCIAS APLICADAS

LICENCIATURA EN  
SISTEMAS DE INFORMACIÓN

TRABAJO FINAL

**SISTEMA DE GESTIÓN  
ADMINISTRATIVA PARA LA ESCUELA  
SECUNDARIA N° 44  
“JORGE LUIS BORGES”**

Autores:

**GONZÁLEZ JORGE CARLOS M.U. N° 990**

**ROGER ETEL SILVINA M.U. N° 621**

Directora:

**MGTR. CAROLA VICTORIA FLORES**

**JULIO 2017**

## **AGRADECIMIENTOS**

A la Facultad de Tecnología y Ciencias Aplicadas dependiente de la Universidad Nacional de Catamarca, donde aprendimos a conocer esta profesión.

A nuestra directora Mgtr. Carola Victoria Flores por su tiempo dedicado, la claridad y la buena predisposición y apoyo al desarrollo de este trabajo.

A las autoridades de la Esc. N°44 “Jorge Luis Borges” por el compromiso que asumieron a lo largo de todo el proyecto.

En especial

A nuestros padres

A nuestras familias

A la memoria de nuestros abuelos

## TABLA DE CONTENIDOS

Agradecimientos .....	II
<b>TABLA DE CONTENIDOS</b> .....	<b>III</b>
<b>ÍNDICE DE FIGURAS</b> .....	<b>V</b>
<b>ÍNDICE DE TABLAS</b> .....	<b>V</b>
Resumen.....	VI
<b>INTRODUCCIÓN</b> .....	<b>7</b>
1 Capítulo I – Marco Teórico .....	8
1.1 Introducción .....	9
1.2 Metodologías Agiles .....	9
1.2.1 Manifiesto Agil.....	9
1.2.2 Metodologías Agiles vs. Metodologías Tradicionales .....	11
1.2.3 Metodologías Agiles existentes .....	12
1.2.4 Cuando usar Metodologías Agiles .....	17
1.2.5 Ventajas de las Metodologías Agiles .....	18
1.2.6 Desventajas de las Metodologías Agiles .....	18
1.2.7 ¿Quiénes utilizan las Metodologías Agiles y por qué las usan? .....	19
1.2.8 Uso de Métodos Agiles .....	19
1.3 Programación Extrema .....	21
1.3.1 Historia .....	21
1.3.2 La filosofía de XP .....	21
1.3.3 Valores de XP .....	21
1.3.4 Principios de XP .....	23
1.3.5 Reglas y Practicas .....	24
1.3.6 Comentarios respecto de las prácticas .....	33
1.3.7 Roles en XP .....	34
1.3.8 Proceso XP .....	34
1.3.9 Ciclo de vida .....	35
2 Capítulo II – Marco Metodológico .....	38
2.1 Introducción .....	39
2.2 Planteamiento del problema .....	39
2.3 Justificación .....	39
2.4 Objetivos .....	40
2.4.1 Objetivo General .....	40

2.4.2	Objetivos Específicos .....	40
2.5	Metodología de Trabajo .....	40
2.5.1	Análisis Exploratorio .....	40
2.5.2	Desarrollo del Sistema de Información .....	41
2.5.3	Análisis de Resultados y Elaboración de Conclusiones .....	41
2.5.4	Elaboración del Informe Final del Trabajo .....	41
3	Capítulo III- Desarrollo del Software .....	42
3.1	Introducción .....	43
3.2	Actividades en las Fases Desarrolladas.....	43
3.2.1	Requerimientos y Análisis .....	43
3.2.2	Diseño .....	43
3.2.3	Implementación.....	44
3.2.4	Pruebas.....	44
3.3	Fase de Exploración.....	44
3.3.1	Descripción de la organización.....	44
3.3.2	Historias de Usuario .....	46
3.3.3	Herramientas y Tecnologías que se usaron .....	46
3.4	Fase de Planificación .....	49
3.4.1	Tareas realizadas .....	49
3.4.2	Planificación de Prototipos .....	49
3.5	Fase de Iteración .....	50
3.5.1	DESARROLLO DEL PRIMER PROTOTIPO .....	50
3.5.2	DESARROLLO DEL SEGUNDO PROTOTIPO .....	68
3.5.3	DESARROLLO DEL TERCER PROTOTIPO.....	80
3.5.4	DESARROLLO DEL CUARTO PROTOTIPO .....	92
4	Capítulo IV - Análisis de Resultados y Conclusiones.....	108
4.1	Resultados Alcanzados .....	109
4.2	Conclusiones.....	110
	Referencias.....	112
	Bibliografía .....	113
	Anexos .....	114
	Anexo I: Estándares de programación .....	115
	Anexo II: Historias de Usuarios .....	117
	Anexo III: Documentación utilizada por la escuela .....	126

## ÍNDICE DE FIGURAS

Figura 1-1 Empresas que utilizan metodología ágil en sus proyectos [2].....	19
Figura 1-2 El uso de metodologías ágiles en las empresas [2].....	20
Figura 1-3 Valores de XP [7] .....	23
Figura 1-4 Costo Tiempo (enfoque tradicional vs. XP) [2].....	25
Figura 1-5 Las prácticas se refuerzan entre si [1].....	33
Figura 1-6 Ciclo de vida de XP [8].....	35
Figura 3-1– Esquema de la arquitectura / Tecnología del sistema .....	47

## ÍNDICE DE TABLAS

Tabla 1.1 Esencia del manifiesto ágil [2].....	11
Tabla 1.2 Diferencias entre metodologías ágiles y no ágiles [1] .....	12
Tabla 1.3 Metodologías ágiles [2].....	13
Tabla 3.1 Planificación de los Prototipos .....	49
Tabla 3.2 Planificación del Primer Prototipo .....	50
Tabla 3.3 Nuevos Requisitos .....	67
Tabla 3.4 Planificación del Segundo Prototipo.....	68
Tabla 3.5 Planificación del Tercer Prototipo .....	80
Tabla 3.6 Nuevos Requisitos .....	91
Tabla 3.7 Planificación del Cuarto Prototipo.....	92
Tabla 3.8 Nuevos Requisitos .....	105

## RESUMEN

La Metodología Ágil *Programación Extrema* (Extreme Programming -XP) está orientada a proyectos pequeños o medianos, con plazos reducidos, requisitos volátiles y nuevas tecnologías. Valora el trabajo cooperativo entre el equipo de desarrollo y el cliente, al desarrollo incremental del software con iteraciones muy cortas, responde a los cambios de último momento, el método en si es sencillo, fácil de aprender y modificar.

El presente trabajo trata sobre el desarrollo de un Sistema de Gestión Administrativa para la Escuela Secundaria N° 44 “Jorge Luis Borges”, aplicando la Metodología Ágil *Programación Extrema*, se considera la más adecuada para este proyecto, ya que el contexto es muy cambiante debido a la implementación de la nueva ley de educación, en donde se exige reducir drásticamente los tiempos de desarrollo manteniendo una alta calidad.

El objetivo principal del sistema es optimizar la gestión administrativa, centralizar la información de la institución, agilizar los trámites escolares, evitar la carga manual que origina errores de registración de datos, brindar a las autoridades herramientas de control sobre el manejo de la gestión administrativa y la posibilidad de contar con información clara, precisa, confiable en el momento y lugar oportuno para el planeamiento y toma de decisiones. El sistema fue implementado utilizando tecnología web, por lo que se puede acceder a él a través de Internet en forma sencilla y rápida.



---

## INTRODUCCIÓN

En la Provincia de Catamarca y más precisamente en ámbitos de la educación pública y privada del nivel secundario se observa que no existe un sistema de gestión administrativa que pueda ser utilizado desde la Web, sí lo tiene la Universidad Nacional de Catamarca por medio del sistema SIU-GUARANI que mejora la gestión de la institución, permitiéndole contar con información segura, íntegra y disponible, optimizando así sus recursos, por lo que se decide realizar un sistema para una escuela pública de la provincia, la escuela secundaria N°44 “Jorge Luis Borges” de Miraflores, departamento Capayán de la Provincia de Catamarca, en la que se observa los siguientes problemas: el incremento significativo de la matrícula y del personal de la escuela, lo que produce un aumento en la cantidad de la documentación que se recibe, analiza, organiza, registra y emite, además se llevan a cabo las tareas en forma manual, lo que origina errores de registración de datos. Otro problema es la duplicación de la información, ya que se registra varias veces en distintos documentos, por una o varias personas y por último el acceso a la información no es inmediata, debido a que está dispersa en distintos documentos.

Para dar solución a esta problemática se desarrolló un Sistema de Gestión Administrativa, utilizando la Metodología Ágil XP, por ser la más adecuada en contextos muy cambiantes. Los objetivos del sistema son: crear una base de datos para registrar, actualizar y consultar la información; centralizar la información para eliminar la duplicación de datos; automatizar las tareas para disminuir la carga de trabajo respecto a los controles que se realizan en forma manual y mejorar el acceso a los datos para obtener información oportuna para la toma de decisiones.

El sistema está diseñado para ser utilizado por la comunidad educativa de la escuela permitiendo realizar: la inscripción de los alumnos y admisión si cumple los requisitos, el seguimiento de la trayectoria escolar de los alumnos, y ciclos lectivos según la modalidad de estos, centralización de la información de docentes, la inscripción a examen y generación de actas de examen, la emisión de recibos, certificados, planillas y libretas.

Este trabajo está organizado en capítulos cuyo contenido es el siguiente: En el Capítulo I se introducen las principales características de las metodologías ágiles, recogidas en el Manifiesto, se hace una revisión de los principales métodos ágiles y se centra en *eXtreme Programming*(XP), presentando sus características particulares, como son las historias de usuario y los roles, así como el proceso que se sigue y las prácticas que propone. El Capítulo II, hace referencia al Marco Metodológico, planteando el problema detectado en la escuela secundaria N° 44 “Jorge Luis Borges”, la justificación por lo cual se desarrolla el sistema, los objetivos que se propusieron y la metodología de trabajo empleada. En el Capítulo III se describe como fueron implementadas las fases de la metodología Programación XP en el sistema. Y finalmente en el Capítulo IV, el análisis de resultados y conclusiones.

# CAPÍTULO I

---

## **Marco Teórico**

## 1.1 INTRODUCCIÓN

Este capítulo aborda el marco teórico en el que se basa el trabajo, el cual surgió del análisis bibliográfico y del contexto de la investigación. Se expone sobre que son las metodologías ágiles, los distintos tipos con sus características principales, cuando usarlas, sus ventajas y desventajas, quienes la utilizan y por qué la eligen. También se describe con mayor profundidad la metodología XP que es la usada para el desarrollo del sistema, comenzando con su historia y filosofía, siguiendo con sus valores, principios, prácticas y la descripción de los roles de acuerdo con la propuesta original de Kent Beck, al finalizarse expone el ciclo de vida de XP además de las reglas y prácticas de la metodología.

## 1.2 METODOLOGÍAS AGILES

Nace el término “ágil” aplicado al desarrollo de software, en una reunión realizada en Uta Estados Unidos en el mes de Febrero del año 2001, participaron un grupo de expertos de la industria del software, incluyendo algunos creadores o impulsores de metodologías de software como XP, SCRUM, DSDM, Adaptive Software Development, Crystal, Feature-Driven Development, y Programatic Programming. El objetivo fue idear los valores y principios que deberían permitir desarrollar software rápidamente, respondiendo a los cambios que puedan surgir, y ofrecer otra alternativa a los procesos de desarrollo de software tradicionales, caracterizados por ser rígidos y dirigidos por la documentación que se genera en cada una de las actividades desarrolladas. Así se creó la organización *The Agile Alliance*, dedicada a promover los conceptos relacionados con el desarrollo ágil de software y ayudar a las organizaciones para que adopten dichos conceptos [1].

### 1.2.1 Manifiesto Agil

El Manifiesto comienza enumerando los principales valores del desarrollo ágil. Donde se valora [1]:

- **Al individuo y las interacciones del equipo de desarrollo sobre el proceso y las herramientas.** El principal factor del éxito de un proyecto de software, es la gente. Si el equipo falla, aunque se siga un buen proceso de desarrollo, el éxito no está asegurado; sin embargo, si el equipo funciona, es más fácil conseguir el objetivo final, aunque no se tenga un proceso bien definido. Se necesitan desarrolladores que se adapten bien al trabajo en equipo, a pesar de que no sean brillantes. Así mismo, las herramientas (compiladores, depuradores, IDEs de desarrollo, etc.) son importantes para mejorar el rendimiento del equipo, pero el disponer más recursos que los estrictamente necesarios también pueden afectar negativamente.

En resumen, es más importante construir un buen equipo que construir el entorno de trabajo perfecto.

- **Desarrollar software que funciona más que conseguir una buena documentación.** Se parte de la base de que el software sin documentación es un desastre, la regla a seguir es “no producir documentos a menos que sean necesarios de forma inmediata para tomar una decisión importante”. Estos documentos deben ser cortos y centrarse en lo fundamental.

- **La colaboración con el cliente más que la negociación de un contrato.** Muchos proyectos han fracasado por intentar cumplir plazos y costos preestablecidos al inicio del mismo, según los requisitos que el cliente manifestaba en ese momento. Por ello, se propone que exista una interacción constante entre el cliente y el equipo de desarrollo. Esta colaboración entre ambos será la que marque la marcha del proyecto y asegure su éxito.
- **Responder a los cambios más que seguir estrictamente un plan.** La habilidad de responder a los cambios que puedan surgir a lo largo del proyecto (cambios en los requisitos, en la tecnología, en el equipo, etc.) determina también el éxito o fracaso del mismo. Por lo tanto, la planificación no debe ser estricta puesto que hay muchas variables en juego, debe ser flexible para poder adaptarse a los cambios que puedan surgir.

Los valores anteriores inspiran los doce principios del manifiesto, estos **principios** son las características que diferencian un proceso ágil de uno tradicional. Los dos primeros son generales y resumen gran parte del espíritu ágil:

**I. La prioridad es satisfacer al cliente mediante tempranas y continuas entregas de software que le aporte un valor.** Un proceso es ágil si a las pocas semanas de empezar ya se entrega software que funcione aunque sea rudimentario. El cliente decide si pone en marcha dicho software con la funcionalidad que ahora le proporciona o simplemente lo revisa e informa de posibles cambios a realizar.

**II. Dar la bienvenida a los cambios. Se capturan los cambios para que el cliente tenga una ventaja competitiva.** Los cambios en los requisitos deben verse como algo positivo. Les va a permitir aprender más, a la vez que logran una mayor satisfacción del cliente. Este principio implica además que la estructura del software debe ser flexible para poder incorporar los cambios sin demasiado costo añadido.

Luego existen una serie de principios que tienen que ver directamente con el proceso de desarrollo de software a seguir.

**III. Entregar frecuentemente software que funcione desde un par de semanas a un par de meses, con el menor intervalo de tiempo posible entre entregas.** Las entregas al cliente se insisten en que sean software, no planificaciones, ni documentación de análisis o de diseño.

**IV. La gente del negocio y los desarrolladores deben trabajar juntos a lo largo del proyecto.** El proceso de desarrollo necesita ser guiado por el cliente, por lo que la interacción con el equipo es muy frecuente.

**V. Construir el proyecto en torno a individuos motivados.** Darles el entorno y el apoyo que necesitan y confiar en ellos para conseguir finalizar el trabajo. La gente es el principal factor de éxito, todo lo demás (proceso, entorno, gestión, etc.) queda en segundo plano. Si cualquiera de ellos tiene un efecto negativo sobre los individuos debe ser cambiado.

**VI. El diálogo cara a cara es el método más eficiente y efectivo para comunicar información dentro de un equipo de desarrollo.** Los miembros de equipo deben hablar entre ellos, éste es el principal modo de comunicación. Se pueden crear documentos pero no todo estará en ellos, no es lo que el equipo espera.

**VII. El software que funciona es la medida principal de progreso.** El estado de un proyecto no viene dado por la documentación generada o la fase en la que se encuentre,

sino por el código generado y en funcionamiento. Por ejemplo, un proyecto se encuentra al 50% si el 50% de los requisitos ya están en funcionamiento.

**VIII. Los procesos ágiles promueven un desarrollo sostenible. Los promotores, desarrolladores y usuarios deberían ser capaces de mantener una paz constante.** No se trata de desarrollar lo más rápido posible, sino de mantener el ritmo de desarrollo durante toda la duración del proyecto, asegurando en todo momento que la calidad de lo producido es máxima.

Finalmente los últimos principios están más directamente relacionados con el equipo de desarrollo, en cuanto metas a seguir y organización del mismo.

**IX. La atención continua a la calidad técnica y al buen diseño mejora la agilidad.**

Producir código claro y robusto es la clave para avanzar más rápidamente en el proyecto.

**X. La simplicidad es esencial.** Tomar los caminos más simples que sean consistentes con los objetivos perseguidos. Si el código producido es simple y de alta calidad será más sencillo adaptarlo a los cambios que puedan surgir.

**XI. Las mejores arquitecturas, requisitos y diseños surgen de los equipos organizados por sí mismos.** Todo el equipo es informado de las responsabilidades y éstas recaen sobre todos sus miembros. Es el propio equipo el que decide la mejor forma de organizarse, de acuerdo a los objetivos que se persigan.

**XII. En intervalos regulares, el equipo reflexiona respecto a cómo llegar a ser más efectivo, y según esto ajusta su comportamiento.** Puesto que el entorno está cambiando continuamente, el equipo también debe ajustarse al nuevo escenario de forma continua. Puede cambiar su organización, sus reglas, sus convenciones, sus relaciones, etc., para seguir siendo ágil.

Los firmantes de los valores y principios de este Manifiesto son: Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland y Dave Thomas.

En la tabla 1.1 se puede ver la esencia del manifiesto ágil.

DESCONFIAMOS	PREFERIMOS
Los procesos y las herramientas	A las personas y su comunicación
La documentación exhaustiva	El software que funciona
La negociación contractual	La colaboración con el cliente
Seguimiento de un plan	La respuesta al cambio

Tabla 1.1 Esencia del manifiesto ágil [2]

### 1.2.2 Metodologías Ágiles vs. Metodologías Tradicionales

Antes de resumir algunas metodologías ágiles, se enumeran las principales diferencias respecto de las metodologías tradicionales (“no ágiles”). La Tabla 1.2 recoge esquemáticamente estas diferencias que no se refieren sólo al proceso en sí, sino también

al contexto de equipo y organización que es más favorable a cada uno de estas filosofías de procesos de desarrollo de software.

<b>METODOLOGÍAS AGILES</b>	<b>METODOLOGÍAS TRADICIONALES</b>
Basadas en heurísticas provenientes de prácticas de producción de código	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo
Especialmente preparados para cambios durante el proyecto	Cierta resistencia a los cambios
Impuestas internamente (por el equipo de desarrollo)	Impuestas externamente
Proceso menos controlado, con pocos Principios	Proceso mucho más controlado, con numerosas políticas/normas
No existe contrato tradicional o al menos es bastante flexible	Existe un contrato prefijado
El cliente es parte del equipo de Desarrollo	El cliente interactúa con el equipo de desarrollo mediante reuniones
Grupos pequeños (<10 integrantes) y trabajando en el mismo sitio	Grupos grandes y posiblemente Distribuidos
Pocos artefactos	Más artefactos
Pocos roles	Más roles
Menos énfasis en la arquitectura del Software	La arquitectura del software es esencial y se expresa mediante modelos

**Tabla 1.2 Diferencias entre metodologías ágiles y no ágiles [1]**

### 1.2.3 Metodologías Ágiles existentes

Se cuentan varias metodologías ágiles sin contar los métodos híbridos de desarrollo que integran en sus prácticas como ejemplo a XP con Scrum. A continuación en la Tabla 1.3 el listado de metodologías ágiles en desarrollo de software más representativas con el año de creación, acrónimo y autor.

<b>METODOLOGÍA</b>	<b>ACRONIMO</b>	<b>CREACIÓN</b>
SCRUM	SCRUM	Sutherland 1994 Schwaber 1995
CRYSTAL METHODOLOGIES (Metodologías Cristal)	CM	Cockburn 1998
DYNAMIC SYSTEMS DEVELOPMENT METHOD (Método de Desarrollo de Sistema Dinámico)	DSDM	Stapleton 1994
ADAPTIVE SOFTWARE DEVELOPMENT (Desarrollo de Software Adaptable)	ASD	Highsmith 2000
FEATURE-DRIVEN DEVELOPMENT (Desarrollo Manejado por Rasgos)	FDD	De Luca &Coad 1998 Palmer &Felsing 2002
LEAN DEVELOPMENT (Desarrollo Magro)	LD	Charette 2001 Mary Tom Poppendieck
RATIONAL UNIFIED PROCESS (Proceso Racional Unificado)	RUP	Kruchten 1996
AGILE RUP (Ágil RUP)	dX	Booch Martin Newkirk 1998
EXTREME PROGRAMMING (Programación Extrema)	XP	Beck 1999
AGILE MODELING	AM	Ambler 2002
EVOLUTIONARY PROYECT MANAGEMENT	Evo	Gilb 1976
MICROSOFT SOLUTIONS FRAMEWORK	MSF	Microsoft 1994
RAPID DEVELOPMENT	RAD	Mcconnell1996

**Tabla 1.3 Metodologías ágiles [2]**

Metodologías que se basan en procesos rápidos, pensados en ser funcionales, adaptables y además manejan similitudes entre ellos y pueden complementarse o formar métodos híbridos.

Aunque los creadores e impulsores de las metodologías ágiles más populares han suscrito el manifiesto ágil y coinciden con los principios enunciados anteriormente, cada metodología tiene características propias y hace hincapié en algunos aspectos más específicos. A continuación se resumen dichas metodologías ágiles.

### 1.2.3.1 SCRUM

Define un marco para la gestión de proyectos, que se ha utilizado con éxito durante los últimos 10 años. Está especialmente indicada para proyectos con un rápido cambio de requisitos.

Sus principales características se pueden resumir en dos:

- El desarrollo de software se realiza mediante iteraciones, denominadas *sprints*, con una duración de 30 días. El resultado de cada *sprint* es un incremento ejecutable que se muestra al cliente.
- La segunda característica importante son las reuniones a lo largo del proyecto, estas son las verdaderas protagonistas, especialmente la reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración [1].

### 1.2.3.2 Crystal Methodologies (Metodologías Cristal)

Se trata de un conjunto de metodologías para el desarrollo de software caracterizadas por estar centradas en las personas que componen el equipo (de ellas depende el éxito del proyecto) y la reducción al máximo del número de artefactos producidos. El desarrollo de software se considera un juego cooperativo de invención y comunicación, limitado por los recursos a utilizar. El equipo de desarrollo es un factor clave, por lo que se deben invertir esfuerzos en mejorar sus habilidades y destrezas, así como tener políticas de trabajo en equipo definidas. Estas políticas dependerán del tamaño del equipo, estableciéndose una clasificación por colores, por ejemplo Crystal Clear considera de 3 a 8 miembros y Crystal Orange considera de 25 a 50 miembros [1].

### 1.2.3.3 Dynamic Systems Development Method (Método de Desarrollo de Sistema Dinámico-DSDM)

Tiene una organización de tiempo completo que lo apoya con manuales, cursos de entrenamiento, programas de certificación y demás. También lleva una etiqueta de precio, lo que limita su investigación.

Sus principales características son: es un proceso iterativo e incremental y el equipo de desarrollo y el usuario trabajan juntos.

Propone cinco fases:

- Estudio viabilidad: considera si es apropiado para el proyecto.
- Estudio de negocio: es una serie corta de talleres para entender el área de negocio dónde tiene lugar el desarrollo. También propone arquitecturas de esbozos del sistema y un plan del proyecto.
- Modelado funcional: produce documentación de análisis y prototipos.
- Diseño y construcción: diseña el sistema para uso operacional.
- Implementación: se ocupa del despliegue al uso operacional.

Las tres últimas son iterativas, además de existir realimentación a todas las fases.

#### 1.2.3.4 Adaptive Software Development (Desarrollo de Software Adaptable - ASD)

Su impulsor es Jim Highsmith ha pasado muchos años trabajando con metodologías predictivas; él las desarrolló, instaló, enseñó, y concluyó que son profundamente defectuosas: particularmente para los negocios modernos.

Su reciente libro se enfoca en la naturaleza adaptable de las nuevas metodologías, con un énfasis particular en aplicar las ideas que se originaron en el mundo de los sistemas complejos adaptables (normalmente conocida como teoría del caos). No proporciona el tipo de prácticas detalladas como lo hace XP, pero proporciona la base fundamental de por qué el desarrollo adaptable es importante y las consecuencias a los más profundos niveles de la organización y la gerencia. Sus principales características son: iterativo, orientado a los componentes software más que a las tareas y tolerante a los cambios.

El ciclo de vida que propone tiene tres fases esenciales: especulación, colaboración y aprendizaje. En la primera de ellas se inicia el proyecto y se planifican las características del software; en la segunda desarrollan las características y finalmente en la tercera se revisa su calidad, y se entrega al cliente. La revisión de los componentes sirve para aprender de los errores y volver a iniciar el ciclo de desarrollo.

#### 1.2.3.5 Feature-DrivenDevelopment (Desarrollo Manejado por Rasgos - FDD)

FDD fue desarrollado por Jeff De Luca y el viejo gurú de la OO Peter Coad. Como las otras metodologías adaptables, se enfoca en iteraciones cortas que entregan funcionalidad tangible.

Define un proceso iterativo que consta de 5 procesos.

- Desarrollar un Modelo Global
- Construir una Lista de los Rasgos
- Planear por Rasgo
- Diseñar por Rasgo
- Construir por Rasgo

Los primeros tres procesos se hacen al principio del proyecto y los últimos dos se hacen en cada iteración. Cada proceso se divide en tareas y se da un criterio de comprobación. Las iteraciones son cortas (hasta 2 semanas). Se centra en las fases de diseño e implementación del sistema partiendo de una lista de características que debe reunir el software.

Los desarrolladores entran en dos tipos: dueños de clases y programadores jefe. Los programadores jefe son los desarrolladores más experimentados. A ellos se les asignan rasgos a construir. Sin embargo ellos no los construyen solos. Solo identifican qué clases se involucran en la implantación de un rasgo y juntan a los dueños de dichas clases para que formen un equipo para desarrollar ese rasgo. El programador jefe actúa como el coordinador, diseñador líder y mentor mientras los dueños de clases hacen gran parte de la codificación del rasgo.

### 1.2.3.6 Lean Development (Desarrollo Magro - LD)

Definida por Bob Charette's a partir de su experiencia en proyectos con la industria japonesa del automóvil en los años 80 y utilizada en numerosos proyectos de telecomunicaciones en Europa. En LD, los cambios se consideran riesgos, pero si se manejan adecuadamente se pueden convertir en oportunidades que mejoren la productividad del cliente. Su principal característica es introducir un mecanismo para implementar dichos cambios.

Últimamente LD ha evolucionado como Lean Software Development (LSD); LD se inspira en doce valores centrados en estrategias de gestión:

- Satisfacer al cliente es la máxima prioridad.
- Proporcionar siempre el mejor valor por la inversión.
- El éxito depende de la activa participación del cliente.
- Cada proyecto LD es un esfuerzo de equipo.
- Todo se puede cambiar.
- Soluciones de dominio, no puntos.
- Completar, no construir.
- Una solución al 80% hoy, en vez de una al 100% mañana.
- El minimalismo es esencial.
- La necesidad determina la tecnología.
- El crecimiento del producto es el incremento de sus prestaciones, no de su tamaño.
- Nunca empujes LD más allá de sus límites.

LD y LSD han sido pensados como complemento de otros métodos como XP o Scrum y no como una metodología excluyente a implementar en la empresa.

### 1.2.3.7 Rational Unified Process (Proceso Racional Unificado -RUP)

Es un proceso de ingeniería de software que suministra un enfoque para asignar tareas y responsabilidades dentro de una organización de desarrollo. Su objetivo es asegurar la producción de software de alta calidad que satisfaga la necesidad del usuario final dentro de un tiempo y presupuesto previsible.

Es una metodología de desarrollo iterativo enfocada hacia “los casos de uso, manejo de riesgos y el manejo de la arquitectura”.

El RUP mejora la productividad del equipo ya que permite que cada miembro del grupo sin importar su responsabilidad específica acceda a la misma base de datos de conocimiento. Esto hace que todos compartan el mismo lenguaje, la misma visión y el mismo proceso acerca de cómo desarrollar software.

En el ciclo de vida RUP veremos una implementación del desarrollo en espiral. Con el ciclo de vida se establecen tareas en fases e iteraciones.

El RUP maneja el proceso en cuatro fases, dentro de las cuales se realizan varias iteraciones. A continuación se enumeran las fases:

- Fase de inicio: Durante esta fase de inicio las iteraciones se centran con mayor énfasis en las actividades de modelamiento de la empresa y en sus requerimientos
- Fase de elaboración: Durante esta fase de elaboración, las iteraciones se centran al desarrollo de la base del diseño, encierran más los flujos de trabajo de requerimientos, modelo de la organización, análisis, diseño y una parte de implementación orientada a la base de la construcción.
- Fase de construcción: Durante esta fase de construcción, se lleva a cabo la construcción del producto por medio de una serie de iteraciones las cuales se seleccionan algunos casos de uso, se redefine su análisis y diseño y se procede a su implantación y pruebas. En esta fase se realiza una pequeña cascada para cada ciclo, se realizan tantas iteraciones hasta que se termine la nueva implementación del producto.
- Fase de transición: Durante esta fase de transición busca garantizar que se tiene un producto preparado para su entrega al usuario.

Agile RUP (Ágil RUP – DX) es una versión totalmente dócil del RUP que simplemente es idéntico a la XP. Está diseñado para gente que tiene que usar el RUP pero quiere usar XP.

### **1.2.3.8 Extreme Programming (Programación Extrema – XP)**

Es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo.

XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico.

Los principios y prácticas son de sentido común pero llevadas al extremo, de ahí proviene su nombre. En el punto 1.3 se expone esta metodología en forma más exhaustiva.

### **1.2.4 Cuando usar Metodologías Ágiles**

No existe una metodología universal para hacer frente con éxito a cualquier proyecto de desarrollo de software. Toda metodología debe ser adaptada al contexto del proyecto (recursos técnicos y humanos, tiempo de desarrollo, tipo de sistema).

Históricamente, las metodologías tradicionales han intentado abordar la mayor cantidad de situaciones de contexto del proyecto, exigiendo un esfuerzo considerable para ser adaptadas, sobre todo en proyectos pequeños y con requisitos muy cambiantes.

Las metodologías ágiles ofrecen una solución casi adecuada para una gran cantidad de proyectos. Sin embargo existen métodos más generales y con mejores resultados que otros; saber qué reglas y metodologías aplicar en cada caso es más importante y útil que seguir ciegamente siempre las mismas [4].

Las propias prácticas de los métodos ágiles limitan o descartan su uso en algunos proyectos. A continuación se detallan algunos casos donde no conviene usar métodos ágiles.

- Aplicaciones distribuidas. Las pruebas unitarias son complicadas de aplicar entre componentes. Sería necesario construir una arquitectura de pruebas para probar directamente los componentes, que podría ser tan complicada como el sistema que se desea construir.
- Aplicaciones que requieren seguir un diseño estricto. Por ejemplo: sistemas operativos, software de telecomunicaciones.
- Aplicaciones que requieren una documentación exhaustiva. Por ejemplo: sistemas militares, médicos o industriales.
- Aplicaciones basadas fundamentalmente en interfaces gráficas de usuario: No es fácil aplicar pruebas unitarias a las interfaces gráficas.
- Aplicaciones con código heredado. Habría que reescribir todo el código heredado siguiendo los principios ágiles.
- Proyectos muy grandes. La comunicación entre los miembros del equipo es difícil de conseguir.
- Aplicaciones donde la escalabilidad o la eficacia sean importantes. La escalabilidad o la eficacia no son características que se pueden añadir durante el proceso del desarrollo del software o que puedan obtenerse refactorizando.

### 1.2.5 Ventajas de las Metodologías Ágiles

Las metodologías ágiles presentan diversas ventajas como [4]:

- Rápida respuesta a cambios de requisitos a lo largo del desarrollo.
- Entrega continua y en plazos cortos de software funcional.
- Trabajo conjunto entre el cliente y el equipo de desarrollo.
- Minimiza los costos frente a cambios.
- Importancia de la simplicidad, al eliminar el trabajo innecesario.
- Atención continua a la excelencia técnica y al buen diseño.
- Mejora continua de los procesos y el equipo de desarrollo.
- Evita malentendidos de requerimientos entre el cliente y el equipo.
- El equipo de desarrollo no malgasta el tiempo y dinero del cliente desarrollando soluciones innecesariamente generales y complejas que en realidad no son un requisito del cliente.
- Cada componente del producto final ha sido probado y satisface los requerimientos.

### 1.2.6 Desventajas de las Metodologías Ágiles

Como en cualquiera otra metodología, también hay desventajas y problemas que surgen a la hora de implementarlas [4]:

- Falta de documentación del diseño. El código no puede tomarse como una documentación. En sistemas de tamaño grande se necesita leer los cientos o miles de páginas del listado de código fuente.
- Problemas derivados de la comunicación oral. Este tipo de comunicación resulta difícil de preservar cuando pasa el tiempo y está sujeta a muchas ambigüedades.

- Fuerte dependencia de las personas. Como se evita en lo posible la documentación y los diseños convencionales, los proyectos ágiles dependen críticamente de las personas.
- Falta de reusabilidad. La falta de documentación hacen difícil que pueda reutilizarse el código ágil.
- Sobre costos y retrasos derivados de la refactorización continua. Para un sistema de ciertas proporciones, los costos y retrasos derivados de la refactorización no pueden despreciarse.
- Restricciones en cuanto a tamaño de los proyectos abordables.
- Cambios. Los modelos de datos son “pesados” y no pueden cambiarse así como así solo porque el cliente quiera incorporar más funciones al sistema.
- Problemas derivados del fracaso de los proyectos ágiles. Si un proyecto ágil fracasa no hay documentación o hay muy poca; lo mismo ocurre con el diseño. La comprensión del sistema se queda en las mentes de los desarrolladores.

### 1.2.7 ¿Quiénes utilizan las Metodologías Ágiles y por qué las usan?

Los métodos ligeros o ágiles son otra opción para el desarrollo de sistemas, además presenta adeptos y gurús en contra, algunos expertos mencionan que los procesos ágiles son una moda y quedaran ahí, sin embargo existen empresas que desde hace tiempo utilizan y han evolucionado gracias a dichos métodos.

En la Figura 1.1 se muestra las empresas que utilizan estas metodologías en sus proyectos [2].



Figura 1-1 Empresas que utilizan metodología ágil en sus proyectos [2]

### 1.2.8 Uso de Métodos Ágiles

Desde el surgimiento de estas revolucionarias metodologías que no solo nacen para el desarrollo de sistemas software sino para el desarrollo de productos, los incrementos en adeptos se presentan gradualmente con el tiempo y las tecnologías, esto lo podemos ver en la Figura 1-2.



Figura 1-2 El uso de metodologías ágiles en las empresas [2]

Las razones porque usan estas metodologías son:

- Para reducir el tiempo de desarrollo: 45%
- Para mejorar la calidad: 43%
- Para reducir costos: 23%
- Para alinear el desarrollo con los objetivos de negocio: 39%
- Otras razones: 12%.

Todas estas razones tienen un único objetivo común: **reducir el presupuesto** que se pueda ofrecer al cliente sin bajar los parámetros clave de la calidad del software.

- **eficiencia**: la cantidad de recursos de tiempo y espacio que el software economiza.
- **fiabilidad**: el software hace lo que debería hacer, y no "truenas".
- **usabilidad**: es fácil de usar.
- **evaluable**: es sencillo de examinar. Se pueden correr fácilmente las pruebas unitarias.
- **Mantenibilidad y escalabilidad**: pueden agregarse funciones de manera rápida y consistente.
- **portabilidad**: la portabilidad implica que un programa puede correr en distintas plataformas, no sólo de sistemas operativos sino diferentes versiones, diferentes ambientes y esquemas o bases de datos.
- **seguridad**: es el software seguro contra ataques externos.
- **compatibilidad**: es el software amigable con otros sistemas.

El ranking de preferencias entre modelos ágiles [2]

1º.- Extreme Programming (28%).

2º.- FDD (26%).

3º.- Scrum (20%).

4º.- Crystal (6%) ágil.

5º.- DSDM (4%).

## 1.3 PROGRAMACIÓN EXTREMA

### 1.3.1 Historia

La Programación Extrema, nace de la mano de Kent Beck (gurú de la XP y autor de los libros más influyentes sobre el tema).

**Chrysler Corporation** hacía tiempo que estaba desarrollando una aplicación de nóminas, pero sin demasiado éxito por parte de la gente que tenía en el proyecto. El verano de 1996, Beck entró en nómina en la compañía y se le pidió hacer esta aplicación como trabajo. Es en esta aplicación cuando nace la Programación Extrema como tal.

Beck reconoció que el proceso (o metodología) de creación de software o la carencia de este era la causa de todos los problemas y llegó a la conclusión que para proporcionar un proceso que fuera flexible era necesario realizar ciertos cambios en la estructura o manera de hacer de los programadores, los cuales se tenían que acomodar al cambio a realizar. Él tenía varias ideas de metodologías para la realización de programas que eran cruciales para el buen desarrollo de cualquier sistema.

Las ideas primordiales de su sistema las comunicó en la revista **C++ Magazine** en una entrevista que ésta le hizo el año 1999. En ésta decía que él estaba convencido que la mejor metodología era un proceso que enfatizase la comunicación dentro del equipo, que la implementación fuera sencilla, que el usuario tenía que estar muy informado e implicado y que la toma de decisiones tenía que ser muy rápida y efectiva.

Los propulsores de la Programación Extrema, empezaron a hablar de ella, promocionarla en cada ocasión que tenían y difundirla en las páginas de programación. [5]

### 1.3.2 La filosofía de XP

La filosofía XP consiste en satisfacer por completo las necesidades del cliente, por eso lo integra como parte del equipo de desarrollo. XP es una metodología ágil, que se la puede aplicar para el desarrollo de software pequeño o mediano, cuando los requerimientos son ambiguos o cambiantes, en XP se realiza el software que el cliente solicita y necesita, en el momento que lo precisa, alentando a los programadores a responder a los requerimientos cambiantes que plantea el cliente en cualquier momento. Esto es posible porque está diseñado para adaptarse inmediatamente a los cambios, con bajos costos, en cualquier etapa del ciclo de vida. En pocas palabras, XP “abrazo” el cambio.

El ciclo de vida de un proyecto XP es muy dinámico, se puede separar en fases (Exploración, Planificación de la Entrega, Iteraciones, Producción, Mantenimiento y Muerte del Proyecto, que se verán más adelante) y varios de los detalles acerca de las tareas de éstas fases se detallan, en la sección 1.3.5. “Reglas y Practicas”.

### 1.3.3 Valores de XP

Para poder implementar las prácticas que caracterizan a la metodología XP, hay que conocer cuáles son sus principales valores para que el proyecto tenga éxito ellos son [7]:

## **Comunicación**

Es uno de los valores más importantes y es por eso que el cliente es integrado al equipo, de esta forma, cualquier duda sobre los requerimientos puede ser evacuada inmediatamente y este puede estar al tanto del avance del proyecto.

XP ha sido diseñada para minimizar el grado de documentación como forma de comunicación, haciendo énfasis en la interacción personal. De esta manera se puede avanzar rápidamente y de forma efectiva, realizando solo la documentación necesaria.

## **Simplicidad**

XP propone una regla muy simple: “hacer algo que funcione de la manera más sencilla”. En el caso de tener que añadir nueva funcionalidad al sistema se deben examinar todas las posibles alternativas y seleccionar la más sencilla. En otras ocasiones se hace uso de la refactorización que permite mantener el código en funcionamiento pero mucho más simple y organizado.

Otra regla muy importante es “realizar solo lo necesario”, con esto se pretende agregar nueva funcionalidad que cumpla con los objetivos actuales sin necesidad de preocuparse por futuros requerimientos. Esto hace que se progrese de manera más segura y rápida en el proyecto.

XP apuesta a realizar algo simple hoy y destinar un poco más de esfuerzo para realizar un cambio en el futuro, que a realizar algo más complicado hoy y no utilizarlo nunca.

## **Retroalimentación (Feedback)**

La retroalimentación debe funcionar en forma permanente. Brindar una retroalimentación correcta y precisa hace que se pueda mantener una buena comunicación y esta le ofrece al cliente la posibilidad de conseguir un sistema apto a sus necesidades, ya que se le va mostrando el proyecto a tiempo para poder ser cambiado, retrocediendo a una fase anterior para rediseñarlo a su gusto.

Los resultados de las pruebas unitarias son también una retroalimentación permanente que tienen los desarrolladores acerca de la calidad de su trabajo.

La retroalimentación trabaja a diferentes escalas de tiempo. Uno es la retroalimentación que se realiza minuto a minuto. Cuando un cliente escribe sus historias, los programadores realizan la estimación de cada una de ellas y el cliente puede obtener inmediatamente la retroalimentación sobre la calidad de dichas historias.

El otro tipo de retroalimentación que se realiza es a través de pequeñas entregas del sistema. De esta manera, el cliente está al tanto del avance del proyecto. Además, el sistema es puesto en producción en menor tiempo, con lo cual los programadores saben si realizaron un buen trabajo y si sus decisiones fueron acertadas.

## **Coraje**

Cada uno de los valores antes mencionados tiene una gran interacción entre ellos. Como se ve en la Figura 1-3 la comunicación, la simplicidad y la retroalimentación forman el coraje, el cual se convierte en el objetivo de XP. Esto hace que se tenga el coraje de modificar el

código en cualquier momento por cualquier miembro del equipo sabiendo que no se afectará el correcto funcionamiento del sistema, realizar cambios cuando algo no funciona del todo bien, diseñar e implementar solo lo necesario para el presente, pedir ayuda o reducir el alcance de una entrega si el tiempo no alcanza.

Si no hay coraje en un proyecto no se puede clasificar como extremo y es necesario que los otros tres valores estén presentes.

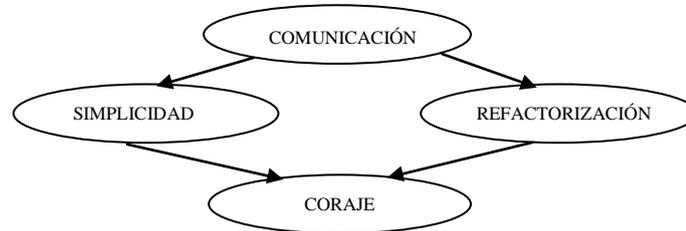


Figura 1-3 Valores de XP [7]

#### 1.3.4 Principios de XP

Los cuatro valores mencionados anteriormente – comunicación, simplicidad, retroalimentación y coraje –brindan un estándar para obtener buenos resultados. Sin embargo, los valores son muy vagos a la hora de ayudarnos a decidir que prácticas utilizar. Para ello se necesita separar estos valores en principios que puedan ser utilizados.

A continuación se detallan los principios más importantes de XP [7].

- **Rápida retroalimentación.** El tiempo transcurrido entre una acción y su retroalimentación es crítico en la práctica, tener una rápida retroalimentación nos permite: interpretarla, aprender de ella y poner en práctica lo asimilado lo antes posible.
- **Asumir la simplicidad.** Es el más difícil de llevar a la práctica, casi siempre se planifica para el futuro y se diseña para poder rehusar, en lugar de esto XP dice que hay que hacer un buen trabajo para las necesidades actuales y confiar en nuestra habilidad para solucionar problemas futuros.
- **Cambios incrementales.** Realizar grandes cambios en una sola oportunidad no es una buena solución, cada problema debe ser resuelto con una serie de cambios pequeños para poder resolverlo.
- **Aceptar el cambio.** En XP el cambio es asimilado como algo habitual e inevitable, la mejor estrategia es aquella que preserva la mayor cantidad de opciones mientras resuelve los problemas más precisos.
- **Trabajo de calidad.** Uno de los objetivos más importantes en XP es realizar un producto de buena calidad, Si cada integrante realiza su trabajo de la mejor manera posible se puede asegurar la calidad del producto.

## Otros principios

Además de los principios antes mencionados surgen algunos otros de menor trascendencia que pueden ayudar a tomar buenas decisiones en situaciones específicas.

- **Aceptar la responsabilidad:** Es bueno permitir que cada uno de los integrantes se ofrezca de forma voluntaria para poder realizar una tarea determinada y que este se comprometa ante el resto del equipo. Imponer una tarea hace que el miembro del equipo haga algo que pueda no cumplir más adelante y pierda su motivación.
- **Adaptación local:** Para que XP funcione de la mejor manera posible es necesario que se adapte a las condiciones particulares de cada proyecto.
- **Comunicación abierta y honesta:** La comunicación es uno de los pilares fundamentales en XP. Es necesario que cada uno de los integrantes sepa exponer las consecuencias de sus propias decisiones y de las tomadas por los demás, así como también, comunicar los problemas sin miedo de anunciar malas noticias.
- **Enseñar conocimientos:** XP se concentra en enseñar distintas estrategias para saber cómo adoptar esta metodología sin tratar de imponer ninguna de estas.
- **Experimentos concretos:** Toda decisión que se tome a lo largo del proyecto debe ser analizada y probada para evitar así cualquier tipo de errores.
- **Jugar para ganar:** Todos los integrantes deben trabajar en equipo de manera agresiva y con mucho coraje para lograr el objetivo común que es el éxito de todo el proyecto.
- **Mediciones honestas:** Toda estimación o métrica realizada debe ser lo más lógica posible ya que no tiene sentido dar algo tan exacto que quizás nunca se cumpla.
- **Pequeña inversión inicial:** Al igual que la realización de un sistema en pequeñas entregas, es bueno realizar inversiones de igual manera e ir avanzando de forma progresiva.
- **Trabajar con los instintos de las personas:** XP está a favor de trabajar con los instintos de las personas y no en contra de ellos. Lo que incentiva los instintos de las personas es hacer bien su trabajo y realizar las tareas en grupo, interactuando con el resto del equipo.
- **Viajar liviano:** Para poder adaptarse rápidamente a los cambios constantes es bueno llevar solo lo indispensable.

### 1.3.5 Reglas y Practicas

En la (Figura 1-4), se observa que el enfoque tradicional con respecto al tiempo incrementa los costos del desarrollo del sistema.

Mientras que en la Programación Extrema al manejar las prácticas, que se describen a continuación, con pocos artefactos disminuye el tiempo de desarrollo e implementación obteniendo un producto conocido por el usuario y a su medida.

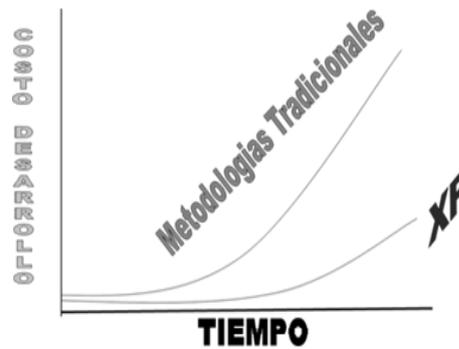


Figura 1-4 Costo Tiempo (enfoque tradicional vs. XP) [2]

La metodología XP tiene un conjunto importante de reglas y prácticas. En forma genérica, se pueden agrupar en [6]:

- Reglas y prácticas para la Planificación
- Reglas y prácticas para el Diseño
- Reglas y prácticas para el Desarrollo
- Reglas y prácticas para las Pruebas

#### 1.3.5.1 Planificación

La metodología XP plantea la planificación como un dialogo continuo entre las partes involucradas en el proyecto, incluyendo al cliente, a los programadores y a los coordinadores o gerentes. El proyecto comienza recopilando “historias de usuarios”, las que sustituyen a los tradicionales “casos de uso”. Una vez obtenidas las “historias de usuarios”, los programadores evalúan rápidamente el tiempo de desarrollo de cada una. Si alguna de ellas tiene “riesgos” que no permiten establecer con certeza la complejidad del desarrollo, se realizan pequeños programas de prueba, para reducir estos riesgos. Una vez realizadas estas estimaciones, se organiza una reunión de planificación, con los diversos actores del proyecto (cliente, desarrolladores, gerentes), a los efectos de establecer un plan o cronograma de entregas en los que todos estén de acuerdo. Una vez acordado este cronograma, comienza una fase de iteraciones, en dónde en cada una de ellas se desarrolla, prueba e instala unas pocas “historias de usuarios”.

Según Martín Fowler (uno de los firmantes del “Manifiesto Ágil”), los planes en XP se diferencian de las metodologías tradicionales en tres aspectos:

- Simplicidad del plan. No se espera que un plan requiera de un “gurú” con complicados sistemas de gerenciamiento de proyectos.
- Los planes son realizados por las mismas personas que realizarán el trabajo.
- Los planes no son predicciones del futuro, sino simplemente la mejor estimación de cómo saldrán las cosas. Los planes son útiles, pero necesitan ser cambiados cuando las circunstancias lo requieren. De otra manera, se termina en situaciones en las que el plan y la realidad no coinciden, y en estos casos, el plan es totalmente inútil.

Los conceptos básicos de esta planificación son los siguientes:

**Historias de usuarios:** Se trata de tarjetas de papel en las cuales el cliente escribe en su propio lenguaje, descripciones cortas de lo que el sistema debe realizar y la que facilitará que el cliente pueda especificar la importancia relativa entre las diferentes historias de usuario, así como el trabajo de los programadores que podrán catalogarlas correctamente. Este formato también es muy útil en el momento de las pruebas de aceptación.

El tratamiento de las historias de usuario es muy dinámico y flexible, en cualquier momento historias de usuario pueden romperse, reemplazarse por otras más específicas o generales, añadirse nuevas o ser modificadas.

Las historias de usuario deben tener el detalle mínimo como para que los programadores puedan realizar una estimación poco riesgosa del tiempo que llevará su desarrollo. Cuando llegue el momento de la implementación, los desarrolladores dialogarán directamente con el cliente para obtener todos los detalles necesarios.

Respecto de la información contenida en la historia de usuario, existen varias plantillas sugeridas pero no existe un consenso al respecto. En muchos casos sólo se propone utilizar un nombre y una descripción o sólo una descripción, o una estimación de esfuerzo en días. Beck en su libro [2] presenta un ejemplo de ficha en la cual pueden reconocerse los siguientes contenidos: fecha, tipo de actividad (nueva, corrección, mejora), prueba funcional, número de historia, prioridad técnica y del cliente, referencia a otra historia previa, riesgo, estimación técnica, descripción, notas y una lista de seguimiento con la fecha, estado de cosas por terminar y comentarios.

El equipo técnico será el encargado de catalogar las historias del cliente y asignarles una duración. La norma es que cada historia de usuario tiene que poder ser realizable en un espacio entre una y cuatro semanas de programación. Las que requieran menos tiempo serán agrupadas, y las que necesiten más serán modificadas o divididas.

**Entregas pequeñas:** Producir rápidamente versiones del sistema que sean operativas, aunque no cuenten con toda la funcionalidad del sistema. Esta versión ya constituye un resultado de valor para el negocio. Una entrega no debería tardar más 3 meses. El poder tener un producto terminado cada cierto tiempo hace que el cliente vea como va tomando forma su producto y además hace que los programadores puedan retroalimentarse de las opiniones de los clientes y mantengan su nivel de motivación ya que lo más agradable de programar es poder entregar un producto terminado.

**Plan de entrega:** Después de tener ya definidas las historias de usuario es necesario crear un plan de entrega, este plan será el resultado de una reunión entre todos los actores del proyecto (cliente, desarrolladores, gerentes, etc.). XP denomina a esta reunión “El Juego de planificación”, pero puede denominarse de la manera que sea más apropiada al tipo de empresa y cliente (por ejemplo, Reunión de planificación), donde se establece qué historias de usuario serán agrupadas para conformar una entrega, el orden de las mismas y las fechas en las que se implementaran.

Después de un plan de entrega tienen que estar claros estos cuatro factores: los objetivos que se deben cumplir (que son principalmente las historias que se deben desarrollar en cada entrega), el tiempo que tardarán en desarrollarse y publicarse las versiones del

programa, el número de personas que trabajarán en el desarrollo y cómo se evaluará la calidad del trabajo realizado

**Iteraciones:** En la metodología XP, la creación del sistema se divide en etapas para facilitar su realización. Por lo general, los proyectos constan de más de tres etapas, las cuales toman el nombre de iteraciones, de allí se obtiene el concepto de metodología iterativa. La duración ideal de una iteración es de una a tres semanas. Para cada iteración se define un módulo o conjuntos de historias que se van a implementar.

**Plan de iteraciones:** Las historias de usuarios seleccionadas para cada entrega son desarrolladas y probadas en un ciclo de iteración, de acuerdo al orden preestablecido.

Al comienzo de cada ciclo, se realiza una reunión de planificación de la iteración.

Cada historia de usuario se traduce en tareas específicas que son asignadas a los programadores y son ellos los que estiman el tiempo necesario para desarrollarlas. Las tareas que no se realicen en una iteración son tomadas en cuenta para la próxima iteración, donde se definen, junto al cliente, si se deben realizar o si deben ser removidas de la planeación del sistema. Asimismo, para cada historia de usuario se establecen las pruebas de aceptación.

Estas pruebas se realizan al final del ciclo en el que se desarrollan, pero también al final de cada uno de los ciclos siguientes, para verificar que subsiguientes iteraciones no han afectado a las anteriores.

Las pruebas de aceptación que hayan fallado en el ciclo anterior son analizadas para evaluar su corrección, así como para prever que no vuelvan a ocurrir.

**Reuniones diarias de seguimiento:** El objetivo de tener reuniones diarias es mantener la comunicación entre el equipo, y compartir problemas, e ideas que se estén presentando, con el fin que el equipo en conjunto construya una mejor solución. Estas reuniones se realizan a comenzar la jornada laboral, tiene que ser fluida y todo el mundo tiene que tener voz y voto.

Es de vital importancia evitar las discusiones largas, ya que se está desperdiciando tiempo laboral que puede ser utilizado a la construcción del sistema, como también las conversaciones separadas, las dudas que se presenten serán solucionadas por el equipo en conjunto.

**Velocidad del proyecto:** La velocidad del proyecto es una medida de la capacidad que tiene el equipo de desarrollo para producir las historias de usuario en una determinada iteración, esta medida se calcula totalizando el número de historias de usuarios realizadas en una iteración. Para la iteración siguiente se podrá (teóricamente) implementar el mismo número de historias de usuarios que la iteración anterior.

**Roles XP:** En esta metodología se utiliza el concepto de roles para organizar quienes se encargan de cada una de las actividades que deben realizarse en el transcurso del proyecto. Cada uno de estos papeles son desempeñados por uno o varios integrantes del grupo, sin descartar la posibilidad de rotar roles entre el equipo durante la realización del sistema.

**Trabajo en equipo:** Es deseable cambiar a los desarrolladores de tareas en el proyecto, de tal forma que hayan realizado tareas en la mayor parte posible de módulos. Esta estrategia favorece la unión del equipo de desarrollo, y el aprendizaje. El objetivo es disminuir las dependencias necesarias del proyecto (nunca debe haber una sola persona que sepa hacer algo).

**Ajuste XP:** Todos los proyectos tienen características específicas por lo cual XP puede ser modificado para ajustarse bien al proyecto en cuestión. Al iniciar el proyecto se debe aplicar XP tal como es, sin embargo no se debe dudar en modificar aquellos aspectos en que no funcione. Eso no quiere decir que los desarrolladores pueden hacer lo que se le antoje. Antes de implementarse un cambio, este debe ser discutido y aprobado por el grupo.

### 1.3.5.2 Diseño

La metodología XP hace especial énfasis en los diseños simples y claros. Los conceptos más importantes de diseño en esta metodología son los siguientes:

#### Diseño simple

En XP se hace mucho énfasis en el diseño de una solución lo más simple posible que pueda ser implementada en el momento. El código complejo o innecesario es eliminado inmediatamente. Se sugiere nunca adelantar la implementación de funcionalidades que no correspondan a la iteración en la que se esté trabajando.

El diseño se hace en forma progresiva lo cual evita que se cree un diseño altamente complejo por querer abarcar todos los aspectos posibles de una sola vez. Un buen diseño debe cumplir los siguientes puntos:

- Corre todos los casos de prueba
- Enuncia todas las ideas que se quieren exhibir.
- No tiene código duplicado
- Posee el menor número de clases y métodos posibles.

Para poder obtener un diseño simple se descartan todos aquellos elementos innecesarios mientras que los tres primeros puntos se cumplan.

#### Soluciones “spike”

Cuando aparecen problemas técnicos, o cuando es difícil de estimar el tiempo para implementar una historia de usuario, pueden utilizarse pequeños programas de prueba (llamados “spike”), para explorar diferentes soluciones. Estos programas son únicamente para probar o evaluar una solución, y suelen ser desechados luego de su evaluación.

#### Refactorización

Consiste en escribir nuevamente parte del código de un programa, sin cambiar su funcionalidad, a los efectos de hacerlo más simple, conciso y/o entendible. Muchas veces, al terminar de escribir un código de programa, pensamos que, si lo comenzáramos de nuevo, lo hubiéramos hecho en forma diferente, más clara y eficientemente. Sin embargo, como ya está pronto y “funciona”, rara vez es reescrito. Las metodologías de XP sugieren recodificar cada vez que sea necesario. Si bien, puede parecer una pérdida de tiempo innecesaria en el plazo inmediato, los resultados de esta práctica tienen sus frutos en las siguientes iteraciones, cuando sea necesario ampliar o cambiar la funcionalidad. La filosofía que se

persigue es, como ya se mencionó, tratar de mantener el código más simple posible que implemente la funcionalidad deseada.

### **Metáfora**

Una “metáfora” es algo que todos entienden, XP sugiere utilizar este concepto como una manera sencilla de explicar el propósito del proyecto, y guiar la estructura y arquitectura del mismo. Por ejemplo, puede ser una guía para la nomenclatura de los métodos y las clases utilizadas en el diseño del código.

Tener nombres claros, que no requieran de mayores explicaciones, redundará en un ahorro de tiempo.

Es muy importante que el cliente y el grupo de desarrolladores estén de acuerdo y compartan esta “metáfora”, para que puedan dialogar en un “mismo idioma”. Una buena metáfora debe ser fácil de comprender para el cliente y a su vez debe tener suficiente contenido como para que sirva de guía a la arquitectura del proyecto.

Las tarjetas CRC (Clase, Responsabilidad y Colaboración) ayudan al equipo a definir actividades durante el diseño del sistema. Cada tarjeta representa una clase en la programación orientada a objetos y define sus responsabilidades (lo que ha de hacer) y las colaboraciones con las otras clases (cómo se comunica con ellas).

### **No solucionar antes de tiempo**

Los desarrolladores tienden a predecir las necesidades futuras e implementarlas antes.

Según mediciones, esta es una práctica ineficiente, concluyendo que tan solo el 10% de las soluciones para el futuro son utilizadas, desperdiciando tiempo de desarrollo y complicando el diseño innecesariamente.

En XP solo se analiza lo que se desarrollara en la iteración actual olvidando por completo cualquier necesidad que se puede presentar en el futuro lo que supone uno de los preceptos más radicales de la programación extrema.

### **1.3.5.3 Desarrollo del código**

#### **Cliente en el sitio**

El cliente tiene que estar presente y disponible durante todo el proyecto, no solamente como apoyo a los desarrolladores, sino formando parte del equipo. Éste es uno de los principales factores de éxito del proyecto XP. El cliente conduce constantemente el trabajo hacia lo que aportará mayor valor de negocio y los programadores pueden resolver de manera inmediata cualquier duda asociada.

Al comienzo del proyecto, el cliente debe proporcionar las historias de usuarios.

Pero, dado que estas historias son expresamente cortas y de “alto nivel”, no contienen los detalles necesarios para realizar el desarrollo del código. Estos detalles deben ser proporcionados por el cliente, y discutidos con los desarrolladores, durante la etapa de desarrollo. No se requieren de largos documentos de especificaciones, sino que los detalles son proporcionados por el cliente, en el momento adecuado, “cara a cara” a los desarrolladores.

Si bien esto parece demandar del cliente recursos por un tiempo prolongado, debe tenerse en cuenta que en otras metodologías este tiempo es insumido por el cliente en realizar los documentos detallados de especificación.

Adicionalmente, al estar el cliente en todo el proceso, puede prevenir a tiempo de situaciones no deseables, o de funcionamientos que no eran los que en realidad se deseaban. En otras metodologías, estas situaciones son detectadas en forma muy tardía del ciclo de desarrollo, y su corrección puede llegar a ser muy complicada.

Algunas recomendaciones propuestas para dicha situación son las siguientes: intentar conseguir un representante que pueda estar siempre disponible y que actúe como interlocutor del cliente, contar con el cliente al menos en las reuniones de planificación, establecer visitas frecuentes de los programadores al cliente para validar el sistema, anticiparse a los problemas asociados estableciendo llamadas telefónicas frecuentes y conferencias, reforzando el compromiso de trabajo en equipo.

### **Estándares de codificación**

XP enfatiza que la comunicación de los programadores es a través del código, con lo cual es indispensable que se sigan ciertos estándares de programación para mantener el código legible y que facilite la recodificación.

En un equipo de desarrollo los programadores acuerdan un estándar para el código, dejando de lado estilos de programación particulares. Todos deben conocer y seguir el estándar de manera tal que al final el sistema parezca ser programado por una sola persona.

Algunos de los puntos a tener en cuenta son los siguientes:

- Mantener métodos pequeños (el código puede ser visto en una sola pantalla)
- Ser coherentes en el uso de mayúsculas.
- Solo comentar el código cuando sea realmente necesario.
- Usar formatos de nombres consistentes y similares.
- Utilizar siempre el mismo formato de sangría.

### **Programación dirigida por las pruebas**

En las metodologías tradicionales, la fase de pruebas, incluyendo la definición de los tests, es usualmente realizada sobre el final del proyecto, o sobre el final del desarrollo de cada módulo. La metodología XP propone un modelo inverso, en el que, lo primero que se escribe son los test que el sistema debe pasar. Luego, el desarrollo debe ser el mínimo necesario para pasar las pruebas previamente definidas.

Las pruebas a que se refiere esta práctica, son las pruebas unitarias, realizados por los desarrolladores. La definición de estos test al comienzo, condiciona o “dirige” el desarrollo.

### **Programación en parejas**

Toda la producción de código debe realizarse con trabajo en parejas de programadores.

La programación en parejas tiene las siguientes ventajas:

- La mayoría de los errores se descubren en el momento en que se codifican, ya que el código es permanentemente revisado por dos personas.
- La cantidad de defectos encontrados en las pruebas es estadísticamente menor.
- Los diseños son mejores y el código más corto.
- El equipo resuelve problemas en forma más rápida.
- Las personas aprenden significativamente más, acerca del sistema y acerca de desarrollo de software.
- El proyecto termina con más personas que conocen los detalles de cada parte del código.
- Las personas aprenden a trabajar juntas, generando mejor dinámica de grupo y haciendo que la información fluya rápidamente.
- Las personas disfrutan más de su trabajo.

Dichos beneficios se consiguen después de varios meses de practicar la programación en parejas (de 3 a 4 meses).

### **Integración continúa**

Cada pieza de código es integrada en el sistema una vez que esté lista, así, el sistema puede llegar a ser integrado y construido varias veces en un mismo día. Para evitar errores, solo una pareja de desarrolladores puede integrar su código a la vez.

Todas las pruebas son ejecutadas y tienen que ser aprobadas para que el nuevo código sea incorporado definitivamente. La integración continua a menudo reduce la fragmentación de los esfuerzos de los desarrolladores por falta de comunicación sobre lo que puede ser reutilizado o compartido.

Lo bueno de este sistema es que no ocurren problemas de integración ya que los errores se encuentran y se solucionan al momento de ser integrados por los mismos programadores que tienen en la cabeza el código implementado. Esto evita que se pasen meses para la integración y tener que solucionar estos problemas más adelante.

### **Propiedad colectiva del código**

En un proyecto XP, todo el equipo puede contribuir con nuevas ideas que apliquen a cualquier parte del proyecto. Asimismo, cualquier pareja de programadores puede cambiar el código que sea necesario para corregir problemas, agregar funciones o recodificar.

En otras metodologías, este concepto puede parecer extraño. Muchas veces se asume que, si hay algo de propiedad colectiva, la responsabilidad también es colectiva. Y que “todos sean responsables”, muchas veces significa que “nadie es responsable”, este razonamiento no es correcto cuando se trabaja con la metodología de XP. En este caso, quienes encuentran un problema, o necesitan desarrollar una nueva función, pueden resolverlo directamente, sin necesidad de “negociar” con el “dueño” o autor del módulo (ya que, de hecho, este concepto no existe en XP).

Muchas veces, una solución pasa por la recodificación de varios módulos, que atraviesan de forma horizontal una determinada jerarquía vertical. Si es necesario dialogar y convencer al encargado de cada módulo, posiblemente la solución no se pueda implementar, por lo menos en tiempos razonables. En XP, se promueve la recodificación, en áreas de generar códigos más simples y adaptados a las realidades cambiantes. Cualquier pareja de programadores puede tomar la responsabilidad de este cambio. Las pruebas permanentes deberían de asegurar que los cambios realizados cumplan con lo requerido, y además, no afecten al resto de las funcionalidades.

#### **40 horas semanales.**

Se debe trabajar un máximo de 40 horas por semana, no se trabajan horas extras en dos semanas seguidas, si esto ocurre, probablemente está ocurriendo un problema que debe corregirse, el trabajo extra desmotiva al equipo e impacta en la calidad del producto.

Los proyectos que requieren trabajo extra para intentar cumplir con los plazos suelen al final ser entregados con retraso. En lugar de esto se puede realizar el juego de la planificación para cambiar el ámbito del proyecto o la fecha de entrega.

En la medida de lo posible, se debería renegociar el plan de entregas, realizando una nueva reunión de planificación con el cliente, los desarrolladores y los gerentes. Adicionalmente, agregar más desarrolladores en proyectos ya avanzados no siempre resuelve el problema.

El concepto que se desea establecer con esta práctica es el de planificar el trabajo de manera de mantener un ritmo constante y razonable, sin sobrecargar al equipo.

#### **1.3.5.4 Pruebas**

Otro de los puntos importantes de XP son las pruebas, los cuales se realizan continuamente. Existen dos tipos de pruebas; pruebas unitarias y pruebas funcionales o de aceptación.[6]

##### **Pruebas unitarias**

Son una de las piedras angulares de XP. Todos los módulos deben de pasar las pruebas unitarias antes de ser liberados o publicados, las pruebas deben ser definidas antes de realizar el código. Que todo código liberado pase correctamente las pruebas unitarias es lo que habilita que funcione la propiedad colectiva del código. En este sentido, el sistema y el conjunto de pruebas debe ser guardado junto con el código, para que pueda ser utilizado por otros desarrolladores, en caso de tener que corregir, cambiar o recodificar parte del mismo.

##### **Detección y corrección de errores**

Cuando se encuentra un error, éste debe ser corregido inmediatamente, y se deben tener precauciones para que errores similares no vuelvan a ocurrir.

Asimismo, se generan nuevas pruebas para verificar que el error haya sido resuelto.

### Pruebas de aceptación

Son creadas en base a las historias de usuarios, en cada ciclo de la iteración del desarrollo. El cliente debe especificar uno o diversos escenarios para comprobar que una historia de usuario ha sido correctamente implementada.

Las pruebas de aceptación son consideradas como “pruebas de caja negra” (donde se definirán las entradas al sistema y los resultados esperados de estas entradas). Los clientes son responsables de verificar que los resultados de estas pruebas sean correctos. Asimismo, en caso de que fallen varias pruebas, deben indicar el orden de prioridad de resolución.

Una historia de usuario no se puede considerar terminada hasta tanto pase correctamente todas las pruebas de aceptación.

Dado que la responsabilidad es grupal, es recomendable publicar los resultados de las pruebas de aceptación, de manera que todo el equipo esté al tanto de esta información.

#### 1.3.6 Comentarios respecto de las prácticas

El mayor beneficio de las prácticas se consigue con su aplicación conjunta y equilibrada puesto que se apoyan unas en otras. Esto se ilustra en la Figura 1.6, donde una línea entre dos prácticas significa que las dos prácticas se refuerzan entre sí. La mayoría de las prácticas propuestas por XP no son novedosas sino que en alguna forma ya habían sido propuestas en ingeniería del software e incluso demostrado su valor en la práctica. El mérito de XP es integrarlas de una forma efectiva y complementarlas con otras ideas desde la perspectiva del negocio, los valores humanos y el trabajo en equipo.

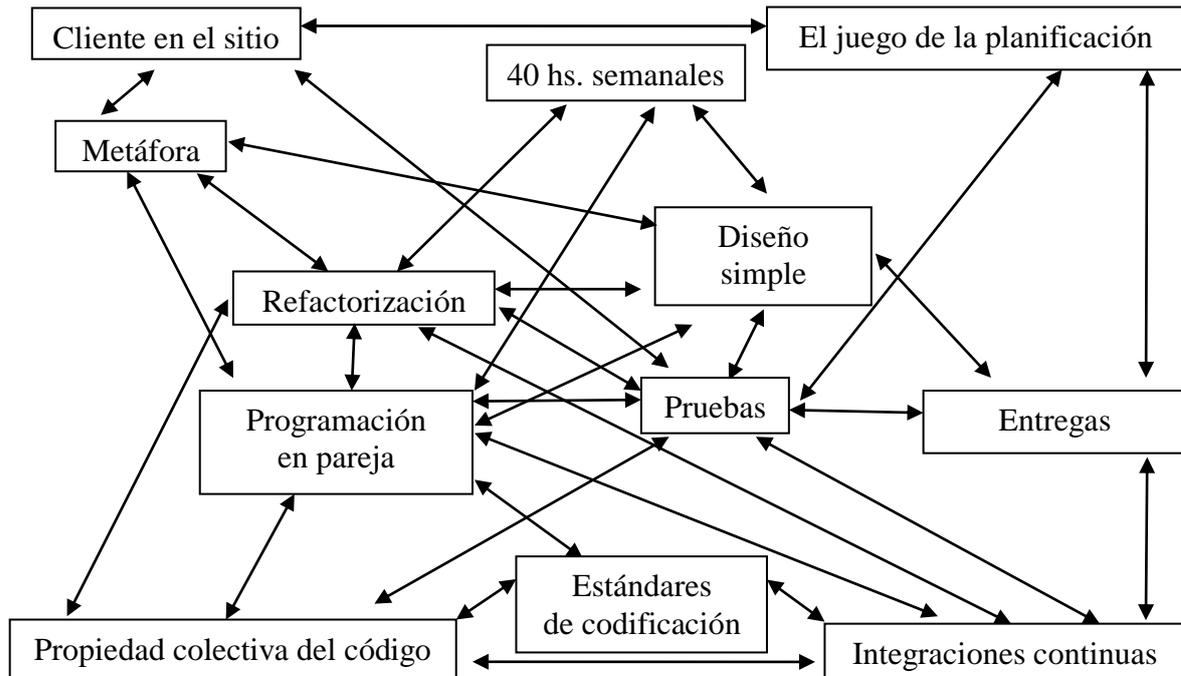


Figura 1-5 Las prácticas se refuerzan entre si [1]

### 1.3.7 Roles en XP

En este apartado describiremos los roles de acuerdo con la propuesta original de Kent Beck [1].

- **Programador:** son los responsables de decisiones técnicas y de construir el sistema, lo programan y lo prueban
- **Cliente:** es parte del equipo, determina qué construir y cuándo, también establece las pruebas funcionales, es sólo uno dentro del proyecto, pero puede corresponder a un interlocutor que está representando a varias personas que se verán afectadas por el sistema, el escribe las historias de usuario y las pruebas funcionales para validar su implementación, además, asigna la prioridad a las historias de usuario y decide cuáles se implementan en cada iteración centrándose en aportar mayor valor al negocio.
- **Encargado de pruebas (Tester):** ayuda al cliente a escribir las pruebas funcionales. Ejecuta las pruebas regularmente, difunde los resultados en el equipo y es responsable de las herramientas de soporte para pruebas.
- **Encargado de seguimiento (Tracker):** proporciona realimentación al equipo en el proceso XP. Su responsabilidad es verificar el grado de acierto entre las estimaciones realizadas y el tiempo real dedicado, comunicando los resultados para mejorar futuras estimaciones.  
También realiza el seguimiento del progreso de cada iteración y evalúa si los objetivos son alcanzables con las restricciones de tiempo y recursos presentes. Determina cuándo es necesario realizar algún cambio para lograr los objetivos de cada iteración.
- **Entrenador (Coach):** es responsable del proceso en su conjunto, debe conocer bien el proceso XP para guiar a los miembros del equipo de forma que se apliquen las prácticas XP y se siga el proceso correctamente.
- **Consultor:** es un miembro externo del equipo con un conocimiento específico en algún tema necesario para el proyecto. Guía al equipo para resolver un problema específico.
- **Gestor (Big boss):** es el vínculo entre clientes y programadores, ayuda a que el equipo trabaje efectivamente creando las condiciones adecuadas. Su labor esencial es de coordinación.

### 1.3.8 Proceso XP

Un proyecto XP tiene éxito cuando el cliente selecciona el valor de negocio a implementar basado en la habilidad del equipo para medir la funcionalidad que puede entregar a través del tiempo. El ciclo de desarrollo consiste (a grandes rasgos) en los siguientes pasos:

1. El cliente define el valor de negocio a implementar.
2. El programador estima el esfuerzo necesario para su implementación.
3. El cliente selecciona qué construir, de acuerdo con sus prioridades y las restricciones de tiempo.
4. El programador construye ese valor de negocio.
5. Vuelve al paso 1.



- **Fase del planeamiento:** En esta fase el cliente establece la prioridad de cada historia de usuario, y correspondientemente, los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente. Una entrega debería obtenerse en no más de tres meses. Esta fase dura unos pocos días.

Las estimaciones de esfuerzo asociado a la implementación de las historias la establecen los programadores utilizando como medida el punto. Un punto, equivale a una semana ideal de programación. Las historias generalmente valen de 1 a 3 puntos. Por otra parte, el equipo de desarrollo mantiene un registro de la “velocidad” de desarrollo, establecida en puntos por iteración, basándose principalmente en la suma de puntos correspondientes a las historias de usuario que fueron terminadas en la última iteración.

La planificación se puede realizar basándose en el tiempo o el alcance. La velocidad del proyecto es utilizada para establecer cuántas historias se pueden implementar antes de una fecha determinada o cuánto tiempo tomará implementar un conjunto de historias. Al planificar por tiempo, se multiplica el número de iteraciones por la velocidad del proyecto, determinándose cuántos puntos se pueden completar. Al planificar según alcance del sistema, se divide la suma de puntos de las historias de usuario seleccionadas entre la velocidad del proyecto, obteniendo el número de iteraciones necesarias para su implementación [1].

- **Fase de iteraciones:** Esta fase incluye varias iteraciones sobre el sistema antes de ser entregado. El Plan de Entrega está compuesto por iteraciones de no más de tres semanas. En la primera iteración se puede intentar establecer una arquitectura del sistema que pueda ser utilizada durante el resto del proyecto. Esto se logra escogiendo las historias que fueren la creación de esta arquitectura, sin embargo, esto no siempre es posible ya que es el cliente quien decide qué historias se implementarán en cada iteración (para maximizar el valor de negocio). Al final de la última iteración el sistema estará listo para entrar en producción. Los elementos que deben tomarse en cuenta durante la elaboración del Plan de la Iteración son: historias de usuario no abordadas, velocidad del proyecto, pruebas de aceptación no superadas y tareas no terminadas en la iteración anterior. Todo el trabajo de la iteración es expresado en tareas de programación, cada una de ellas es asignada a un programador como responsable, pero llevadas a cabo por parejas de programadores y luego se realiza la prueba de aceptación [1].
- **Fase de producción:** requiere de pruebas adicionales y revisiones de rendimiento antes de que el sistema sea trasladado al entorno del cliente. Al mismo tiempo, se deben tomar decisiones sobre la inclusión de nuevas características a la versión actual, debido a cambios durante esta fase.

Es posible que se rebaje el tiempo que toma cada iteración, de tres a una semana. Las ideas que han sido propuestas y las sugerencias son documentadas para su posterior implementación (por ejemplo, durante la fase de mantenimiento) [1].

Si bien al final de cada iteración se entregan módulos funcionales y sin errores, puede ser deseable por parte del cliente no poner el sistema en producción hasta tanto no se tenga la funcionalidad completa. En esta fase no se realizan más desarrollos funcionales, pero pueden ser necesarias tareas de ajuste [6].

- **Fase de mantenimiento:** Mientras la primera versión se encuentra en producción, el proyecto XP debe mantener el sistema en funcionamiento al mismo tiempo que desarrolla nuevas iteraciones. Para realizar esto se requiere de tareas de soporte para el cliente. De esta forma, la velocidad de desarrollo puede bajar después de la puesta del sistema en producción. La fase de mantenimiento puede requerir nuevo personal dentro del equipo y cambios en su estructura [1].
- **Fase de muerte:** Es cuando el cliente no tiene más historias para ser incluidas en el sistema. Esto requiere que se satisfagan las necesidades del cliente en otros aspectos como rendimiento y confiabilidad del sistema. Se genera la documentación final del sistema y no se realizan más cambios en la arquitectura. La muerte del proyecto también ocurre cuando el sistema no genera los beneficios esperados por el cliente o cuando no hay presupuesto para mantenerlo [1].

# CAPÍTULO II

---

## **Marco Metodológico**

## 2.1 INTRODUCCIÓN

En el siguiente capítulo se presenta los problemas detectados en la Escuela Secundaria N° 44 “Jorge Luis Borges” de la Loc. Miraflores, Departamento Capayán, Provincia de Catamarca, los objetivos que se propusieron para llevar a cabo el desarrollo del sistema de gestión administrativa de este establecimiento secundario, la metodología de trabajo donde se establece las técnicas e instrumentos que se utilizaron para la recolección de datos y las tareas llevadas a cabo.

## 2.2 PLANTEAMIENTO DEL PROBLEMA

El trabajo final se llevó a cabo en la Escuela Secundaria N°44 “Jorge Luis Borges”, ubicada en la localidad de Miraflores Departamento Capayán, Provincia de Catamarca, en la misma se observaron los siguientes problemas:

- El incremento significativo de la matrícula y del personal de la escuela, produce un aumento en la cantidad de la documentación que se recibe, analiza, organiza, registra y emite.
- Se llevan a cabo las tareas en forma manual, lo que origina errores de registración de datos.
- Duplicación de la Información, ya que se registra varias veces en distintos documentos, por una o varias persona.
- El acceso a la información no es inmediata, debido a que está dispersa en distintos documentos y no esta digitalizada.

Estos problemas se desarrollan con mayor detalle en la descripción de la organización, de la fase de exploración, del Capítulo III.

## 2.3 JUSTIFICACIÓN

El gobierno de la nación a través del Decreto 459/10 crea el Programa Conectar Igualdad que define una política de inclusión digital para la educación pública y establece la distribución de una computadora portátil a alumnas, alumnos y docentes de educación secundaria, de educación especial y de Institutos de Formación Docente [3].

En el año 2013, a través del programa antes mencionado la escuela se ve beneficiada con la entrega de netbooks, para ayudar a obtener una educación de excelencia. Se pretende usar estas nuevas TICs provistas, con el sistema de software que se desarrolló, que centralizará la información, optimizando así la gestión administrativa, accediendo a través de Internet en forma sencilla y rápida, disminuyendo la tarea que se realiza en forma manual.

## 2.4 OBJETIVOS

A continuación se presenta el objetivo general y los objetivos específicos planteados para el desarrollo del trabajo final.

### 2.4.1 Objetivo General

Desarrollar un sistema de información, para mejorar el proceso de gestión administrativa de la Escuela Secundaria N°44 “Jorge Luis Borges”, utilizando la Metodología Ágil XP.

### 2.4.2 Objetivos Específicos

- Crear una base de datos para registrar, actualizar y consultar la información.
- Centralizar la información para eliminar la duplicación de datos.
- Automatizar las tareas para disminuir la carga de trabajo respecto a los controles que se realizan en forma manual.
- Mejorar el acceso a los datos para obtener información oportuna para la toma de decisiones.

## 2.5 METODOLOGÍA DE TRABAJO

El trabajo final involucro la aplicación de la Metodología XP, para desarrollar el sistema de gestión administrativa de la Escuela Secundaria N°44 “Jorge Luis Borges”.

Para la recolección de datos se utilizaron las siguientes Técnicas e Instrumentos:

- Análisis de Contenidos: permitió realizar la sistematización bibliográfica.
- Observación ordinaria y/o participante: permitió acumular y normalizar información sobre las tareas de la escuela, para ello se utilizó el cuaderno de notas como instrumento.
- Entrevista: se emplearon entrevistas semi-estructuradas debido a que se elaboraron interrogantes previamente y otras preguntas fueron formuladas en el momento. Para esta técnica se utilizó el instrumento de la hoja de entrevista.
- Las historias de usuario: son la técnica utilizada en XP para especificar los requisitos del software. Se trata de tarjetas de papel en las cuales el cliente describe brevemente las características que el sistema debe poseer, sean requerimientos funcionales o no funcionales.

Para el desarrollo del trabajo se cubrieron las tareas que se describen a continuación.

### 2.5.1 Análisis Exploratorio

Consistió en la búsqueda, recolección, lectura comprensiva y análisis de las fuentes de información (bibliografía, publicaciones, sitios web, entre otros) referidas al tema que trata el trabajo, que permitió expresar las bases teóricas y conceptuales para la aplicación de la metodología XP.

### **2.5.2 Desarrollo del Sistema de Información**

Para el desarrollo del Sistema de Información se utilizó la metodología XP la cual consta de las siguientes fases.

#### **Fase de Exploración:**

- Redacción de las historias de usuarios(requerimientos)
- Selección y uso de tecnología, herramientas, recursos.
- Definición de la arquitectura del sistema.

#### **Fase de Planificación:**

- Priorización de las historias de usuarios, se estimaron recursos y establecieron acuerdos de contenidos y las reuniones con el cliente.
- Planificación de las historias de usuarios utilizadas en cada prototipo.
- Ajustes al proyecto XP.

#### **Fase Iteraciones:**

- Desarrollo del plan de iteraciones considerando historias de usuarios, recursos, prioridades del cliente.
- Entrega o muestra de prototipos.
- Prueba de aceptación.

### **2.5.3 Análisis de Resultados y Elaboración de Conclusiones**

En este punto se evaluaron los resultados obtenidos en el transcurso del trabajo y el desarrollo del sistema.

### **2.5.4 Elaboración del Informe Final del Trabajo**

Esta etapa involucró la redacción del informe del trabajo final, y la aplicación de diferentes técnicas para la documentación de la totalidad de los recursos obtenidos y/o utilizados a lo largo del trabajo. Se efectuó el informe correspondiente donde se comunican y socializan las conclusiones de la investigación y el análisis realizado.

# CAPÍTULO III

---

## Desarrollo del Software

### 3.1 INTRODUCCIÓN

En el presente capítulo se describe las herramientas, tecnologías y el conjunto de actividades que involucraron el desarrollo del producto de software.

### 3.2 ACTIVIDADES EN LAS FASES DESARROLLADAS

A continuación se describen las actividades llevadas a cabo para luego mostrar los artefactos obtenidos en las distintas fases que propone la metodología XP.

#### 3.2.1 Requerimientos y Análisis

Una vez completadas las actividades de presentación con el cliente, se comenzó el desarrollo netamente iterativo. Las iteraciones se suceden mediante la realización de actividades de las disciplinas de requerimientos y análisis, diseño, implementación y prueba. A continuación, se explica en detalle la primera disciplina con que comienzan las iteraciones del sistema desarrollado.

En las actividades de *requerimientos* y *análisis* se capturaron las necesidades de las autoridades de la escuela, se especificó la funcionalidad que sirvió para el desarrollo, diseño e implementación del prototipo.

Para llevar a cabo esta transferencia de conocimiento desde los usuarios que formaron parte del proyecto a los analistas del equipo de desarrollo, se utilizaron técnicas que no demandaron gran cantidad de recursos o una significativa infraestructura. Una de las técnicas más utilizadas fueron las entrevistas, las cuales tuvieron un carácter más informal ya que se dieron en cualquier momento que el analista requería aclarar cuestiones respecto a los requerimientos.

Posteriormente a la captura de requerimientos se realiza la primera aproximación de los mismos en una forma estructurada que se registraron en historias de usuarios.

#### 3.2.2 Diseño

Dentro del Sistema desarrollado el *diseño* fue realizado tomando como base las historias de usuarios escritos en la disciplina anterior.

Durante las primeras iteraciones se diseñó la infraestructura sobre la cual fue construido el software. Para esto se tomó del conjunto de las historias de usuarios relevadas las más importantes desde un punto de vista técnico. Con esto se construyó un modelo de arquitectura que especifica los elementos a partir de los cuales fue construido el sistema. Para probar la factibilidad de la arquitectura se construyó un primer prototipo ejecutable utilizado para verificar la conformación a los requerimientos de la aplicación.

Este proceso fue plasmado mediante la generación del modelo de datos y las interfaces de usuarios. Una vez finalizado el proceso de diseño detallado durante la iteración, se procedió a la siguiente disciplina que son la *implementación* y *prueba*.

### 3.2.3 Implementación

En la disciplina de *Implementación* se llevó a cabo la producción del software. El equipo de desarrollo se encargó de implementar la funcionalidad especificada en las historias de usuarios mediante el lenguaje de programación elegido.

Se usaron estándares de codificación para mantener el código legible, ya que XP enfatiza que la comunicación de los programadores es a través del código, en el Anexo I se describen los estándares seguidos para la codificación.

También se aplicó para todo el desarrollo del código fuente la práctica de programación en parejas.

### 3.2.4 Pruebas

En paralelo a la implementación de los componentes se realizó la revisión de los mismos mediante las *pruebas*, siendo está una de las maneras en que se llevan a cabo los mecanismos de control de calidad.

Para el trabajo se utilizaron:

- Pruebas unitarias: se realizaron estas pruebas a cada módulo construido para la aplicación. Estas pruebas fueron implementadas por los mismos programadores.
- Pruebas funcionales: se realizaron este tipo de pruebas para verificar el flujo de interacción de la funcionalidad definida en las historias de usuarios y fueron creadas en conjunto con el usuario durante cada iteración.
- Pruebas de Aceptación: las mismas fueron utilizadas durante todo el desarrollo y su alcance fue determinado por el conjunto completo de requerimientos funcionales definidos en la Fase de Exploración.

## 3.3 FASE DE EXPLORACIÓN

En esta fase se llevaron a cabo las siguientes actividades:

- Descripción de la organización
- Generación de las historias de usuarios
- Definición de las herramientas y tecnologías usadas para el desarrollo del sistema.

### 3.3.1 Descripción de la organización

La Escuela Secundaria N° 44 “Jorge Luis Borges”, ubicada en la localidad de Miraflores Departamento Capayán, Provincia de Catamarca, comienza a funcionar en el año 1990, a ella asisten alumnos de distintas localidades aledañas a la zona de Miraflores, Coneta, Colonia del Valle, Los Pocitos, El Bañado, Colonia Nueva Coneta, Río Ongolí, La Paraguaya, SisiHuasi y Los Ángeles.

Debido al cambio en el Sistema Educativo, a partir de la Ley Nacional de Educación Nro. 26.206 sancionada y promulgada en el año 2006; el nivel obligatorio es la secundaria con una duración de 6 años, que corresponde al E.G.B.3 (Tercer Ciclo Educación General

Básica) y Polimodal de la Ley Federal de Educación, en consecuencia se incrementó en la escuela la matrícula de alumnos, el personal docente y no docente que pertenecían a las escuelas primarias vinculantes: Esc. N° 225 “Fundación de Catamarca”, de la localidad de Coneta; Esc. N° 284 “Sarmiento” y la Esc. N° 490 “Miraflores Este”, de la localidad de Miraflores; Esc. N° 416 “Gregoria Matorra” y la Esc. N° 374 “O.E.A.”, de la localidad de Los Ángeles; en donde actualmente funcionan los ciclos básicos, dependiendo de la escuela secundaria, en lo que hace a la gestión administrativa, institucional y pedagógica.

Se observa los siguientes problemas:

- El incremento significativo de la matrícula y del personal de la escuela, produce un aumento en la cantidad de la documentación que se recibe, analiza, organiza, registra y emite.
- Se llevan a cabo las tareas en forma manual, lo que origina errores de registración de datos.
- Duplicación de la Información, ya que se registra varias veces en distintos documentos, por una o varias personas.
- El acceso a la información no es inmediata, debido a que está dispersa en distintos documentos y no está digitalizada.

Algunas de las tareas administrativas que se realizan en forma manual son:

- Las **inscripción de los alumnos** la realiza las señoras preceptoras, a los alumnos que ingresan por primera vez se le solicita una serie de requisitos que deben entregar en un tiempo prudencial para la confección del legajo, los que ya concurrían se lo inscriben en el curso correspondiente, se cobra la inscripción y se extiende un recibo de pago, en algunos casos solo se acepta lo que pueda entregar el tutor debido a la situación económica. También se becan a alumnos que no pueden pagar y aquellos que poseen más de un hermano se les cobra un monto mínimo de cooperadora por mes, ambos pagos son registrados en el Libro de Aranceles y Cooperadora.  
Debido a que los requisitos de inscripción no son entregados todos juntos, se produce un retraso en el armado de los legajos, la asignación de los alumnos ingresantes a las distintas secciones la realiza el director y luego las preceptoras asientan el listado de alumnos por curso y sección en una planilla Excel y en el Registro de Asistencia, donde se consignan los datos personales y la asistencia.
- La **inscripción para examen** la lleva a cabo la señora secretaria en un cuaderno donde consigna el nombre y apellido del alumno, la materia a rendir y de qué año, luego arma las mesas de examen, elabora el permiso de examen del alumno y las actas volantes de exámenes para los docentes, estos llenan los libros de actas de examen consignando la condición del alumno regular, previos o libres.  
En caso de duda por parte del alumno en las materias a rendir se consulta el Libro Matriz, en donde registran la trayectoria escolar del alumno.
- La **asignación de horarios de clase a los docentes**, la efectúa la señora asesora pedagógica, para ello tiene en cuenta la declaración jurada de los docentes y la aceptación de cualquier cambio por parte de los mismos.

En cuanto a **la información relacionada con el personal docente de la escuela**, está muy dispersa, los datos personales se registran en una planilla todos los años; para consultas sobre: el cargo que desempeñan, el título que poseen, altas, bajas, licencias y antigüedad, lo buscan en la Planta Orgánica Nominal, que no está actualizada. Para consultar qué día asisten a la escuela los docentes, lo hacen del horario de clase; la asistencia se registra en las Planillas de Asistencia Mensual del Docente; las inasistencias son registradas por las señoras preceptoras en la Planilla de Inasistencia y la señora secretaria coloca el motivo de la misma, luego realiza la Planilla de Novedades que es enviada junto con la justificación de la licencia a la Dirección de Personal del Ministerio de Educación.

- **La elaboración de las libretas** la realizan las señoras preceptoras utilizando: los datos personales de los alumnos, la asistencia que la obtienen del Registro de Asistencia, las Planillas de Calificaciones con las notas de cada materia entregadas por los docentes, las amonestaciones si hubiera y las notas de conducta; toda esta información es escrita en el Registro Anual de Calificaciones, en la Planilla de Informe Pedagógico y en el Libro Matriz.
- **La emisión de Constancias** como de Alumno Regular y la de Profesor, la realizan las señoras preceptoras y son firmadas por el director del establecimiento.

### 3.3.2 Historias de Usuario

Los requerimientos planteados por el cliente, fueron asentados en Historias de Usuario, se estudió y se mantuvo conversaciones permanentes con el personal de la Escuela.

De esta redacción inicial de historias de usuario se realizó una planificación inicial y posteriormente fue cambiada a lo largo del proyecto, ya que algunas de estas historias fueron eliminadas o modificadas, a medida que variaban los requerimientos del cliente o se tenía una concepción más clara del proyecto.

Para observar el detalle de las historias de usuario ver Anexo II.

### 3.3.3 Herramientas y Tecnologías que se usaron

Para el desarrollo del sistema de gestión administrativa se utilizaron las tecnologías que se muestran en la Figura 3.1.

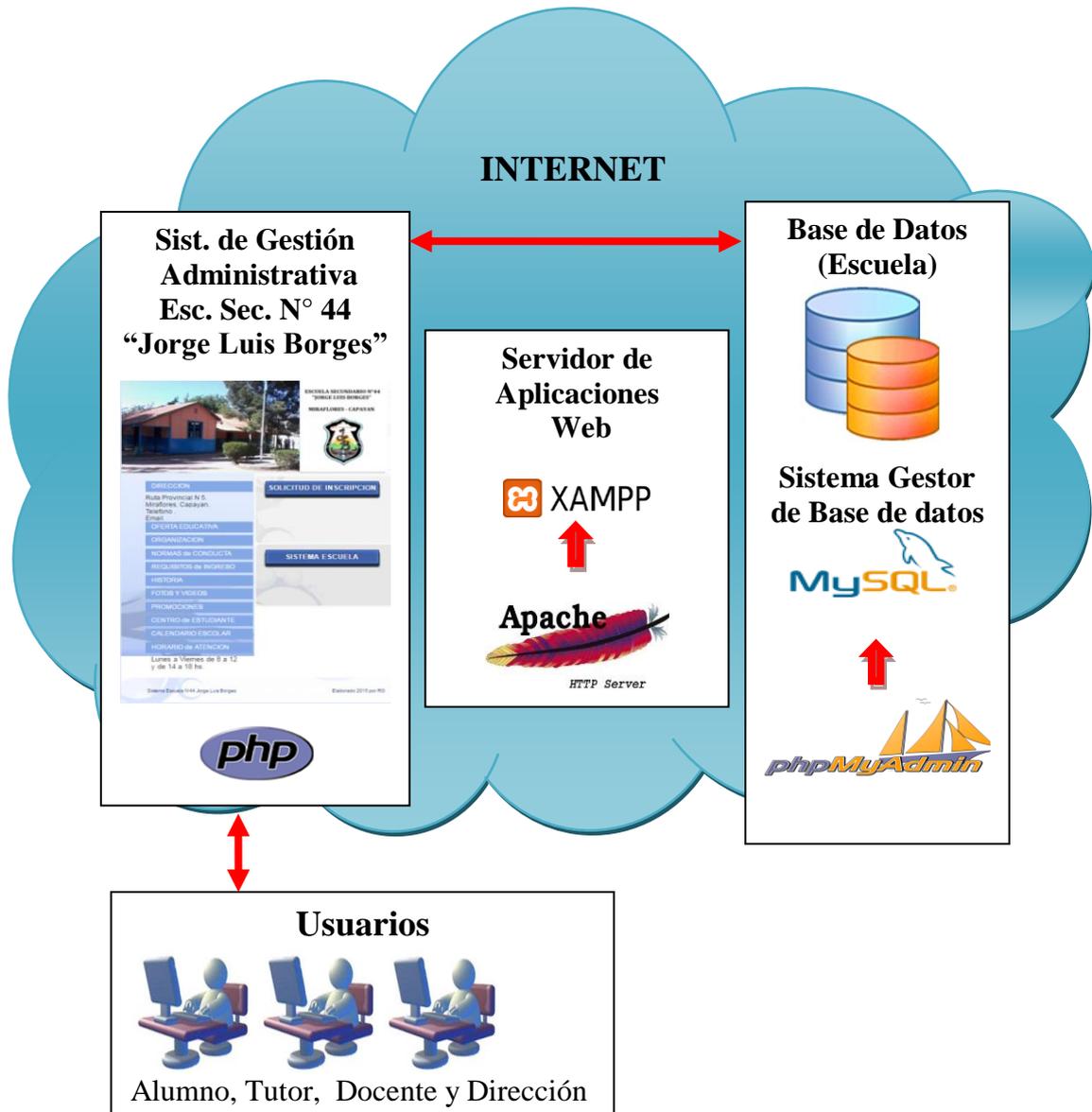


Figura 3-1– Esquema de la arquitectura / Tecnología del sistema

### 3.3.3.1 PHP como herramienta de desarrollo

Para la implementación de la aplicación se utilizó PHP (HypertextPreprocessor), es un lenguaje de programación interpretado de alto nivel para internet. Su objetivo es desarrollar software dinámico en forma sencilla y rápida.

Esto se debe a múltiples factores, entre los cuales se pueden citar los siguientes:

- Es libre y Gratuito. Está amparado bajo el movimiento open source (código abierto), se lo puede usar en forma gratuita sin pagar una licencia y acceder al código fuente original y cambiarlo.

- Disponibilidad. Se encuentra disponible para varios sistemas operativos: Microsoft Windows, Unix, Linux.
- Soporta a múltiples base de datos como Dbase, MySQL, PostgreSQL.
- Evolución. Como no es propiedad de una empresa comercial, las continuas mejoras y avances dentro del lenguaje son gracias a la comunidad de desarrolladores que contribuyen.
- Facilidad de aprendizaje. Es un lenguaje eficaz y simple a la vez.

### 3.3.3.2 MySQL como Servidor de base de datos

Se eligió MySQL por poseer un conjunto de factores y beneficios que se citan a continuación:

- Licenciamiento. Es de libre distribución.
- Facilidad de uso. Es sencillo con respecto a la administración, posee un gran número de herramientas destinadas a controlar su funcionamiento de manera intuitiva.
- Multiplataforma. Está disponible para distintas plataformas y sistemas operativos, por lo cual su disponibilidad no estará afectada por las características del servicio de alojamiento.
- Rapidez. Los niveles de velocidad alcanzados son buenos.
- Seguridad. Es utilizada en el nivel empresarial en grandes proyectos, permite definir esquemas de seguridad acorde con las necesidades.
- Estabilidad. Genera confianza en quienes lo utilizan para almacenar informaciones sensibles dentro de las organizaciones.

### 3.3.3.3 PHPMYADMIN

PhpMyAdmin es una aplicación realizada en PHP que le permite administrar sus bases de datos MySQL a través de Internet, es un método para gestionar sus bases de datos MySQL. Es un programa cliente que le permite conectarse al servidor de datos con el fin de crear las tablas, subir datos, editar registros.

### 3.3.3.4 XAMPP

Es un servidor de sitios Web, multiplataforma, constituido por un servidor HTTP Apache. El acrónimo *Xampp* está formado por X (la letra inicial de cualquier sistema operativo), A de Apache, de MySQL, y de PHP y de Perl. Actualmente soporta Microsoft Windows, Linux, Sun Solaris y Mac Os X. XAMPP se actualiza regularmente para incorporar las últimas versiones de Apache/MySQL/PHP y Perl.

### 3.3.3.5 Servidor HTTP Apache

Es un servidor web de distribución libre y de código abierto, para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual. El servidor Apache se desarrolla dentro del proyecto HTTP Server (httpd) de la Apache Software Foundation. Apache presenta entre otras características que es altamente configurable, bases de datos de autenticación y negociado de contenido.

### 3.4 FASE DE PLANIFICACIÓN

#### 3.4.1 Tareas realizadas

Las historia de usuario se analizaron para determinar las tareas que se llevaron a cabo en cada historia, estas tareas se registraron en unas tarjetas de papel.

Para ver el detalle de las tareas relevadas ver Anexo II.

Se priorizaron las historias de usuarios, se estimaron los recursos y se establecieron acuerdos con las autoridades de la escuela, lo que permitió la planificación de prototipos los cuales fueron ajustados teniendo en cuenta los nuevos requerimientos planteados por las autoridades de la escuela.

#### 3.4.2 Planificación de Prototipos

La planificación, llevó a la elaboración de un cronograma de entregas realizado junto con las autoridades de la escuela, lo que permitió evaluar el valor del trabajo y dirigir la evolución del sistema. Como resultado de esta actividad se dispone, en la Tabla 3.1, las distintas versiones o prototipos que se desarrollaron, las operaciones que se implementaron y en que versiones se incluyeron cada una de las operaciones. También se incluyen la estimación en semanas que tomó llevar a cabo la implementación de cada prototipo. Y el porcentaje del desarrollo del sistema.

Prototipo	Requerimientos/Prototipo	Semanas	Funcionalidad %
1	<ul style="list-style-type: none"><li>• Solicitud inscripción de alumno</li><li>• Inscripción alumno</li></ul>	12	20%
2	<ul style="list-style-type: none"><li>• Completar inscripción</li><li>• Altas, bajas y modificaciones del personal</li><li>• Registrar horarios</li></ul>	8	40%
3	<ul style="list-style-type: none"><li>• Registrar notas</li><li>• Registrar novedades</li><li>• Registrar exámenes</li></ul>	14	80%
4	<ul style="list-style-type: none"><li>• Emisión de informes, certificados y constancias</li><li>• Registrar cooperadora</li></ul>	7	100%

Tabla 3.1 Planificación de los Prototipos

### 3.5 FASE DE ITERACIÓN

La aplicación se construyó en base a pequeños incrementos que se traducen en prototipos totalmente funcionales, estos se elaboraron según lo planteado en la sección anterior.

La construcción de cada prototipo se llevó a cabo empleando un número definido de iteraciones, donde éstas se planificaron, estimando el nivel de esfuerzo y los tiempos que estas requirieron, en función de los requisitos incluidos en cada iteración y en cada prototipo.

Los objetivos fijados para una iteración no cubiertos o cambios que debieron realizarse en lo ya implementado, se asignaron a la siguiente iteración con la prioridad de ser lo primero en abordarse.

Al terminar el desarrollo de los prototipos fueron presentados a las autoridades de la escuela con las cuales se realizó las pruebas de aceptación.

#### 3.5.1 DESARROLLO DEL PRIMER PROTOTIPO

Alcanzar un diseño base para mostrar los requerimientos iniciales sugeridos por las autoridades de la escuela, es uno de los objetivos principales plasmado en este primer prototipo. En esta primera versión, se buscó implementar en un 20%, la funcionalidad de la aplicación (ver Tabla 3.2).

Primer Prototipo: se implementó las funcionalidades básicas relacionadas con la inscripción del alumno.

PROTOTIPO	ITERACIONES	HISTORIAS DE USUARIO	TAREAS
1	1	1 Solicitud inscripción de alumno.	1.- Altas/Bajas/Modificaciones de cursos/divisiones/turnos/materias
			2.- Verificar existencia del alumno.
			3.- Registrar datos personales del alumno.
			4.- Registrar contactos.
			5.- Registrar ficha médica.
			6.- Registrar datos del tutor.
			7.- Registrar foto del alumno.
	2	2 Inscripción alumno.	1.- Modificar datos personales.
			2.- Asignar curso, división y turno.
			3.- Registrar documentación requerida.
			4.- Registrar baja del alumno.

**Tabla 3.2 Planificación del Primer Prototipo**



### 3.5.1.1 DESCRIPCION DE FUNCIONALIDAD

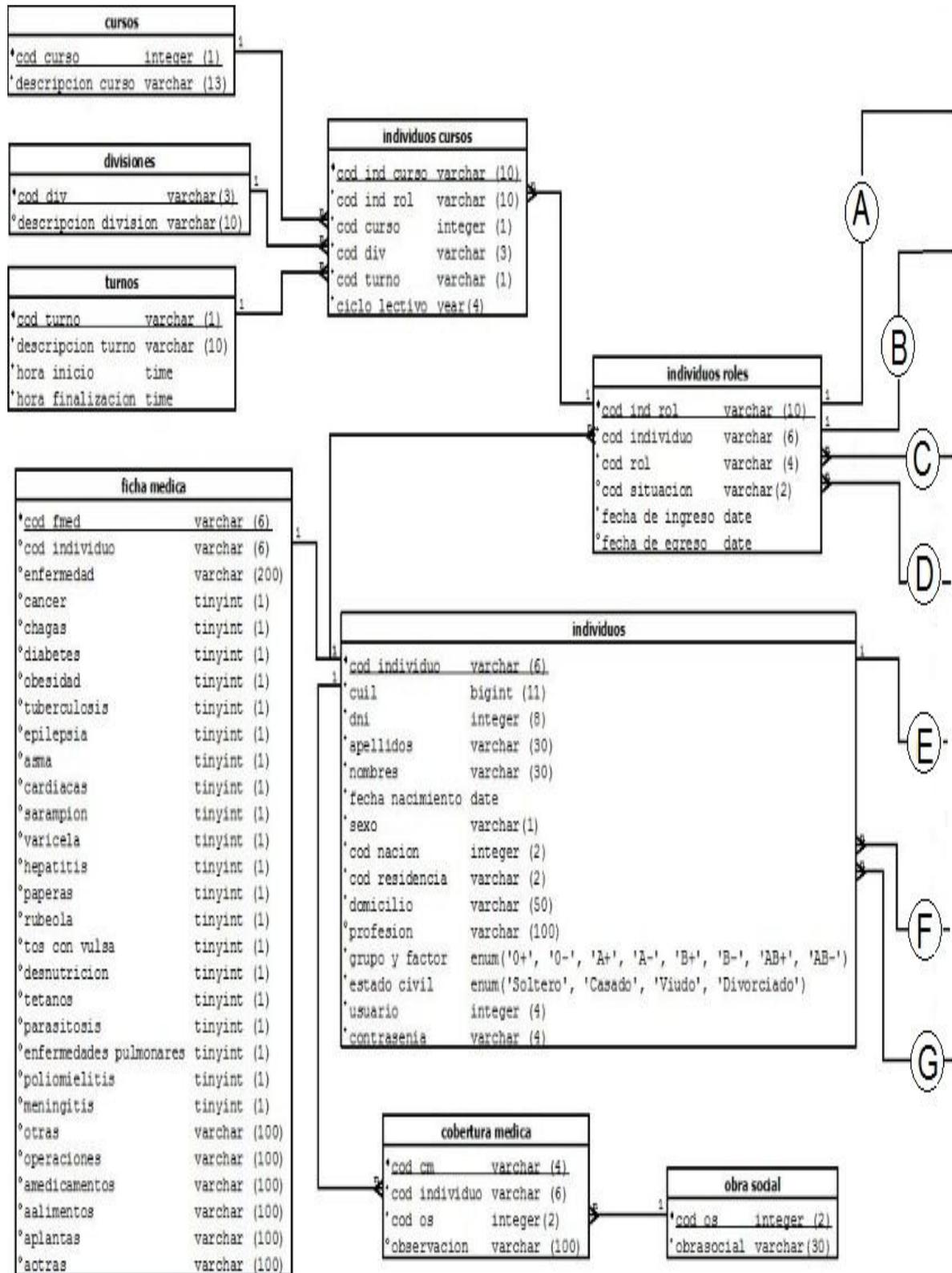
El primer prototipo incluye: un diseño de interfaz de usuario sencillo que soporta la gestión de datos de cursos, divisiones, turnos, materias y el proceso de inscripción de alumno.

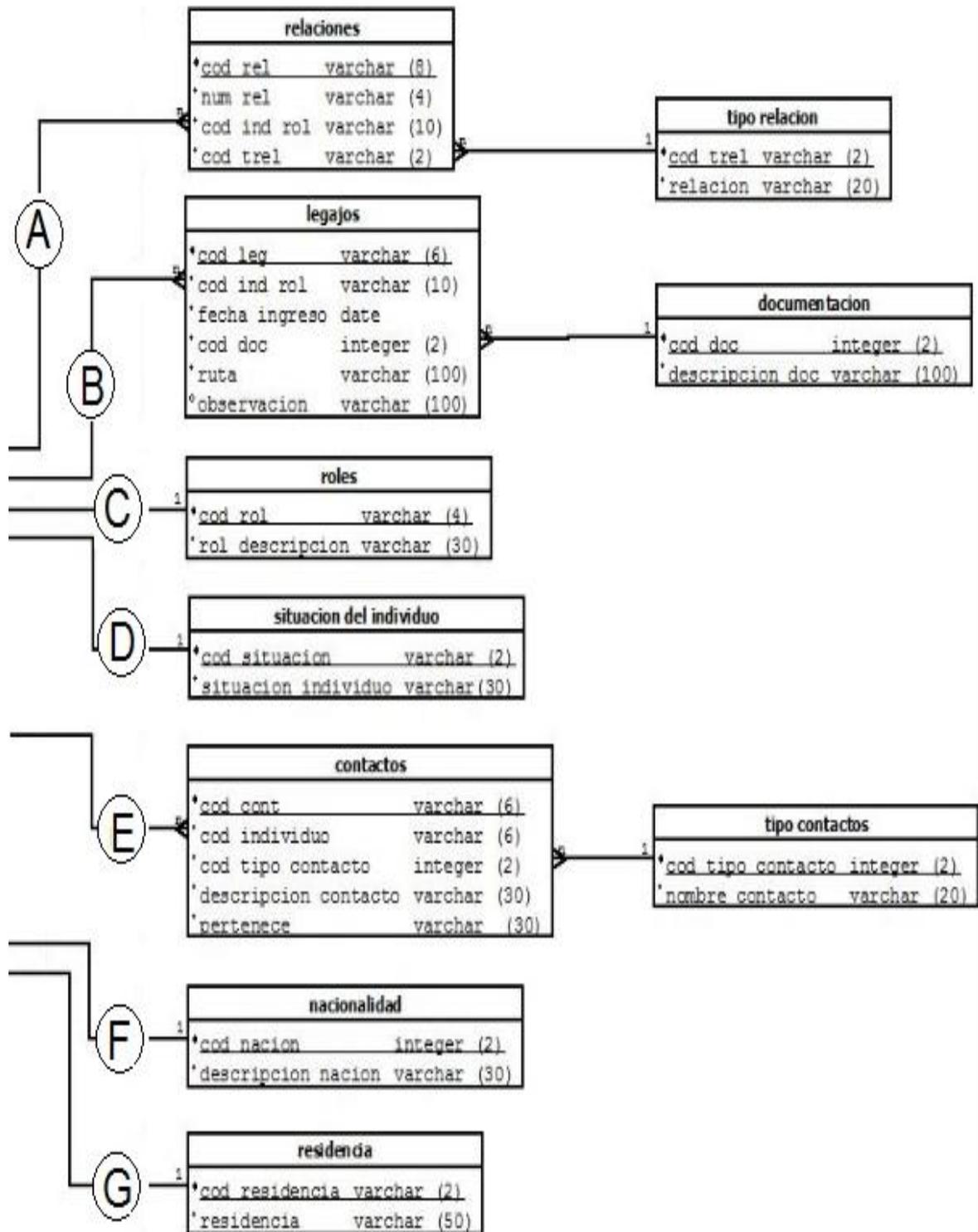
Al realizar la inscripción por primera vez se ingresa al menú principal y se selecciona SOLICITUD DE INSCRIPCION, luego ingresar los datos personales del alumno que son obligatorios también puede ingresar los datos de la ficha médica, contacto, foto y tutor, o los puede completar en otro momento, al final se le asigna usuario y contraseña para ingresar al sistema.

Para completar la inscripción el preceptor le asignara el curso-división-turno definitivo y el sistema las materias a cursar. También permite registrar la documentación solicitada, si esta no está completa se puede dar de baja al alumno por incumplimiento en los requisitos de inscripción. La baja del alumno también puede ser por el pase del alumno.

Tanto el alumno, tutor o preceptor pueden completar la información necesaria para completar la inscripción, ingresando al sistema con su clave de usuario y contraseña.

### 3.5.1.2 MODELOS DE DATOS





### 3.5.1.3 INTERFAZ DE USUARIOS

#### **Historia 1: Solicitud inscripción de alumno.**

**Tarea 1:** Altas/Bajas/Modificaciones de cursos/divisiones/turnos/materias.

La realiza la parte directiva al ingresar al MENU PRINCIPAL, seleccionar SISTEMA ESCUELA, ingresar usuario contraseña, seleccionar el rol con el que desea entrar e ingresa al MENU DIRECCION, seleccionar ALTAS/BAJAS/MODIFICACIONES, se abre la ventana de ALTAS/BAJAS/MODIFICACIONES.





SISTEMA DE GESTION ADMINISTRATIVA	
Ingresar usuario	<input type="text" value="4308"/>
Ingresar contraseña	<input type="password" value="...."/>
<input type="button" value="Buscar"/> <input type="button" value="Cambiar Clave"/> <input type="button" value="Volver"/>	

BIENVENIDO AL SISTEMA
ARIAS ANA SOL
Usted puede ingresar al sistema con los siguientes roles:
<input type="text" value="alumno"/>
<input type="button" value="Seguir"/> <input type="button" value="Volver"/> <input type="button" value="Salir"/>

BIENVENIDO AL SISTEMA
INFANTE HECTOR
Usted puede ingresar al sistema con los siguientes roles:
<input type="text" value="director"/>
<input type="button" value="Seguir"/> <input type="button" value="Volver"/> <input type="button" value="Salir"/>



<b>Altas/Bajas/Modificaciones</b>
<a href="#">Altas</a>
<a href="#">Bajas/Modificaciones</a>
<input type="button" value="Volver"/>

Si se selecciona ALTAS aparece la siguiente pantalla.

<b>Generar Altas</b>
<a href="#">Generar Cursos</a>
<a href="#">Generar Divisiones</a>
<a href="#">Generar Turnos</a>
<a href="#">Generar Curso/División/Turno</a>
<a href="#">Generar Materias</a>
<input type="button" value="Volver"/>



A continuación se muestran las pantallas para generar las altas.

CURSOS	
<u>Dirección:</u> INFANTE HECTOR	Fecha: 04/10/2016 Hora: 13:53
Curso: 4	
Descripción del Curso	<input type="text" value="Cuarto Año"/>
<input type="button" value="Guardar"/> <input type="button" value="Volver"/> <input type="button" value="Salir"/>	

DIVISIONES	
<u>Dirección:</u> INFANTE HECTOR	Fecha: 03/10/2016 Hora: 18:23
División	<input type="text"/>
Descripción de la División	<input type="text"/>
<input type="button" value="Guardar"/> <input type="button" value="Volver"/> <input type="button" value="Salir"/>	

TURNOS	
<u>Dirección:</u> INFANTE HECTOR	Fecha: 03/10/2016 Hora: 18:24
Turnos	<input type="text"/>
Descripción del Turno	<input type="text"/>
Hora de Inicio	<input type="text" value="7"/> Minutos <input type="text" value="0"/>
Hora de Finalización	<input type="text" value="7"/> Minutos <input type="text" value="0"/>
<input type="button" value="Guardar"/> <input type="button" value="Volver"/> <input type="button" value="Salir"/>	



CURSOS/DIVISION/TURNO	
Dirección: INFANTE HECTOR	Fecha: 03/10/2016 Hora: 18:25
Curso: [primer año ▼] División: [Primera ▼] Turno: [mañana ▼]	
<input type="button" value="Guardar"/> <input type="button" value="Volver"/> <input type="button" value="Salir"/>	

MATERIAS	
Dirección: INFANTE HECTOR	Fecha: 03/10/2016 Hora: 18:26
Curso [primer año ▼]	
Ingresar Nombre de la Materia	<input type="text"/>
Cantida de Horas	<input type="text"/>
<input type="button" value="Guardar"/> <input type="button" value="Volver"/> <input type="button" value="Salir"/>	

Para realizar las bajas y modificaciones.

BAJAS/MODIFICACIONES
<a href="#">Baja/Modificación de Cursos</a>
<a href="#">Baja/Modificación de Divisiones</a>
<a href="#">Baja/Modificación de Turnos</a>
<a href="#">Baja/Modificación de Curso/División/Turno</a>
<a href="#">Baja/Modificación de Materias</a>
<input type="button" value="Volver"/>

### CURSOS

Dirección: INFANTE HECTOR Fecha: 04/10/2016  
Hora: 14:37

Seleccionar curso a dar de baja.

Curso	Baja
PRIMER AÑO	<input type="radio"/>
SEGUNDO AÑO	<input type="radio"/>
TERCER AÑO	<input type="radio"/>
CUARTO AÑO	<input type="radio"/>

### DIVISIONES

Dirección: INFANTE HECTOR Fecha: 04/10/2016  
Hora: 14:38

Seleccionar división a dar de baja.

División	Baja
PRIMERA	<input type="radio"/>
SEGUNDA	<input type="radio"/>

### TURNOS

Dirección: INFANTE HECTOR Fecha: 17/10/2016  
Hora: 18:44

Seleccionar turno a dar de baja.

Turno	Baja
MAÑANA	<input type="radio"/>
TARDE	<input type="radio"/>

### CURSOS/DIVISIONES/TURNOS

Dirección: INFANTE HECTOR Fecha: 04/10/2016  
Hora: 14:41

Seleccionar curso/división/turno a dar de baja.  
Turno/Curso/División:

### MATERIA

Dirección: INFANTE HECTOR Fecha: 04/10/2016  
Hora: 15:9

Curso:PRIMER Año División: PRIMERA Turno:MAÑANA

Seleccione la Materia a eliminar

FORMACION ETICA Y CIUDADANA	<input type="checkbox"/>
LENGUA EXTRANJERA	<input type="checkbox"/>
LENGUA Y LITERATURA	<input type="checkbox"/>
MATEMATICA	<input type="checkbox"/>
EDUCACION FISICA	<input type="checkbox"/>
EDUCACION ARTISTICA MUSICA	<input type="checkbox"/>
HISTORIA	<input type="checkbox"/>
GEOGRAFIA	<input type="checkbox"/>
BIOLOGIA	<input type="checkbox"/>
FISICA-QUIMICA	<input type="checkbox"/>
TECNOLOGIA	<input type="checkbox"/>

Tarea 2: Verificar existencia del alumno.

Cuando la inscripción se realiza por primera vez, selecciona SOLICITUD DE INSCRIPCION, del MENU PRINCIPAL. Aparecerá la siguiente ventana.

### Alumno/a

Ingrese su D.N.I.:

Si no existe el alumno en la base de datos del sistema se mostrará la ventana siguiente.

### Alumno/a

Usted no existe como alumno.  
Si desea inscribirse seleccione registrarse.

Tarea 3: Registrar datos personales del alumno.

Al seleccionar la opción REGISTRARSE, se mostrara la pantalla SOLICITUD DE INSCRIPCIÓN, donde ingresa los datos personales del alumno.

### Solicitud de Inscripción

Fecha: 12/09/2016  
Hora: 23:52

Apellido/Apellidos  \*

Nombre/Nombres  \*

C.U.I.L.  \* D.N.I.  \*

Fecha de nacimiento Día  Mes  Año  \*

Sexo  Masculino  Femenino

Estado Civil  Soltero  Casado  Viudo  Divorciado

Nacionalidad  Residencia  \*

Dirección  x \*

Turno - Curso  \*

Los campos con "\*" son obligatorios

Tarea 4: Registrar contactos.

Luego se le da la opción de ingresar si lo desea a las pantallas Contactos, Ficha Médica, Tutor o Foto.

### Alumno/a: ARIAS ANA SOL

Sus datos han sido registrados correctamente.  
Presione el boton para la siguiente opción.

### Contactos

Alumno/a: ARIAS ANA SOL Fecha: 21/09/2016  
Hora: 0:36

1	Telefono Fijo	Ingresar:	<input type="text"/>	Pertenece a:	<input type="text"/> *
2	Telefono Fijo	Ingresar:	<input type="text"/>	Pertenece a:	<input type="text"/> *
3	Telefono Fijo	Ingresar:	<input type="text"/>	Pertenece a:	<input type="text"/>
4	Telefono Fijo	Ingresar:	<input type="text"/>	Pertenece a:	<input type="text"/>
5	Telefono Fijo	Ingresar:	<input type="text"/>	Pertenece a:	<input type="text"/>

Los campos con " \* " son obligatorios

Tarea 5: Registrar ficha médica.

### Ficha Medica

Alumno/a: ARIAS ANA SOL Fecha: 12/09/2016  
Hora: 23:59

Grupo y factor  \* Cobertura Medica

**Enfermedades Actuales** ¿Cuál? ¿Necesita cuidados especiales? ¿Toma alguna medicación?

**Antecedentes Familiares**

<input checked="" type="checkbox"/> Cáncer	<input type="checkbox"/> Chagas	<input type="checkbox"/> Diabetes	<input type="checkbox"/> Obesidad
<input type="checkbox"/> Tuberculosis	<input type="checkbox"/> Epilepsia	<input type="checkbox"/> Asma	<input checked="" type="checkbox"/> Cardiacas

**Antecedentes Personales**

<input type="checkbox"/> Sarampión	<input checked="" type="checkbox"/> Varicela	<input type="checkbox"/> Hepatitis	<input type="checkbox"/> Paperas	<input type="checkbox"/> Tos con vultsa	<input type="checkbox"/> Meningitis
<input type="checkbox"/> Rubeola	<input type="checkbox"/> Tétanos	<input type="checkbox"/> Parasitosis	<input type="checkbox"/> Desnutrición	<input type="checkbox"/> Enfermedades pulmonares	<input type="checkbox"/> Poliomieltis

Otras (especificar)

Intervenciones Quirúrgicas

Alergias Medicamentos

Alergias Alimentos

Alergias Plantas

Otras Alergias

Los campos con " \* " son obligatorios

**Alumno/a: ARIAS ANA SOL**

¿Desea ingresar observaciones sobre las cobertura medicas seleccionadas?

### Observaciones

Alumno/a: ARIAS ANA SOL Fecha: 13/09/2016  
Hora: 0:3

OSEP ▼ 3456 x

Guardar y Seguir    Borrar    Volver

Tarea 6: Registrar datos del tutor.

### Tutor

Alumno/a: ARIAS ANA SOL Fecha: 13/09/2016  
Hora: 0:6

Apellido/Apellidos Arias \*  
Nombre/Nombres José Alberto \*  
C.U.I.L. 54545454545 \* D.N.I. 54545454 \*  
Fecha de nacimiento Día 1 ▼ Mes 1 ▼ Año 1930 ▼ \*  
Sexo  Masculino  Femenino  
Estado Civil  Soltero  Casado  Viudo  Divorciado  
Nacionalidad Argentina ▼ Residencia Catamarca/Capayan/Miraflores ▼ \*  
Dirección ruta n° 5 \*  
Profesión Obrero x \*

Guardar y Salir    Borrar    Volver  
Los campos con " \* " son obligatorios

Tarea 7: Registrar foto del alumno.

### Fotos

Alumno/a: ARIAS ANA SOL Fecha: 13/09/2016  
Hora: 0:8

Elija una foto para su legajo:

C:\Users\Usuario\Pictur Examinar...

Guardar y Seguir    Borrar    Volver

Al finalizar se le asignara su número de usuario y contraseña, como así también la de su tutor.

Escuela Secundaria N°44 "Jorge Luis Borges"	
<u>Alumno/a:</u> ARIAS ANA SOL	<b>Comprobante de inscripción</b>
	Fecha: 13/09/2016 Hora: 0:11
Su número de usuario es: <b>4308</b>	
Su contraseña es: <b>swdx</b>	
El nombre de su tutor es: <b>ARIAS JOSÉ ALBERTO</b>	
El número de usuario de su tutor es: <b>2541</b>	
La contraseña de su tutor es: <b>ipIF</b>	
<input type="button" value="Imprimir"/> <input type="button" value="Guardar"/> <input type="button" value="Salir"/>	

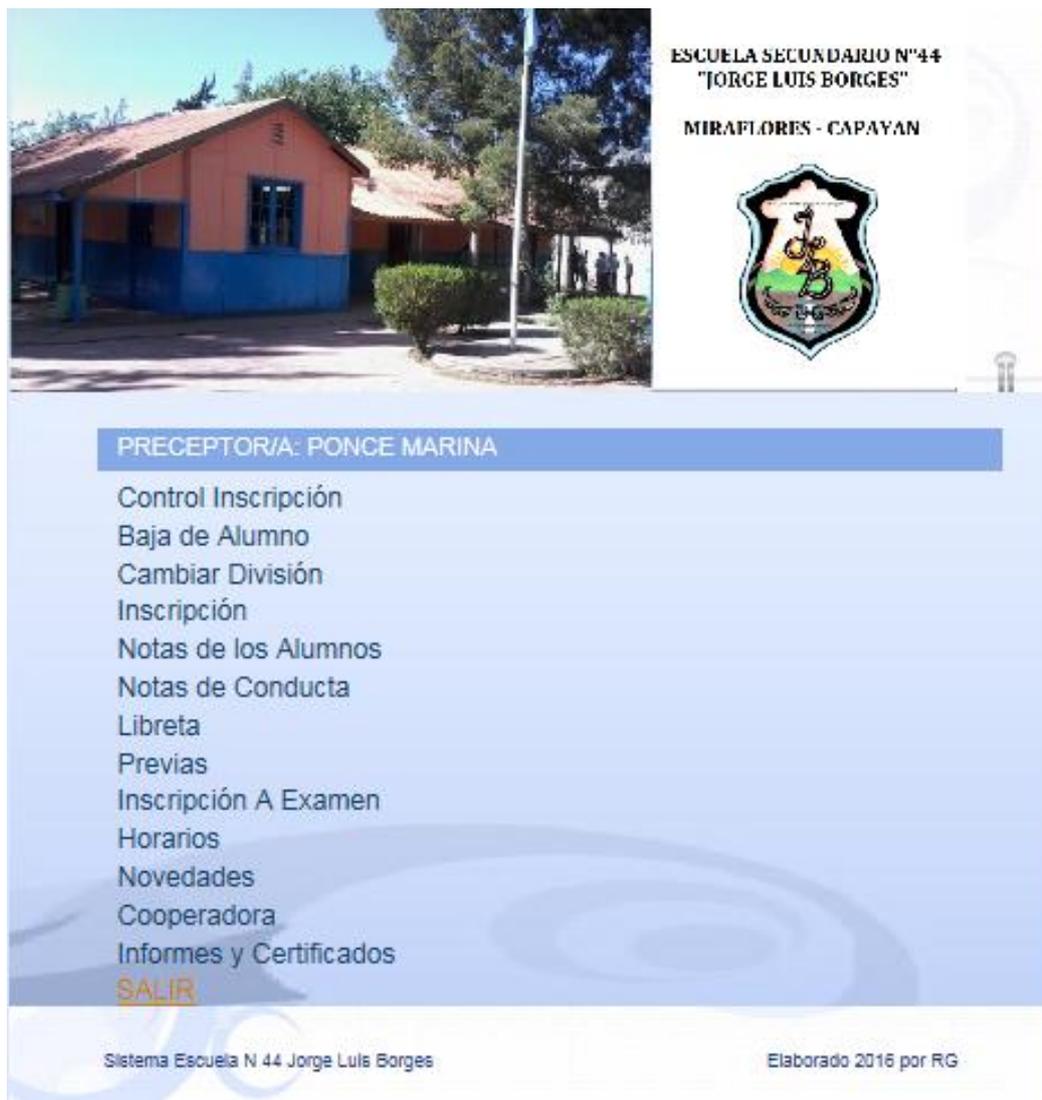
## Historia 2: Inscripción alumno

### Tarea 1: Modificar datos personales.

Para completar datos de la inscripción el alumno, tutor o preceptor, deberá ingresar al MENUPRINCIPAL seleccionar el botón *SISTEMA ESCUELA ingresar usuario contraseña, seleccionar el rol con el que desea entrar y en el menú correspondiente a cada uno seleccionar INSCRIPCION.*

ESCUELA SECUNDARIO N°44 "JORGE LUIS BORGES" MIRAFLORES - CAPAYAN	
	
<b>ALUMNO/A: ARIAS ANA SOL</b>	
Inscripción	
Libreta	
Previas	
Inscripción a Examen	
Exámenes	
Horario	
Novedades	
Cooperadora	
<b>SALIR</b>	
Sistema Escuela N 44 Jorge Luis Borges	
Elaborado 2016 por RG	

El preceptor, en MENÚ PRECEPTOR, debe seleccionar la opción CONTROL INSCRIPCION.



Se mostrara el submenú CONTROL.

<b>Control</b>
<a href="#">Requisitos</a>
<a href="#">Asignar División</a>
<input type="button" value="Volver"/>

### Tarea 2: Asignar curso, división y turno.

Al seleccionar ASIGNAR DIVISION, se introducirá el DNI del alumno, y aparecerá la pantalla siguiente en donde se le seleccionara el Curso, la División y el Turno del alumno.

<b>Alumno/a:</b> ARIAS ANA SOL	Fecha: 13/09/2016 Hora: 0:34	
Curso: 1er. Año	División: Primera	Turno: Mañana
Residencia: Catamarca/Capayan/Miraflores		
Guardar Volver Salir		

### Tarea 3: Registrar documentación requerida.

El preceptor, al seleccionar la opción REQUISITOS, deberá ingresar el DNI del alumno, y aparecerá la siguiente pantalla, en donde se registra si la documentación requerida fue entregada por el alumno.

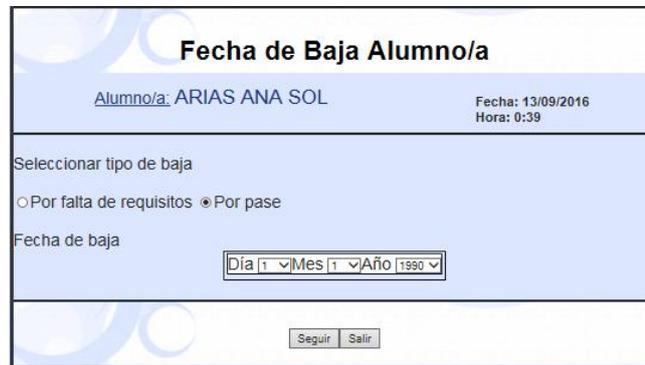
<b>ALUMNO/A:</b> ARIAS ANA SOL	Fecha: 13/09/2016 Hora: 0:36
<b>CURSO:</b> 1er. Año <b>DIVISION:</b> Primera <b>TURNO:</b> Mañana	
<b>DOCUMENTACION</b>	<b>ESTADO</b>
Fotocopia D.N.I.	<input checked="" type="radio"/> Pendiente <input type="radio"/> Entregado
Fotocopia Partida de Nacimiento	<input checked="" type="radio"/> Pendiente <input type="radio"/> Entregado
Constancia de C.U.I.L.	<input checked="" type="radio"/> Pendiente <input type="radio"/> Entregado
Foto	<input type="radio"/> Pendiente <input checked="" type="radio"/> Entregado
Fotocopia D.N.I. Tutor	<input checked="" type="radio"/> Pendiente <input type="radio"/> Entregado
Certificado Analítico Primaria	<input checked="" type="radio"/> Pendiente <input type="radio"/> Entregado
Certificado de Buena Salud	<input checked="" type="radio"/> Pendiente <input type="radio"/> Entregado
Certificado Bucodental	<input checked="" type="radio"/> Pendiente <input type="radio"/> Entregado
Certificado Vista	<input checked="" type="radio"/> Pendiente <input type="radio"/> Entregado
Certificado Oído	<input checked="" type="radio"/> Pendiente <input type="radio"/> Entregado
Guardar Volver Salir	

**Tarea 4: Registrar baja del alumno.**

Para dar de baja a un alumno por falta de requisitos o pase, el preceptor en su menú deberá seleccionar BAJA DE ALUMNO, introducir el DNI del alumno y se abre la pantalla BAJA DE ALUMNO/A.



Luego aparecerá la pantalla FECHA DE BAJA ALUMNO/A.



**3.5.1.4 CRÍTICAS Y SUGERENCIAS**

La Tabla 3.3 reúne las críticas y sugerencias expresadas por las autoridades de la escuela en la evaluación de la presentación del primer prototipo, esta permitió observar nuevos requisitos, y los objetivos que posibilitaron la resolución de las deficiencias de la primera entrega.

NUEVOS REQUISITOS	OBJETIVOS DE LOS REQUISITOS
Emitir un listado con el nombre de los alumnos que adeudan documentación, especificando cuales son.	Mejorar la información para el control de requisito y para dar de baja a un alumno.
Permitir cambiar de división a un alumno.	Satisfacer la solicitud del cambio por parte del tutor o de dirección.

**Tabla 3.3 Nuevos Requisitos**

### 3.5.2 DESARROLLO DEL SEGUNDO PROTOTIPO

La aceptación por parte de las autoridades de la escuela de una primera versión, da apertura a la realización de un segundo prototipo (ver Tabla 3.4), que implementa un segundo grupo de requisitos e incluye los sugeridos en el primer prototipo. Este prototipo está planificado para cubrir un 40% de la funcionalidad total.

*Segundo Prototipo: Se tratara de tener preparadas las funcionalidades especificas relacionadas con completar inscripción que son llevadas a cabo por el preceptor, las altas, bajas y modificaciones del personal, realizada por la dirección del establecimiento escolar y la registración del horario llevada a cabo por el preceptor.*

PROTOTIPO	ITERACIONES	HISTORIAS DE USUARIO	TAREAS
2	1	1 Completar Inscripción	1.- Informar documentación adeudada.
			2.- Cambiar división y turno.
	2	2 Altas, bajas y modificaciones del personal.	1.- Registrar datos del personal del establecimiento.
			2.- Registrar contactos.
			3.- Registrar el cargo a cubrir.
			4.- Registrar foto.
			5.- Registrar el alta del personal.
	3	3 Registrar horario.	6.- Registrar la baja del personal.
			1.- Registrar horario.
2.- Modificar horario.			
			3.- Informar horario.

**Tabla 3.4 Planificación del Segundo Prototipo**

#### 3.5.2.1 DESCRIPCION DE FUNCIONALIDAD

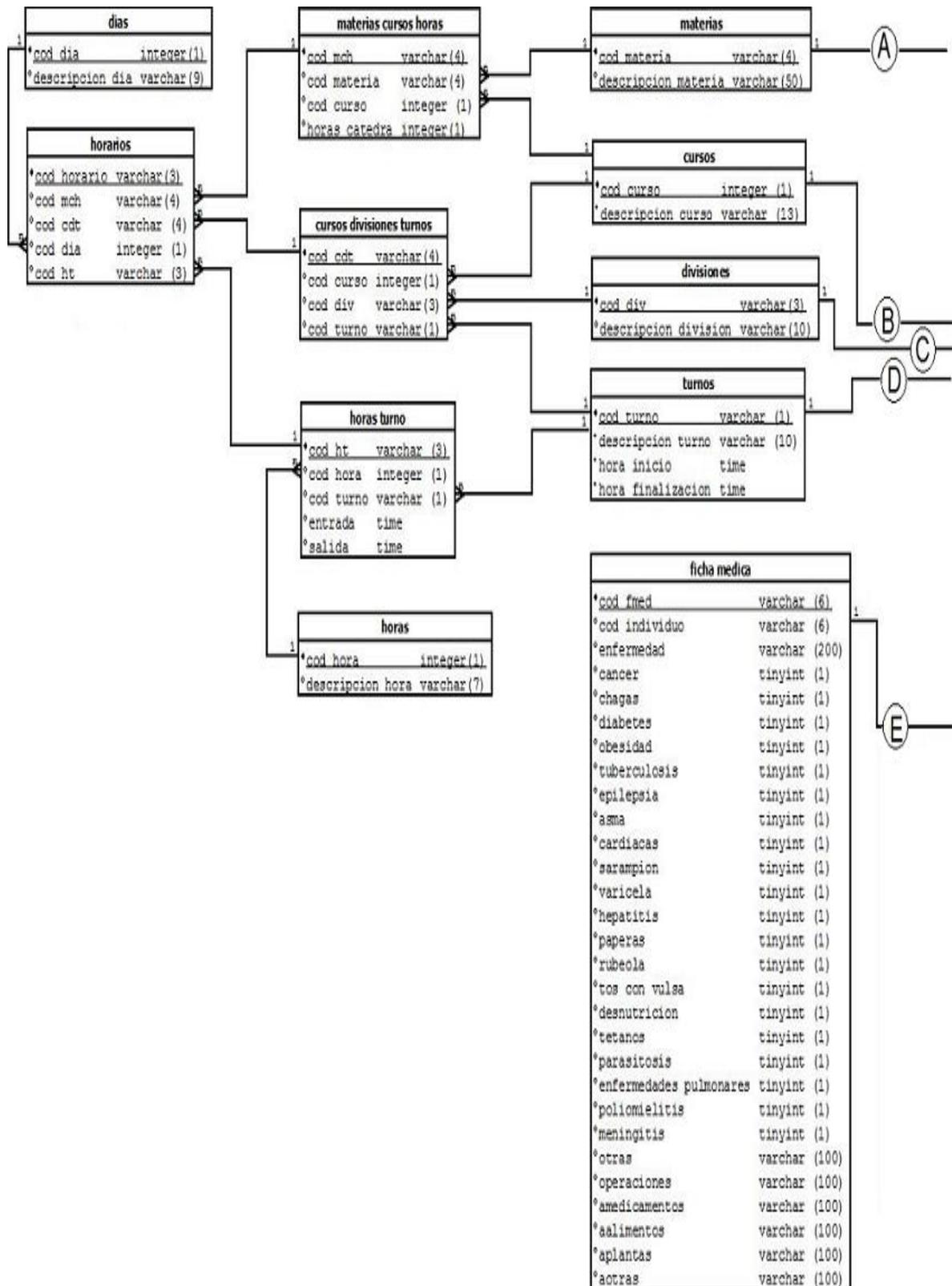
En la Tabla 3.4 se observa la segunda versión del producto de software. Esta incluye un diseño de interfaz que soporta los procesos de: 1) Completar Inscripción; 2) Altas, bajas y modificaciones del personal y 3) Registrar horario.

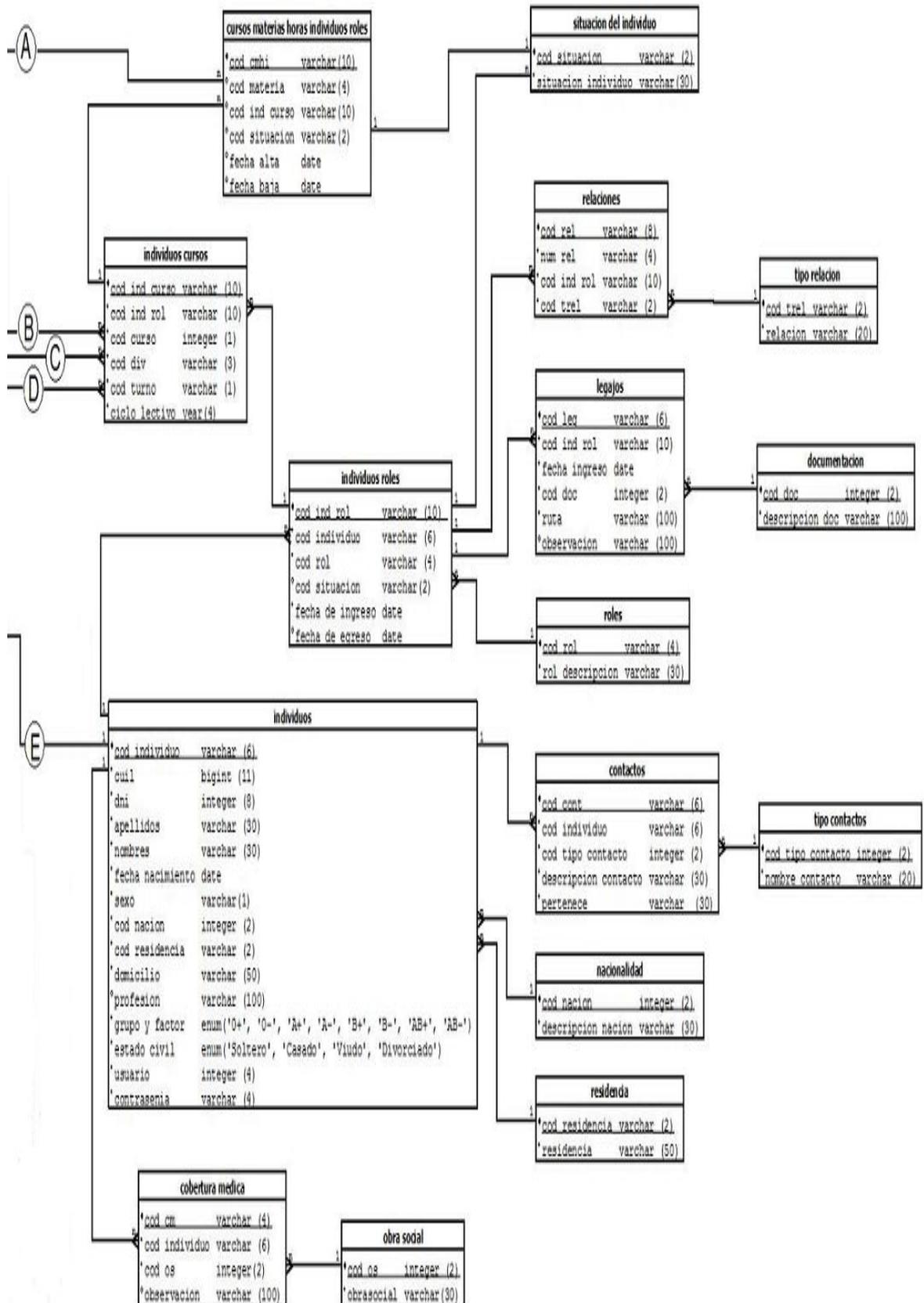
En el proceso completar inscripción, el preceptor puede ver la documentación adeudada por los alumnos y también puede cambiarlos de división y turno.

La parte directiva del establecimiento escolar, puede cargar los datos personales, dar de alta en el/los cargos, modificar los datos ingresados incorrectamente del personal y dar de baja al personal.

El preceptor puede cargar los horarios, modificarlos y verlos, la dirección y los profesores solo pueden verlos y los alumnos ver su horario de clase.

### 3.5.2.2 MODELO DE DATOS





### 3.5.2.3 INTERFAZ DE USUARIOS

#### Historia 1: Completar inscripción.

Tarea 1: Informar documentación adeudada.

Para ver la documentación adeudada por el alumno, el preceptor ingresara en el MENÚ PRECEPTOR, seleccionara CONTROL INSCRIPCIÓN, aparecerá el submenú CONTROL, en donde seleccionara DOCUMENTACIÓN ADEUDADA y aparecerá la pantalla DOCUMENTACIÓN NO ENTREGADA.

Control
<a href="#">Requisitos</a>
<a href="#">Documentación adeudada</a>
<a href="#">Asignar División</a>
<input type="button" value="Volver"/>

Documentación No Entregada	
<u>Preceptor/a:</u> PONCE MARINA	Fecha: 13/09/2016 Hora: 17:24
<u>Apellido y Nombre:</u> ADA AZUL <u>D.N.I.:</u> 66666666 <u>C/D/T:</u> 1M	
<u>Apellido y Nombre:</u> ADA VERDE <u>D.N.I.:</u> 77777777 <u>C/D/T:</u> 1M	
<u>Apellido y Nombre:</u> ARIAS ANA SOL <u>D.N.I.:</u> 45454545 <u>C/D/T:</u> 1M Certificado Analítico Primaria Certificado de Buena Salud Certificado Bucodental Certificado Vista Certificado Oído	
<u>Apellido y Nombre:</u> ADA CELESTE <u>D.N.I.:</u> 88888888 <u>C/D/T:</u> 1M	
<u>Apellido y Nombre:</u> ADA NEGRA <u>D.N.I.:</u> 99999999 <u>C/D/T:</u> 1M	
<input type="button" value="Imprimir"/> <input type="button" value="Salir"/>	

### Tarea 2: Cambiar división y turno.

En el MENÚ PRECEPTOR el preceptor seleccionara CAMBIAR DIVISIÓN Y TURNO, y se verá la pantalla CAMBIAR DIVISION, donde deberá ingresar el D.N.I. del alumno y finalmente se abrirá la pantalla para cambiar la división y turno del alumno.

<b>ALUMNO/A:</b> ARIAS ANA SOL	Fecha: 14/09/2016 Hora: 16:37
Curso: 1er. Año División: <input type="text" value="Primera"/> Turno: <input type="text" value="Mañana"/>	
<input type="button" value="Guardar"/> <input type="button" value="Volver"/> <input type="button" value="Salir"/>	

### Historia 2: Altas, bajas y modificaciones del personal.

#### Tarea 1: Registrar datos del personal del establecimiento.

La parte directiva del establecimiento escolar, para ingresar los datos del personal deberá ingresar al MENÚ DIRECCIÓN, elegir DATOS DEL PERSONAL, e ingresar el DNI del personal.

<b>Busqueda de Personal</b>
Ingresar el D.N.I.: <input type="text" value="23232323"/> x
<input type="button" value="Buscar"/> <input type="button" value="Volver"/>

Si el personal no existe en la base de datos, mostrara la siguiente pantalla.

<b>Mensaje</b>
No existe en el sistema de la institución. Presione SEGUIR si desea ingresar nuevo personal.
<input type="button" value="Seguir"/> <input type="button" value="Volver"/> <input type="button" value="Salir"/>

Al presionar el botón seguir deberá completar la pantalla de PERSONAL.

### Personal

Fecha: 14/09/2016  
Hora: 17:26

Apellido/Apellidos  \*

Nombre/Nombres  \*

C.U.I.L.  \* D.N.I.  \*

Fecha de nacimiento Día  Mes  Año  \*

Sexo  Masculino  Femenino \*

Estado Civil  Soltero  Casado  Viudo  Divorciado \*

Nacionalidad  \* Residencia  \*

Dirección  \*

Título  \*

Los campos con " \* " son obligatorios

Tarea 2: Registrar contactos.

### Contactos

Personal: CARRIZO DARIO  
Fecha: 14/09/2016  
Hora: 17:30

1  Ingresar:  Pertenece a:  \*

2  Ingresar:  Pertenece a:  \*

3  Ingresar:  Pertenece a:

4  Ingresar:  Pertenece a:

5  Ingresar:  Pertenece a:

Los campos con " \* " son obligatorios

**Tarea 3: Registrar el cargo a cubrir.**

<b>Personal: CARRIZO DARIO</b>		
Cargo a cubrir	docente	▼
Situación de Revista:	titular	▼
Fecha de ingreso: Día	1	▼
Mes	1	▼
Año	2016	▼
<input type="button" value="Guardar y Seguir"/>		

**Tarea 4: Registrar foto.**

<b>Fotos</b>	
Personal: CARRIZO DARIO	Fecha: 14/09/2016 Hora: 17:37
Elija una foto para su legajo:	
C:\Users\Usuario\Pictur <input type="button" value="Examinar..."/>	
<input type="button" value="Guardar y Seguir"/>	

Por último se le asigna el número de usuario y contraseña al personal.

<b>Personal: CARRIZO DARIO</b>		
Su Nombre de usuario es: <b>1311</b>		
Su contraseña es: <b>v9Ph</b>		
<input type="button" value="Guardar"/>	<input type="button" value="Imprimir"/>	<input type="button" value="Salir"/>

### Tarea 5: Registrar el alta del personal.

Para dar de alta al personal en su/s cargos, la parte directiva deberá ingresar al MENU DIRECCION, seleccionar ALTA DEL PERSONAL y aparecerán las siguientes pantallas.

### Busqueda de Personal

Ingresar el D.N.I.:

### Personal

CARRIZO DARIO

Elija el rol a cubrir:

### Alta personal

Personal: CARRIZO DARIO Fecha: 14/09/2016  
Hora: 17:47

Rol: DOCENTE

Situación de Revista:  \*

Fecha de alta: Día  Mes  Año  \*

Curso:  División:  Turno:  \* Ciclo Lectivo:  \*

Area:

Los campos con "\*" son obligatorios

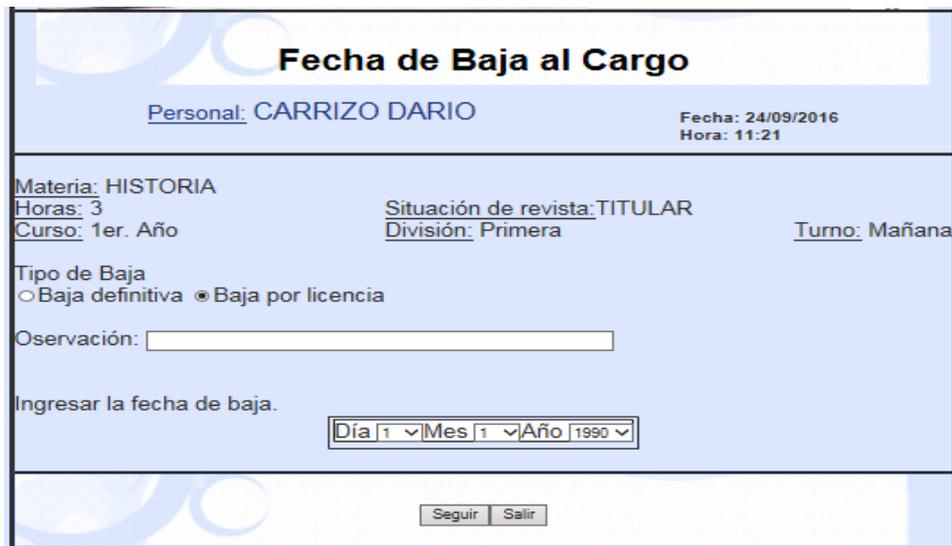
**Tarea 6: Registrar la baja del personal.**

En el MENÚ DIRECCIÓN, también puede seleccionar BAJA DEL PERSONAL se ingresa el DNI del mismo, luego muestra la pantalla CARGOS PARA DAR DE BAJA.



Materia	Horas	Curso	División	Turno	Sit.Rev.	Fecha Alta	Baja
HISTORIA	3	1er. Año	Primera	Mañana	titular	2016-01-01	<input checked="" type="checkbox"/>

Por último aparece la pantalla Fecha de Baja al Cargo.



Materia: HISTORIA  
Horas: 3  
Curso: 1er. Año  
Situación de revista: TITULAR  
División: Primera  
Turno: Mañana

Tipo de Baja  
 Baja definitiva  Baja por licencia

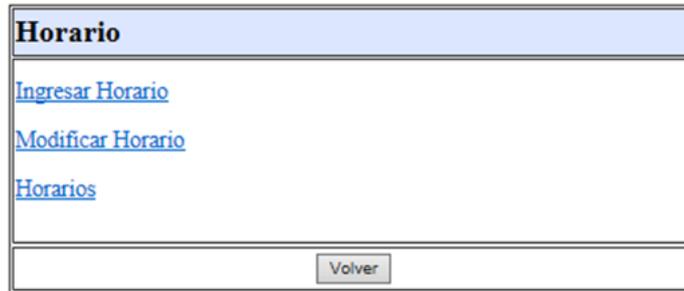
Observación:

Ingresa la fecha de baja.  
Día [1] Mes [1] Año [1990]

### **Historia 3: Registrar horario.**

#### **Tarea 1: Registrar horario.**

El preceptor al ingresar en su menú y escoger HORARIOS, aparecerá el submenú HORARIO.



The screenshot shows a web interface titled "Horario". It contains three blue hyperlinks: "Ingresar Horario", "Modificar Horario", and "Horarios". At the bottom right, there is a "Volver" button.

Si selecciona la opción INGRESAR HORARIO, deberá elegir el Turno/Curso/División, luego aparece la pantalla HORARIOS.



The screenshot shows the "Horarios" registration screen. At the top, it displays "Preceptor/a: PONCE MARINA" and "Fecha: 14/09/2016 Hora: 18:3". Below this, it shows "Turno: Mañana. Curso: 1er. Año. División: Primera." and a "Dias" dropdown menu set to "lunes". A table lists hours and subjects:

Horas	Materias *
1-Primera	BIOLOGIA
2-Segunda	BIOLOGIA
3-Tercera	MATEMATICA
4-Cuarta	MATEMATICA
5-Quinta	LENGUA Y LITERATURA
6-Sexta	LENGUA Y LITERATURA
7-Septima	LENGUA Y LITERATURA

At the bottom, there are "Guardar y Seguir" and "Volver" buttons.

## Tarea 2: Modificar horario.

También en el submenú HORARIO puede elegir la opción MODIFICAR HORARIO, seleccionar Turno/Curso/División, luego la Materia y se mostrara la pantalla HORA A CAMBIAR, y después la pantalla HORARIOS.

### Hora a Cambiar

Preceptor/a: PONCE MARINA Fecha: 14/09/2016  
Hora: 18:19

---

Curso: 1er. Año. División: Primera. Turno: Mañana  
Materia: BIOLOGIA

Seleccionar el día/hora/turno que desea cambiar de la materia

Día	Hora	Cambiar
martes	6° hora	<input type="radio"/>
lunes	1° hora	<input checked="" type="radio"/>
lunes	2° hora	<input type="radio"/>

### Horarios

Preceptor/a: PONCE MARINA Fecha: 14/09/2016  
Hora: 18:21

---

Curso: 1er. Año. División: Primera. Turno: Mañana  
Materia: BIOLOGIA

Seleccionar el día/hora

Día  Hora

**Tarea 3: Informar horario**

En el submenú HORARIO también se podrá seleccionar HORARIOS, luego seleccionar Turno/Curso/División, donde se puede ver el horario del curso seleccionado, y también imprimirlo. Desde el menú Dirección se selecciona horario, al igual que en el menú Docente y Alumno en el caso de los alumnos estos solo pueden ver su horario.

Preceptor/a: PONCE MARINA

---

Seleccionar Turno/Curso/Division

---

Horarios

Preceptor/a: PONCE MARINA Fecha: 14/09/2016  
Hora: 23:6

Turno:Mañana. Curso:1er. Año. Division: Primera.

Hora	Lunes	Martes	Miercoles	Jueves	Viernes
1	FISICA- QUIMICA <small>(cuello jorge)</small>	HISTORIA <small>(Carrizo Dario)</small>	TECNOLOGIA <small>(perea davis)</small>	LENGUA EXTRANJERA <small>(carrizo sergio)</small>	FORMACION ETICA Y CIUDADANA <small>(roger etel)</small>
2	FISICA- QUIMICA <small>(cuello jorge)</small>	HISTORIA <small>(Carrizo Dario)</small>	TECNOLOGIA <small>(perea davis)</small>	LENGUA EXTRANJERA <small>(carrizo sergio)</small>	FORMACION ETICA Y CIUDADANA <small>(roger etel)</small>
3	MATEMATICA <small>(roger etel)</small>	MATEMATICA <small>(roger etel)</small>	TECNOLOGIA <small>(perea davis)</small>	LENGUA EXTRANJERA <small>(carrizo sergio)</small>	EDUCACION ARTISTICA MUSICA <small>(perea davis)</small>
4	MATEMATICA <small>(roger etel)</small>	FISICA- QUIMICA <small>(cuello jorge)</small>	HISTORIA <small>(Carrizo Dario)</small>	LENGUA Y LITERATURA <small>(cuello gabriela)</small>	MATEMATICA <small>(roger etel)</small>
5	LENGUA Y LITERATURA <small>(cuello gabriela)</small>	BIOLOGIA <small>(cuello jorge)</small>	GEOGRAFIA <small>(Carrizo Dario)</small>	LENGUA Y LITERATURA <small>(cuello gabriela)</small>	MATEMATICA <small>(roger etel)</small>
6	LENGUA Y LITERATURA <small>(cuello gabriela)</small>	BIOLOGIA <small>(cuello jorge)</small>	GEOGRAFIA <small>(Carrizo Dario)</small>	LENGUA Y LITERATURA <small>(cuello gabriela)</small>	FORMACION ETICA Y CIUDADANA <small>(roger etel)</small>
7	EDUCACION ARTISTICA MUSICA <small>(perea davis)</small>	BIOLOGIA <small>(cuello jorge)</small>	GEOGRAFIA <small>(Carrizo Dario)</small>	EDUCACION ARTISTICA MUSICA <small>(perea davis)</small>	FORMACION ETICA Y CIUDADANA <small>(roger etel)</small>

### 3.5.3 DESARROLLO DEL TERCER PROTOTIPO

La entrega de esta versión, viene a completar un 80% de la funcionalidad del producto. La descripción, de esta versión se encuentra en la Tabla 3.5.

*Tercer Prototipo:* en este prototipo se lleva a cabo el registro de las notas, realizadas por el docente o el preceptor; los registros de novedades que lo realiza el preceptor o la dirección; la conformación de las mesas de examen llevadas a cabo por la dirección; la inscripción de exámenes que puede ser realizada por el alumno, el tutor o el preceptor; y por último el registro de exámenes por el docente.

PROTOTIPO	ITERACIONES	HISTORIAS DE USUARIO	TAREAS
3	1	1 Registrar notas.	1.- Registrar notas de la materia.
			2.- Informar calificaciones registradas.
			3.- Registrar notas de conducta.
	2	2 Registrar novedades	1.- Registrar novedades
			2.- Informar novedades.
	3	3 Registrar exámenes.	1.- Generar mesa de examen.
			2.- Inscripción de examen.
			3.- Registrar exámenes.

**Tabla 3.5 Planificación del Tercer Prototipo**

#### 3.5.3.1 DESCRIPCION DE FUNCIONALIDAD

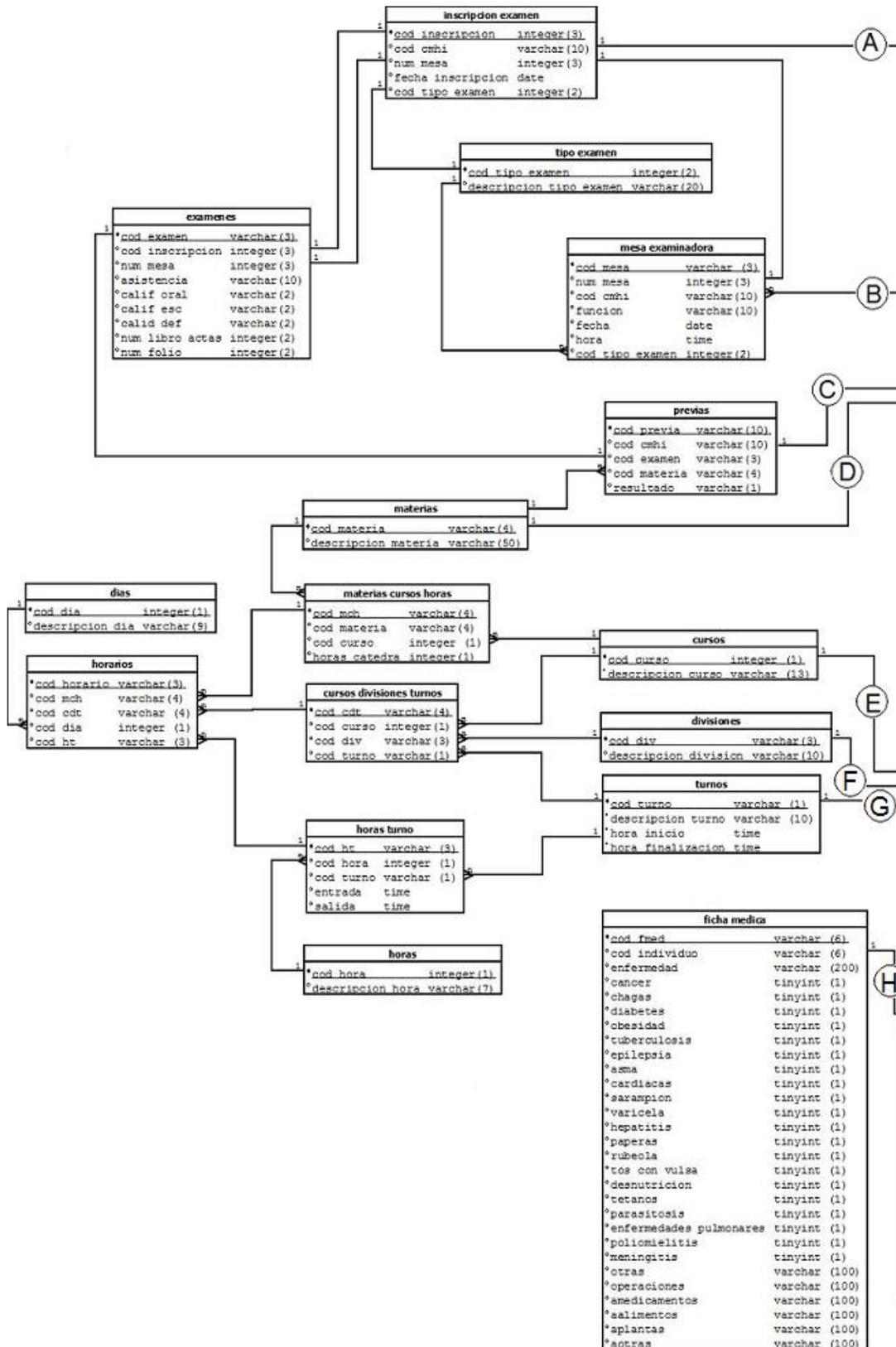
En la Tabla 3.5 se observa la tercera versión del producto de software. Esta incluye: un diseño de interfaz que soporta el registro de notas de las materias dictadas cuya carga la pueden realizar el docente o el preceptor, emitiendo finalmente un informe de las calificaciones; las notas de conducta las registra el preceptor.

En cuanto al registro de las novedades lo podrá realizar el preceptor, estableciendo el tipo de novedad, la fecha de inicio y la fecha final, la cantidad y la observación que se desea realizar. También muestra un informe con las novedades, previa selección del tipo de novedad, la fecha de inicio y fecha final a consultar. Dirección también puede registrar y ver informe de novedades. El alumno y el docente solo pueden ver sus novedades.

Con respecto a la tercera historia, registrar exámenes, la dirección realiza la primera tarea generar las mesas de exámenes. La inscripción de examen, la puede realizar el alumno, el tutor o el preceptor y luego se emite el permiso de examen.

El registro de los exámenes tomados lo realizara el docente, registrando el número de libro de actas, el número de folio, la asistencia, la calificación escrita, oral, y definitiva del alumno examinado, luego se podrá imprimir el acta volante del examen.

### 3.5.3.2 MODELO DE DATOS





### 3.5.3.3 INTERFAZ DE USUARIO

#### Historia 1: Registrar notas.

##### Tarea 1: Registrar notas de la materia

El docente, en su menú podrá seleccionar NOTAS DE LOS ALUMNOS, luego la materia, después el curso división y turno, finalmente podrá registrar las notas en la pantalla NOTAS DE LOS ALUMNOS.

**MATERIAS**

Profesor/a: CARRIZO DARIO Fecha: 16/09/2016  
Hora: 21:27

---

Materia/s:

---

**CURSOS**

Profesor/a: CARRIZO DARIO Fecha: 16/09/2016  
Hora: 21:29

---

Materia: HISTORIA  
Seleccione el curso en el que quiere ingresar notas

---

**Notas de los Alumnos**

Profesor/a: CARRIZO DARIO Fecha: 16/09/2016  
Hora: 21:32

---

Materia: HISTORIA  
 Curso: 1er. Año División: Primera Turno: Mañana  
 Evaluación: Primer Trimestre

N°	Alumno	Trimestres			Calificación	Compensación		Calificación Final
		1°	2°	3°		1°	2°	
1	ADA CELESTE	3						
2	ADA VERDE	3						
3	ARIAS ANA SOL	3						

---

**Tarea 2:** Informar calificaciones registradas.

Luego se genera el informe de las calificaciones registradas.

**Escuela Secundaria N° 44**  
**"Jorge Luis Borges"**



---

Profesor/a: CARRIZO DARIO

Fecha: 16/09/2016  
Hora: 21:34

---

Materia: HISTORIA  
Curso: 1er. Año División: Primera Turno: Mañana  
Evaluación: Primer Trimestre

N°	Alumno	Trimestres			Calificación	Compensación		Calificación Final
		1º	2º	3º		1º	2º	
1	ADA CELESTE	3						
2	ADA VERDE	9						
3	ARIAS ANA SOL	6						

Prof. CARRIZO DARIO

Imprimir   Guardar   Salir

El preceptor también podrá registrar las notas, al seleccionar desde su menú la opción NOTAS DE LOS ALUMNOS, luego debe elegir el curso, división y turno, después materia, una vez elegida, se mostrara la pantalla NOTAS DE LOS ALUMNOS.

**Tarea 3:** Registrar notas de conducta.

Para registrar las notas de conducta el preceptor ingresara a su menú, seleccionara NOTAS DE CONDUCTA y se abre la pantalla CURSO A CARGO, se elige el curso, luego se abre la pantalla NOTAS DE CONDUCTA, donde se pueden ingresar las mismas o modificar si ya hubiese notas.

CURSOS A CARGO

Preceptor/a: PONCE MARINA

Fecha: 16/09/2016  
Hora: 21:38

---

Seleccione el curso del que quiere ver las notas

Curso	División	Turno
1er. Año	Primera	Mañana <input type="radio"/>

Seguir   Volver

## Notas de Conducta

Preceptor/a: PONCE MARINA Fecha: 16/09/2016  
Hora: 21:40

---

Notas de Conducta  
Curso: 1er. Año División: Primera Turno: Mañana  
Evaluación: Primer Trimestre

N°	Alumno	Trimestres		
		1°	2°	3°
1	ADA AZUL			
2	ADA VERDE			
3	ADA CELESTE			
4	ARIAS ANA SOL			

### **Historia 2: Registrar novedades.**

#### **Tarea 1: Registrar novedades.**

En cuanto al registro de novedades lo podrá realizar el preceptor, al ingresar en su menú, seleccionar NOVEDADES, se abre el submenú NOVEDADES.

### Novedades

---

[Alumnos](#)

[Personal](#)

---

De esta ventana se puede seleccionar ALUMNOS O PERSONAL, en ambos casos se ingresara el DNI, si el individuo no existe le avisa de su inexistencia, y cuando existe, le muestra la pantalla siguiente, donde podrá registrar las novedades, estableciendo la fecha de inicio, la cantidad de días, la fecha final y la observación.

**ALUMNO/A: ARIAS ANA SOL D.N.I. N°:45454545**

Fecha: 16/09/2016  
Hora: 21:46

Seleccione novedad

Seleccione:

Fecha de Inicio	Dia	Mes	Año	Fecha Final	Dia	Mes	Año
	1	1	1990		1	1	1990

Novedades *	Fecha de Inicio			Fecha Final			Cantidad *	Observacion
	Dia	Mes	Año	Dia	Mes	Año		
<input type="text" value="Inasistencias Justificadas"/>	1	09	2016	1	09	2016		

Los campos con "\*" son obligatorios

**Escuela Secundaria N° 44**  
**"Jorge Luis Borges"**


**Alumno/a: ARIAS ANA SOL**

Comprobante de Novedad

Novedad	Curso	Division	Turno	Fecha Inicio	Cantidad	Fecha Finalizacion	Observacion
Inasistencias Justificadas	1er. Año	Primera	Mañana	1.09.2016	3	1.09.2016	

Dirección también puede registrar novedades si en su menú, selecciona NOVEDADES.

### Tarea 2: Informar novedades.

El preceptor en la pantalla NOVEDADES, al presionar el botón informe previa selección de la fecha podrá ver la siguiente pantalla.

**Escuela Secundaria N° 44**  
**"Jorge Luis Borges"**



**Alumno/a: ARIAS ANA SOL**

---

**Informe de Novedades**

Novedades	Curso	División	Turno	Fecha de Inicio	Cantidad	Fecha de Finalización	Observación
Inasistencias Justificadas	1er. Año	Primera	Mañana	2016-09-01	3	2016-09-01	

---

Dirección puede ver el informe de novedades si en su menú, selecciona NOVEDADES. El alumno y el docente solo pueden ver sus novedades ingresando a sus respectivos menús y seleccionando NOVEDADES.

### Historia 3: Registrar exámenes.

#### Tarea 1: Generar mesa de examen.

La dirección puede generar las mesas de exámenes, al ingresar a su menú, seleccionar MESAS EXAMINADORAS; luego el turno, curso y división; después la materia y por último aparece la pantalla TRIBUNAL PARA EXAMEN, se establece el tipo de examen, la fecha, hora y los integrantes del tribunal.

**Curso Para Examen**

**Dirección:** INFANTE HECTOR **Fecha:** 16/09/2016  
**Hora:** 21:53

Seleccione el curso para examen

**Turno/Curso/División:**

### Materia Para Examen

Dirección: INFANTE HECTOR Fecha: 16/09/2016  
Hora: 21:54

Curso: 1er. Año División: Primera Turno: Mañana

Seleccione la Materia para Examen

FORMACION ETICA Y CIUDADANA	<input type="radio"/>
LENGUA EXTRANJERA	<input type="radio"/>
LENGUA Y LITERATURA	<input type="radio"/>
MATEMATICA	<input type="radio"/>
EDUCACION FISICA	<input type="radio"/>
EDUCACION ARTISTICA MUSICA	<input type="radio"/>
HISTORIA	<input type="radio"/>
GEOGRAFIA	<input checked="" type="radio"/>
BIOLOGIA	<input type="radio"/>
FISICA-QUIMICA	<input type="radio"/>
TECNOLOGIA	<input type="radio"/>

Seguir Volver Salir

### Tribunal Para Examen

Dirección: INFANTE HECTOR Fecha: 16/09/2016  
Hora: 21:56

Mesa Examinadora Número: 4

Materia: GEOGRAFIA Tipo de Examen: Regular

Curso: 1er. Año División: PRIMERA Turno: MAÑANA

Fecha: Día 1 Mes 10 Año 2016

Hora: 8 Minutos 0

**TRIBUNAL**  
Presidente: CARRIZO DARIO  
Vocal 1: CUELLO JORGE Vocal 2: ROGER ETEL

Seguir Volver Salir

#### Tarea 2: Inscripción de examen.

La inscripción de examen, la puede realizar el alumno, o el tutor, al seleccionar en su menú, INSCRIPCIÓN A EXAMEN, se abre la pantalla INSCRIPCIÓN A EXAMEN, selecciona la materia a rendir y luego se emite el permiso de examen que se observa en la pantalla PERMISO DE EXAMEN. El preceptor también podrá realizar la inscripción de examen del alumno, seleccionando en su menú, INSCRIPCIÓN A EXAMEN, luego deberá ingresar el DNI del alumno y aparecerá la pantalla INSCRIPCIÓN A EXAMEN, al seleccionar la materia a rendir se emite el permiso de examen.

### Inscripción a Examen

Alumno/a: ARIAS JOSE

Fecha: 19/05/2016  
Hora: 19:6

Seleccione la/s materia/s a rendir

Materia	Condición	C/D/T	Ciclo	Fecha	Hora	Presidente del Tribunal	Rendir
LENGUA Y LITERATURA	Previa	1 I M	2015	2016-05-20	10:00:00	CUELLO GABRIELA	<input type="checkbox"/>

### Permiso de Exámen

Escuela Secundaria N° 44 Jorge Luis Borges

N° de Permiso: 136

Conste por la presente que el/la alumno/a ARIAS JOSE  
esta habilitado/a para rendir la/s asignatura/s, que se indican a continuación.

N° de Orden	Materia	Condición	Curso	Fecha	Calificación	Firma del Presidente de Mesa
1	LENGUA Y LITERATURA	Previa	1-I-m	2016-05-20		

Miraflores: 20/05/2016

Firma \_\_\_\_\_ Sello \_\_\_\_\_

**Nota:**  
 1) Para poder rendir exámen el alumno deberá presentar a la mesa examinadora este permiso y su documento de identidad  
 2) Los exámenes escritos deben ser hechos con tinta.

**Tarea 3: Registrar exámenes.**

El registro de los exámenes tomados lo realizara el docente, al ingresar en su menú, seleccionar EXAMEN, la materia, el curso, división, turno, y en la pantalla EXAMEN, se registrara el examen, luego se podrá imprimir el acta volante del examen que se puede ver en la pantalla ACTA VOLANTE DE EXAMENES.

**MATERIAS**

Profesor/a: CUELLO GABRIELA

Fecha: 20/05/2016  
Hora: 18:5

Materia/s: LENGUA Y LITERATURA

**Examen**

Profesor/a: CUELLO GABRIELA

Fecha: 20/05/2016  
Hora: 19:11

Materia: LENGUA Y LITERATURA  
Seleccione el curso

Curso	División	Turno
1er. año	Primera	Mañana

**Examen**

Profesor/a: CUELLO GABRIELA

Fecha: 20/05/2016  
Hora: 19:18

Asignatura: LENGUA Y LITERATURA

Curso/División/Turno: 1er. año-Primera-Mañana

Tipo de Exámen: Previa-

Libro de Actas N°: 15 \* Folio N°: 03 \*

N° Orden	N° Permiso	N° Documento	Apellidos y Nombres	Asistencia	Calificación Escrita	Calificación Oral	Calificación Definitiva
1	330	77222222	ARIAS JOSE	P	7	7	7

Asistencia: "A" para los AUSENTES y "P" para los PRESENTES.  
Calificación: "-" para las calificaciones no tomadas.

Presidente: CUELLO GABRIELA  
Vocal: ROGER ETEL  
Vocal: GONZALEZ JORGE CARLOS

Los campos con "\*" son obligatorios

### Acta Volante de Exámenes

**ESCUELA SECUNDARIA N° 44  
"JORGE LUIS BORGES"**

Asignatura: LENGUA Y LITERATURA

Curso/División/Turno: 1er. año-Primera-Mañana

Tipo de Exámen: Previa-

Libro de Actas N°: 15 Folio N°: 03

N° Orden	N° Permiso	N° Documento	Apellidos y Nombres	Calificación Escrita	Calificación Oral	Calificación Definitiva
1	330	77222222	ARIAS JOSE	7	7	7

Asistencia: "A" para los AUSENTES y "P" para los PRESENTES.  
Calificación: "-" para las calificaciones no tomadas.

Total de alumnos: 1
Aprobados: 1
Aplazados: 0
Ausentes: 0

Fecha: 21/05/2016  
 Presidente: CUELLO GABRIELA \_\_\_\_\_  
 Vocal: ROGER ETEL \_\_\_\_\_  
 Vocal: GONZALEZ JORGE CARLOS \_\_\_\_\_

### 3.5.3.4 CRÍTICAS Y SUGERENCIAS

La Tabla 3.6 reúne las críticas y sugerencias expresadas por las autoridades de la escuela en la etapa de evaluación del prototipo, esta permitirá observar nuevos requisitos o los objetivos que posibilitaran la resolución de las deficiencias de esta tercera entrega.

NUEVOS REQUISITOS	OBJETIVOS DE LOS REQUISITOS
Emitir listados de mesas examinadoras, para dirección, preceptores, docentes, alumnos y tutores.	Mejorar la información de las mesas examinadoras.
Imprimir listado de alumnos con previas por materia.	Mejorar la información de los alumnos con materias previas.
Consultar los resultados de los exámenes.	Optimizar la entrega de los resultados de exámenes.

Tabla 3.6 Nuevos Requisitos

### 3.5.4 DESARROLLO DEL CUARTO PROTOTIPO

El cuarto prototipo representa una versión totalmente operativa del sistema. Esta versión cuenta con toda la capacidad de funcionamiento pretendida para el sistema. Las características de esta versión cubren tanto los requisitos generales planteados por las autoridades de la escuela en la fase de exploración, como también los requisitos y/o planteamientos que fueron surgiendo durante el proceso de desarrollo del prototipo (ver Tabla 3.7).

*Cuarto Prototipo: Abarca las funcionalidades específicas relacionadas con la emisión de las libretas, certificados y la registración de cooperadora de los alumnos.*

PROTOTIPO	ITERACIONES	HISTORIAS DE USUARIO	TAREAS
4	1	1 Emisión de informes, certificados y constancias.	1.- Emitir libreta.
			2.- Emitir informe de alumnos por curso.
			3.- Emitir informe mesas examen.
			4.- Emitir informe examen.
			5.- Emitir informe previas.
			6.- Emitir certificación de servicio
			7.- Emitir constancia de docente.
			8.- Emitir constancia de alumno regular.
	2	2 Registrar cooperadora.	1.- Registrar pagos de cooperadora.
			2.- Emitir informe de pagos existentes.

**Tabla 3.7 Planificación del Cuarto Prototipo**

#### 3.5.4.1 DESCRIPCION DE FUNCIONALIDAD

En la Tabla 3.7 se observa la cuarta versión del producto de software. Esta incluye: un diseño de interfaz que soporta el proceso de emisión de informes, certificados y constancias; el registro del pago de cooperadora y el informe de pago, además de los requerimientos planteados por las autoridades de la escuela en el prototipo anterior.

La libreta puede ser visualizada, guardada e imprimida por el preceptor, el tutor o el alumno.

La dirección y el preceptor pueden ver el informe de las mesas de exámenes que se generaron, también el docente puede ver la fecha de los exámenes a tomar.

El informe de exámenes tomados lo podrá ver dirección, el preceptor y el docente; el alumno junto con su tutor pueden ver los exámenes aprobados.



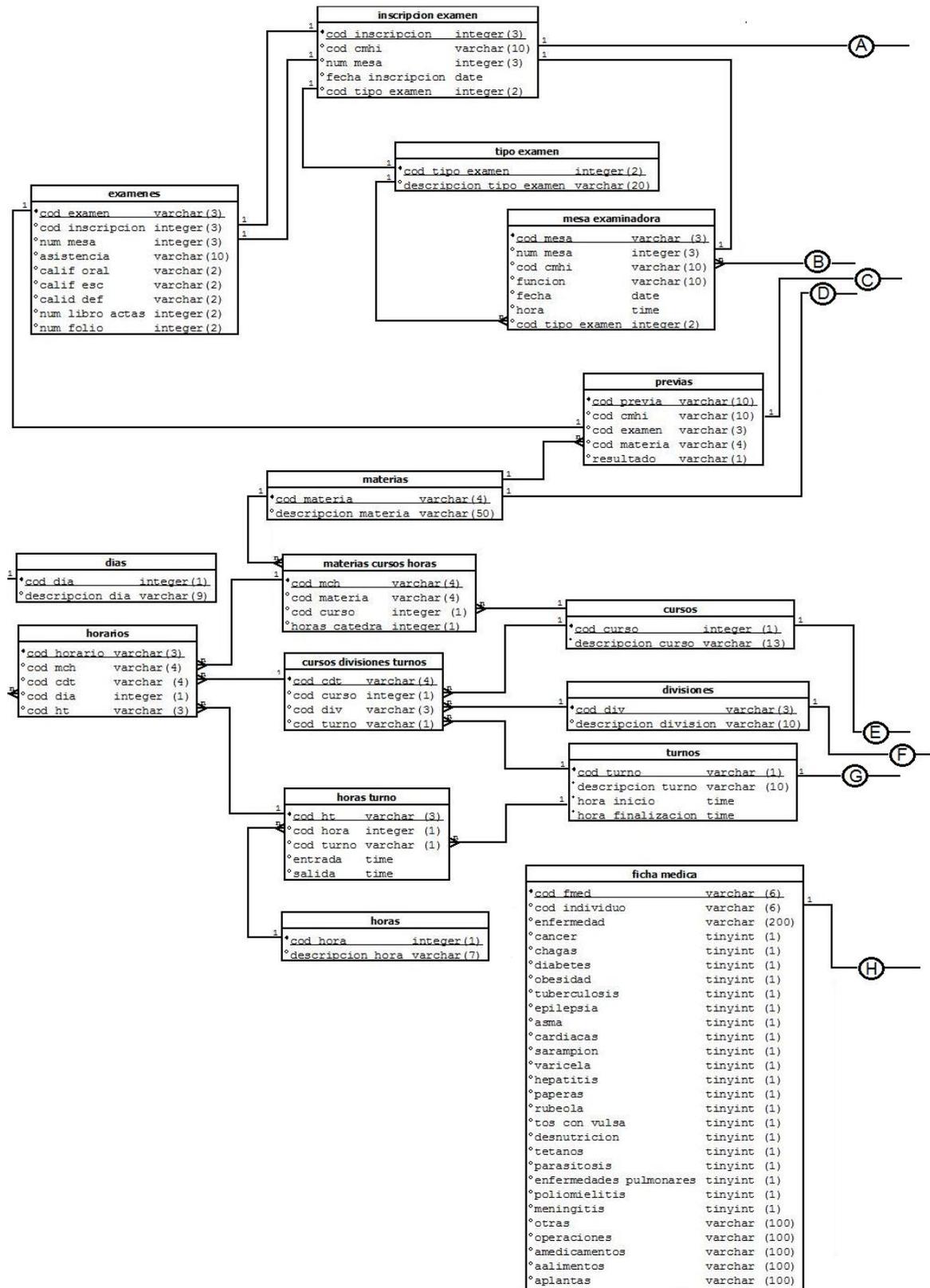
La emisión de los informes de previas la puede ver el preceptor, el profesor ver los alumnos que tienen previa en su materia y el alumno las materias previas que tiene.

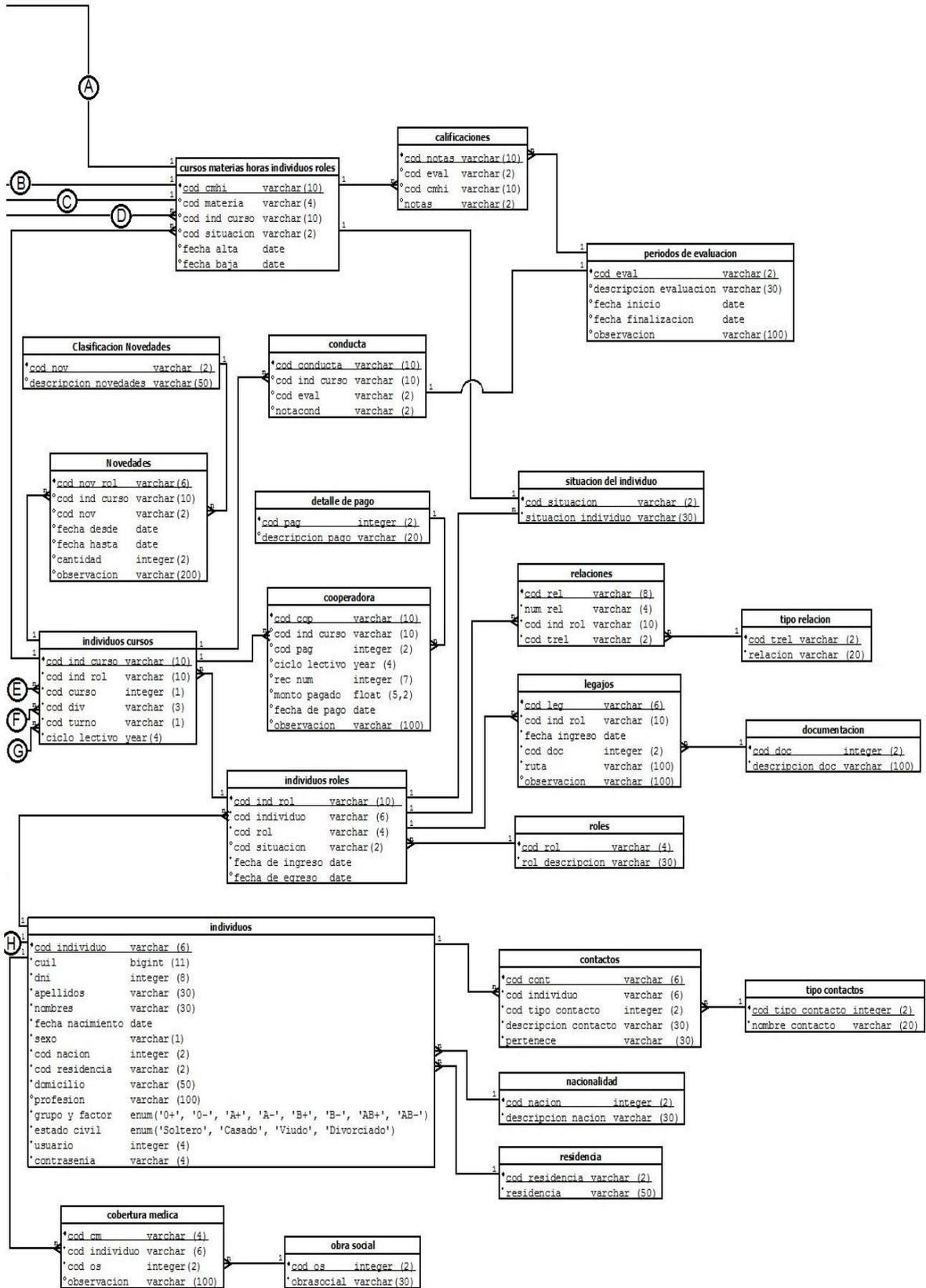
Tanto el preceptor y dirección pueden ver un listado de alumnos por curso, el docente solo puede visualizar el listado de sus alumnos por curso.

La emisión de certificados de servicio, constancia de docente y constancia de alumno regular lo puede hacer el preceptor y dirección.

El registro y emisión de recibo de pago de cooperadora lo realiza el preceptor, el alumno y su tutor pueden ver el informe de los pagos efectuados.

### 3.5.4.2 MODELO DE DATOS

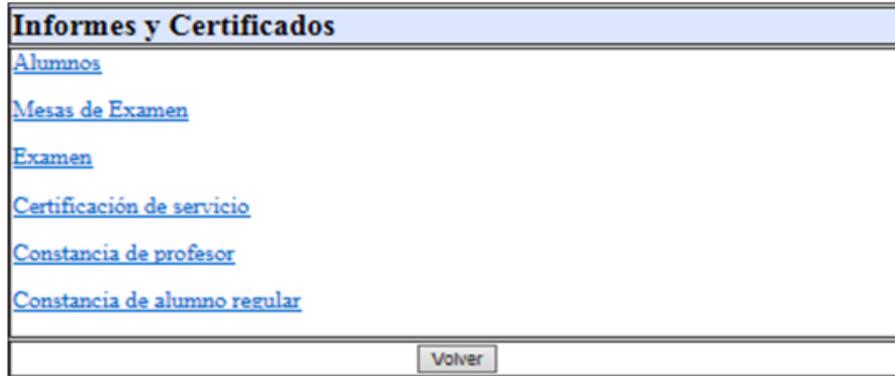




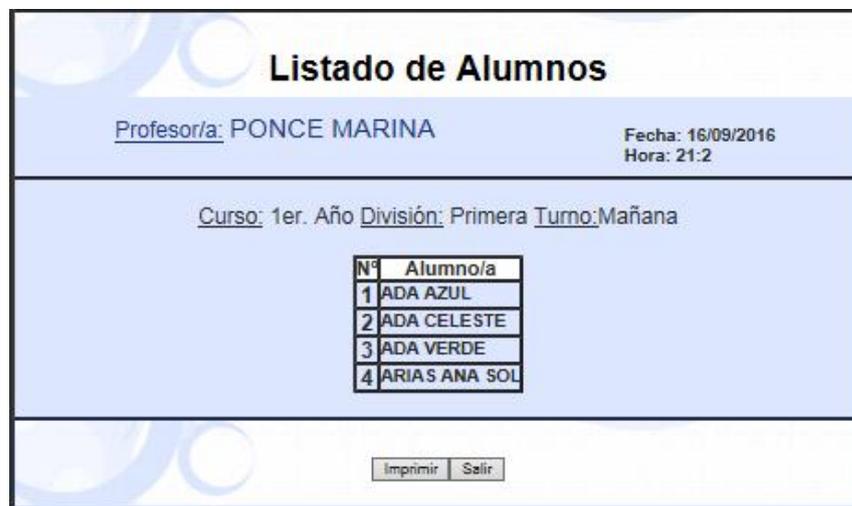


### Tarea 2: Emitir informe de alumnos por curso.

Para ver el listado de alumnos por curso, el preceptor deberá ingresar a su menú, seleccionar INFORMES Y CERTIFICADOS, se abrirá el submenú INFORME Y CERTIFICADOS.



Al seleccionar ALUMNOS y luego el turno, curso, división, se abrirá la pantalla LISTADO DE ALUMNOS.



Dirección también puede ver el listado de alumnos por curso si en su menú selecciona ALUMNOS y luego el turno, curso, división. El docente solo puede ver el listado de sus alumnos si ingresa a su menú, selecciona ALUMNOS, luego la materia y después turno, curso y división.

### Tarea 3: Emitir informe mesas examen.

En el caso de la emisión del informe de las mesas de examen, el preceptor deberá seleccionar MESAS DE EXAMEN, se ingresara fecha de inicio y fecha final, aparecerá la pantalla MESAS DE EXAMEN. Desde dirección se ingresara en el MENÚ DIRECCIÓN, elegirá INFORME DE MESAS DE EXAMEN, los pasos siguientes son similares a los del preceptor. El docente puede ver sus exámenes, al ingresar al MENÚ DOCENTE,

seleccionara MESAS EXAMINADORA, introduciendo la fecha de inicio y fecha final y se abrirá la pantalla MESAS DE EXAMEN.

**Preceptor/a:** PONCE MARINA      Fecha: 23/09/2016  
Hora: 20:51

Seleccione Fecha de Inicio y Finalización de las mesas de examen

Fecha de Inicio	Día	Mes	Año	Fecha Final	Día	Mes	Año
	1	1	2013		1	1	2013

Ver Informe    Borrar    Volver

### Mesas de Examen

Materia	Curso/División/Turno	Fecha	Hora	Tribunal
MATEMATICA	1er. Año Primera Mañana	2016-10-01	08:00:00	ROGER ETEL CUELLO GABRIELA
LENGUA Y LITERATURA	1er. Año Primera Mañana	2016-10-01	08:00:00	CUELLO GABRIELA CUELLO JORGE CARRIZO DARIO
LENGUA EXTRANJERA	1er. Año Primera Mañana	2016-10-01	08:00:00	CARRIZO SERGIO CUELLO JORGE CUELLO GABRIELA

Imprimir    Salir

#### Tarea 4: Emitir informe examen.

Para visualizar el informe de exámenes, el preceptor seleccionara EXÁMEN, se observara la pantalla en donde debe elegir las fechas de inicio y la fecha de finalización y finalmente aparece la pantalla EXAMEN. La dirección también lo puede ver al ingresar al MENÚ DIRECCIÓN seleccionar INFORME DE EXAMENES, los pasos siguientes son los mismos del preceptor.

**Preceptor/a:** PONCE MARINA      Fecha: 16/09/2016  
Hora: 21:4

Seleccione Fecha de Inicio y Finalización de los exámenes

Fecha de Inicio	Día	Mes	Año	Fecha Final	Día	Mes	Año
	1	1	2013		1	1	2013

Ver Informe    Borrar    Volver

## Examen

Preceptor/a: PONCE MARINA Fecha: 26/09/2016  
Hora: 20:58

Materia	C/D/T	Fecha	Apellido y Nombre	Nota	N° de Libro	N° de Folio
MATEMATICA	1/m	2016-10-01	Arias Ana Sol	7	3	34

Para que el docente pueda ver los resultados de sus exámenes tomados deberá ingresar a su menú, seleccionar INFORME DE EXÁMENES y se verá la siguiente pantalla.

## Examen

Profesor/a: CUELLO GABRIELA Fecha: 28/05/2016  
Hora: 18:58

Materia	C/D/T	Fecha	Apellido y Nombre	Nota
LENGUA Y LITERATURA	1/m	2016-05-20	ARIAS JOSE	7

El alumno y el tutor podrán ver los exámenes aprobados, al ingresar en el MENÚ ALUMNO, seleccionar EXÁMENES y verá la pantalla que está a continuación.

## Examen

Alumno/a: ARIAS JOSE Fecha: 28/05/2016  
Hora: 18:38

Materia	C/D/T	Fecha	Nota
LENGUA Y LITERATURA	1/m	2016-05-20	7

### Tarea 5: Emitir informe previas.

Para emitir informes de previas, el preceptor ingresa al MENU PRECEPTOR, selecciona PREVIAS, luego selecciona el turno, curso y división, luego la materia, y aparece el listado de los alumnos con previas, el cual puede imprimir. El docente ingresa al MENU DOCENTE, selecciona ALUMNOS CON PREVIAS, luego selecciona, la materia, curso, división, y turno,

mostrando el informe de los alumnos con previas. Y el alumno ingresa al MENU ALUMNO selecciona PREVIAS y muestra las materias previas que tiene.

### Previas

Preceptor/a: PONCE MARINA Fecha: 23/09/2016  
Hora: 22:7

Seleccione el curso Turno/Curso/División:

### Previa

Preceptor/a: PONCE MARINA Fecha: 23/09/2016  
Hora: 22:8

Curso: 1er. Año División: Primera Turno: Mañana

Seleccione la Materia

FORMACION ETICA Y CIUDADANA	<input type="checkbox"/>
LENGUA EXTRANJERA	<input checked="" type="checkbox"/>
LENGUA Y LITERATURA	<input type="checkbox"/>
MATEMATICA	<input type="checkbox"/>
EDUCACION FISICA	<input type="checkbox"/>
EDUCACION ARTISTICA MUSICA	<input type="checkbox"/>
HISTORIA	<input type="checkbox"/>
GEOGRAFIA	<input type="checkbox"/>
BIOLOGIA	<input type="checkbox"/>
FISICA-QUIMICA	<input type="checkbox"/>
TECNOLOGIA	<input type="checkbox"/>

### Previas

Preceptor/a: PONCE MARINA Fecha: 23/09/2016  
Hora: 22:9

**ALUMNOS CON PREVIA**

**Materia: LENGUA EXTRANJERA**

**Curso/División/Turno: 1er. Año Primera Mañana**

**Nro. Apellidos y Nombres | Ciclo**

En los casos de la emisión de certificados de servicio, constancia de docente y constancia de alumno regular los pasos para el preceptor son, ingresar al MENU PRECEPTOR, seleccionar INFORMES Y CERTIFICADOS, se abre el submenú INFORMES Y CERTIFICADOS, puede seleccionar certificación de servicio, constancia de profesor o

alumno regular, ingresar el D.N.I. y se visualiza certificado o constancia la cual se puede imprimir. En el caso de dirección que también puede emitirlos ingresan al MENU DIRECCION, seleccionan CERTIFICADOS Y CONSTANCIAS, ingresa el D.N.I., y se abre la pantalla donde se visualiza o se puede imprimir la constancia o certificado seleccionado.

**Busqueda de Personal**

Ingresar el D.N.I.:

Tarea 6: Emitir certificación de servicio.



**ESCUELA SECUNDARIA "Jorge Luis Borges" N° 44**

**CERTIFICACION DE SERVICIO**

La Autoridad que suscribe - hace constar que el Prof. CARRIZO SERGIO D.N.I.: 12121212 presta/o servicio en este establecimiento de acuerdo a los siguientes detalles.

ESPACIO CURRICULAR / CARGO	CURSO	SECCION	SIT. REV.	CANT. HS.	DESDE	HASTA	MODALIDAD	OBSERVACION
LENGUA EXTRANJERA	1er. Año	Primera	titular	3	1990-01-01	Continua	CICLO BASICO	<input style="width: 100px;" type="text"/>

Se expide la presente, en Miraflores - Dpto. Capayán, a los días 16 del mes de Septiembre del año 2016, para ser presentado ante las autoridades que la requieran.

Sello

---

 Firma y aclaración de autoridad

**Tarea 7: Emitir constancia de docente.**

Constancia de Personal
<p>La Dirección de la Escuela Secundaria N° "Jorge Luis Borges" dependiente de la Dirección Educación Secundaria, hace constar que El/La PROF. CARRIZO SERGIO D.N.I.: 12121212 en el día de la fecha [_____].</p> <p>A pedido del interesado se extiende la presente constancia en Miraflores - Dpto. Capayán, a los días 16 del mes de Septiembre del año 2016, para ser presentado ante las autoridades que la requieran.</p>
<p style="text-align: center;"><input type="button" value="Seguir"/> <input type="button" value="Volver"/> <input type="button" value="Salir"/></p>

 <b>ESCUELA SECUNDARIA N° 44 "Jorge Luis Borges"</b>
<b><u>CONSTANCIA DE PROFESOR</u></b>
<p>La Dirección de la Escuela Secundaria N° "Jorge Luis Borges" dependiente de la Dirección Educación Secundaria, hace constar que El/La Profesor/a. CARRIZO SERGIO D.N.I.: 12121212 en el día de la fecha participo de reunión institucional.</p> <p>A pedido del interesado se extiende la presente constancia en Miraflores - Dpto. Capayán, a los días 16 del mes de Septiembre del año 2016, para ser presentado ante las autoridades que la requieran.</p>
<p>Sello _____ Firma y aclaración de autoridad _____</p>
<p><input type="button" value="Imprimir"/> <input type="button" value="Guardar"/> <input type="button" value="Salir"/></p>

**Tarea 8:** Emitir constancia de alumno regular.

Busqueda de alumno/a
Ingresar el D.N.I. del alumno/a: <input type="text"/>
<input type="button" value="Buscar"/> <input type="button" value="Volver"/>

 <i>ESCUELA SECUNDARIA "Jorge Luis Borges" N° 44</i>  <b><u>CONSTANCIA DE ALUMNO REGULAR</u></b>  Se hace constar que ARIAS ANA SOL D.N.I.: 45454545 es alumno/a regular de 1er. Año Año "Primera" de este establecimiento.  Se extiende la presente constancia en Miraflores - Departamento Capayán, a los días 16 del mes de Septiembre del año 2016, para ser presentado ante las autoridades que la requieran.  Sello <span style="margin-left: 200px;">_____</span> Firma y aclaración de autoridad  <input type="button" value="Imprimir"/> <input type="button" value="Guardar"/> <input type="button" value="Salir"/>
--

**Historia 2:** Registrar cooperadora.**Tarea 1:** Registrar pagos de cooperadora.

Para registrar el pago cooperadora, en el MENÚ PRECEPTOR, el preceptor selecciona COOPERADORA, ingresa el DNI del alumno, si el individuo existe, muestra la pantalla COOPERADORA, donde podrá seleccionar la cuota de cooperadora a registrar, estableciendo el ciclo, el monto y la observación luego se generara el recibo de pago, con su respectivo número de recibo, el cual podrá imprimirlo, y también puede consultar el informe de pago del alumno estableciendo fecha de inicio y fecha final, e imprimirlo.

**Preceptor/a: PONCE MARINA**

Ingresar el D.N.I. del ALUMNO/A:

**ALUMNO/A: ARIAS ANA SOL** Fecha: 16/09/2016  
 D.N.I. N°: 45454545 Hora: 20:15

Seleccione	Fecha de Inicio			Fecha Final		
	Dia	Mes	Año	Dia	Mes	Año
	1	1	2016	1	1	2016

Pagos *	Ciclo *	Monto *	Recibo Nro.	Observacion
Pago total	2016		6818512	

Los campos con "\*" son obligatorios

Escuela Secundaria N° 44   
 "Jorge Luis Borges"

**Alumno/a: ARIAS ANA SOL**

Fecha de Pago: 16/09/2016

**RECIBO N°: 6818512**

Descripcion del Pago	Año	Monto	Observacion
Inscripcion	2016	550	

**Tarea 2:** Emitir informe de pagos existentes.

**Escuela Secundaria N° 44**  
**“Jorge Luis Borges”**



**Informe de Cooperadora**  
 Alumno/a: **ARIAS ANA SOL**

Detalle de Pago	Ciclo	Curso	División	Turno	Monto	Recibo	Fecha	Observacion
Inscripcion	2016	1er. Año	Primera	Mañana	50.00	6818512	2016-09-16	
Marzo	2016	1er. Año	Primera	Mañana	5.00	6449879	2016-09-16	
Abril	2016	1er. Año	Primera	Mañana	5.00	3714483	2016-09-16	

**3.5.4.4 CRÍTICAS Y SUGERENCIAS**

La tabla 3.8 reúne las sugerencias expresadas por las autoridades de la escuela en la etapa de evaluación del cuarto prototipo, emitiendo nuevos requisitos que resuelven las deficiencias de esta cuarta entrega.

NUEVOS REQUISITOS	OBJETIVOS DE LOS REQUISITOS
Periodos de evaluación	Asignar periodos de evaluación.
Datos personales del docente.	Permitir actualizar los datos personales del docente.
Alta por cese de licencia.	Completar lo concerniente a la licencia de los docentes.

**Tabla 3.8 Nuevos Requisitos**

### 3.5.4.5 MODIFICACION DEL CUARTO PROTOTIPO

De acuerdo a las críticas y sugerencias propuestas por las autoridades de la escuela (ver tabla 3.8) en la presentación del cuarto prototipo, se llevaron a cabo las siguientes tareas:

Dirección asigna los periodos de evaluación, registrando las fechas de inicio y finalización de cada periodo.

El docente podrá actualizar sus datos personales ingresando a su menú y seleccionando datos personales.

Cuando un docente que se encontraba de licencia vuelve a la actividad, se debe registrar su alta, esto lo lleva a cabo dirección.

### 3.5.4.6 INTERFAZ DE USUARIO MODIFICACION DEL CUARTO PROTOTIPO

Dirección asigna los **periodos de evaluación** correspondiente al ciclo lectivo actual en la pantalla que se muestra a continuación a la cual se accede por el MENU DIRECCION, al seleccionar PERIODOS DE EVALUACION.



Periodos	Fecha de Inicio			Fecha Final		
	Dia	Mes	Año	Dia	Mes	Año
1° Trimestre	1	1	2013	1	1	2013
2° Trimestre	1	1	2013	1	1	2013
3° Trimestre	1	1	2013	1	1	2013
1° Compensatorio	1	1	2013	1	1	2013
2° Compensatorio	1	1	2013	1	1	2013
Calificación Final	1	1	2013	1	1	2013

El docente puede modificar sus **datos personales**, al ingresar a su menú y seleccionar DATOS PERSONALES.



**Personal**

Fecha: 16/09/2016  
Hora: 20:29

Apellido/Apellidos  \*  
Nombre/Nombres  \*  
C.U.I.L. 121212121 D.N.I. 12121212  
Fecha de nacimiento 1970-01-01  
Sexo  Masculino  Femenino \*  
Estado Civil  Soltero  Casado  Viudo  Divorciado \*  
Nacionalidad: Argentina Residencia:  \*  
Rol/es:  \*  
Fecha de ingreso: 1990-01-01 Fecha de egreso: 0000-00-00  
Dirección  \*  
Título  \*

Los campos con "\*" son obligatorios

Desde dirección se puede dar de **alta por cese de licencia**, al seleccionar ALTA POR CESE DE LICENCIA, ingresar el D.N.I. del personal, luego el rol a dar de alta, se abrirá la Pantalla CARGOS PARA DAR DE ALTA y se seleccionara el cargo.

**Busqueda de Personal**

Ingresar el D.N.I.:

**Personal**

Elija el rol para alta de CARRIZO SERGIO:

**CARGOS PARA DAR DE ALTA**

Personal: CARRIZO SERGIO Fecha: 16/09/2016  
Hora: 20:40

Seleccionar cargo a dar de alta.

Materia	Horas	Curso	División	Turno	Sit.Rev.	Fecha Alta	Alta
LENGUA EXTRANJERA	3	2do. Año	Primera	Mañana	suplente con licencia	1990-01-01	<input type="radio"/>
LENGUA EXTRANJERA	3	1er. Año	Primera	Mañana	titular	1990-01-01	<input type="radio"/> Continúa

# CAPÍTULO IV

---

## **Análisis de Resultados y Conclusiones**

#### 4.1 RESULTADOS ALCANZADOS

Se desarrolló un sistema de información para brindarle a la Escuela Secundaria N° 44 “Jorge Luis Borges” una herramienta que le permita optimizar la gestión administrativa de la institución, automatizando las tareas para disminuir la carga de trabajo respecto a los controles que se realizan en forma manual (como inscripción de alumno, control de cumplimiento de requisitos, inscripción de examen, generación de actas de examen, altas y bajas de docentes, emisión de recibos, certificados, planillas, libretas, etc.).

Se creó una base de datos relacionales con un servidor MySQL que se deberá instalar en el servidor del aula de informática, de la escuela, esta base de datos es de fácil uso, de múltiples plataformas, rápida, segura, que genera confianza para quien la utiliza, para registrar, actualizar y consultar información. La misma permite centralizar la información y eliminar la duplicación de datos, que provoca problemas importantes, en la forma de registración de notas de los alumnos, entre otros.

El desarrollo del sistema se hizo aplicando la Metodología Ágil XP, esta permitió la comunicación y retroalimentación continua entre el equipo de desarrollo y las autoridades de la escuela, las entregas pequeñas del sistema ayudo a que estos conozcan el estado de avance del proyecto y a los programadores realizar un buen trabajo y tomar decisiones acertadas, de esta manera se pudo avanzar rápidamente y de forma efectiva, realizando solo la documentación necesaria y lograr un software fácil de usar, por su simplicidad, y que cumple con todos los requisitos planteados por las autoridades de la escuela. La elaboración de este sistema mejora el acceso a los datos, para la obtención de información oportuna para la toma de decisiones.

Al sistema desarrollado que tiene el nombre de *Sistema de Gestión Administrativa para la Escuela Secundaria N°44 “Jorge Luis Borges”*, se lo puede acceder a través de la WEB, el mismo brinda los siguientes servicios:

- Registrar inscripción y reinscripción de alumno.
- Controlar requisitos de inscripción.
- Asignar el curso, la división y el turno al alumno.
- Registrar baja de alumno.
- Registrar alta, baja y modificación del personal.
- Registrar, y modificar los horarios de clases.
- Registrar las notas de los alumnos.
- Registrar mesas de exámenes e inscripción a examen.
- Registrar actas de examen
- Registrar novedades del alumno y del docente como inasistencias, sanciones, y licencias.
- Registrar el pago de cooperadora y emitir recibo de pago.
- Generar planillas y libretas de calificaciones, constancias de alumno regular, constancias del docente, certificación de servicios, etc.
- Consultar el listado de alumnos que pertenecen a un curso determinado



- Consultar datos personales de los alumnos, tutores, y del personal del establecimiento educativo.
- Consultas de asistencias de alumnos, y del personal del establecimiento educativo.
- Consultas de pagos cooperadora.
- Consultas de notas de alumnos.
- Consultas de materias previas, de mesa examinadora.
- Consultas de horarios de clase.

En general, es importante resaltar que se cumplió con todos los objetivos planteados al inicio del proyecto, obteniendo como resultado una aplicación que se ajusta a los requerimientos de las autoridades de la escuela.

## 4.2 CONCLUSIONES

El actual movimiento que se observa en la escuela (aumento de la matrícula de los alumnos, del personal de la escuela, de los avances de la tecnología) requiere que la escuela cuente con un sistema de gestión, con la finalidad de optimizar las tareas del sector administrativo y mejorar la comunicación con toda la comunidad educativa. Gracias al sistema desarrollado los directivos y los preceptores pueden saber con exactitud el listado de los alumnos inscriptos, a que sección y turno pertenecen; si entregaron toda la documentación que se requiere. Los docentes podrán registrar y enviar las notas, en tiempo y en forma. Los alumnos y tutores podrán consultar sobre el desempeño escolar, las materias previas, y las mesas de exámenes.

En cuanto a la metodología ágil XP que se empleó para el desarrollo del software fue acertada, porque se aplicó a un proyecto como este, pequeño, con requisitos imprecisos y cambiantes, con un equipo de trabajo de pocos integrantes.

La metodología XP, permitió mantener una estrecha relación con las autoridades de la escuela para desarrollar una aplicación de forma controlada, eficaz y que además fue de gran ayuda para manejar el cambio, en tiempos razonables, logrando interactuar con los usuarios reiteradas veces.

La programación en pareja nos benefició en tener una menor tasa de errores, mejor diseño y una mayor satisfacción como programadores.

El análisis relacionado con el tiempo que se invirtió en la elaboración del producto es el tiempo planificado y estimado en la fase de la planificación, difiere del definitivo, esta divergencia deja ver la poca experiencia en la estimación del esfuerzo y el tiempo por parte del equipo de desarrollo y se consideró que para el aprovechamiento de la aplicación del método XP, se debe contar con una buena disciplina y experiencia en el proceso y las practicas que este involucra.

El Sistema de Gestión Administrativa para la Escuela Secundaria N°44 “Jorge Luis Borges”, fue aprobado por las autoridades del establecimiento escolar, aceptándolo, quedando conformes, y también se lo mostro a los alumnos, futuros usuarios, explicándoles cómo



funciona el sistema, permitiéndoles navegar en el mismo, habiendo quedado entusiasmados, motivados y deseando que pronto sea implementado.

La puesta en funcionamiento del sistema queda sujeta a la autorización del ministerio de educación de la provincia de Catamarca.

## REFERENCIAS

- [1]. Letelier P. & M<sup>a</sup> C. Penadés. (2006). Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP). Técnica Administrativa vol. 5 N°26, Buenos Aires abril/junio 2006. ISSN 1666-1680. Recuperado de: <http://www.cyta.com.ar/ta0502/v5n2a1.htm> Accedida el 12 de Abril de 2015.
- [2]. Metodología ágil de desarrollo de software extremo (AMS XP) y software libre (OSS) de Villegas Adriana. Recuperado de: <http://www.monografias.com/trabajos67/metodologia-desarrollo-sofwares/metodologia-desarrollo-sofwares.shtml>. Accedida el 13 de Abril de 2015.
- [3]. Decreto 459/2010. Buenos Aires 6 de Abril de 2010 [http://www.wp-ontent/themes/conectar\\_igualdad/pdf/decreto.pdf](http://www.wp-ontent/themes/conectar_igualdad/pdf/decreto.pdf). Accedida 21 de Marzo de 2015.
- [4]. Giraldo, Z., J. Gómez (2006) Metodologías Ágiles. Características de las Metodologías Ágiles. EcuRed. Cuba. Recuperado de: [https://www.ecured.cu/Metodolog%C3%ADas\\_Agiles](https://www.ecured.cu/Metodolog%C3%ADas_Agiles) Accedida el 15 de Abril.
- [5]. Historia de la programación extrema (2008). Recuperado de: <https://ingenieriadefsoftware.blogia.com/2008/090201-historia-de-la-programacion-extrema.php> Accedida el 15 de Abril.
- [6]. Joskowicz, J. (2008). Reglas y prácticas de eXtreme Programming. Instituto de Ingeniería Eléctrica, Facultad de Ingeniería Universidad de la República, Montevideo, URUGUAY. Recuperado de <https://iie.fing.edu.uy/~josej/docs/XP%20-%20Jose%20Joskowicz.pdf>. Accedida el 13 de Abril de 2015.
- [7]. Calabria L. & P. Piriz (2003). Metodología XP. Cátedra de Ingeniería de Software de la Universidad ORT Uruguay. Recuperado de [http://fi.ort.edu.uy/innovaportal/file/2021/1/metodologia\\_xp.pdf](http://fi.ort.edu.uy/innovaportal/file/2021/1/metodologia_xp.pdf). Accedida el 15 de Abril de 2015.
- [8]. Castro, M. J. (2006). Caracterización del desempeño de las metodologías ágiles en el desarrollo de un producto de software pequeño. Universidad de los Andes, Mérida Venezuela, Abril de 2006. <http://tesispost.serbi.ula.ve/harvester/index.php/search/results?query=evaluaci%F3n&archiveIds%5B%5D=7&isAdvanced=&searchPage=3>. Accedida 10 de Abril de 2015.
- [10]. Torres, R. (2009). Metodologías Ágiles para Desarrollo de Software eXtreme Programming (xp), Mérida Julio 2009. Recuperado de <http://www.metologa-agiles-desarrollo-software-xp>. Accedida el 15 de Abril de 2015.



## BIBLIOGRAFÍA

Priolo Sebastián. Métodos Ágiles. Desarrolladores USERS. Gradi S.A. 1° edición. 2009.

Rozic Sergio Ezequiel. Bases de datos. Manuales USERS. code. MP Ediciones. S.A. 2004.

Monso Martín Ramón. Programación PHP. Manuales USERS. MP Ediciones. S.A. 2004.

Minera Francisco. PHP6 sitios dinámicos con lenguaje más robusto. Edición Fox Andina en coedición con Grandi S.A. 2010.

Beck, Kent. "Extreme Programming Explained. Embrace Change", Pearson Education, 1999. Traducido al español como: "Una explicación de la programación extrema. Aceptar el cambio", Addison Wesley, 2000.

Fernández Escribano, Gerardo. Introducción a Extreme Programing, Ingeniería de software II., 9 de diciembre de 2002.

<[http://www.ort.edu.uy/fi/ingenieria/SubSitios/ingsoft/investigacion/ayudantias/metodologia\\_xp.pdf](http://www.ort.edu.uy/fi/ingenieria/SubSitios/ingsoft/investigacion/ayudantias/metodologia_xp.pdf)>. Accedida el 15 de Abril de 2015.

Letelier Torres, Patricio; Sánchez López Emilio A. Actas Metodologías Ágiles en el Desarrollo del software. Alicante – España 12 de Noviembre de 2003. <<http://www.issi.dsic.upv.es/archives/f-1069167248521/actas.pdf>>. Accedida el 13 de Abril de 2015.

Extreme Programming. <[http://www.cs.umss.edu.bo/doc/material/mat\\_gral\\_130/Exposicion-Xp.ppt](http://www.cs.umss.edu.bo/doc/material/mat_gral_130/Exposicion-Xp.ppt)> Accedida el 13 de Abril de 2015.

CASTRO, Mario J. Caracterización del desempeño de las metodologías ágiles en el desarrollo de un producto de software pequeño. Venezuela 2006. <<http://tesispost.serbi.ula.ve/harvester/index.php/search/results?query=evaluaci%F3n&archivelds%5B%5D=7&isAdvanced=&searchPage=3>>. Accedida 10 de Abril de 2015.

REINOSO SALAZAR, Ender Alberto; MENDOZA BLANCO, Yeraldine. Sistema para el manejo hotelero bajo ambiente LINUX integrado con aplicaciones en MS-DOS para el Hotel Altamira Suites. Universidad Católica Andrés Bello Facultad de Ingeniería. Escuela de Ingeniería Informática 2006.

TORRES, Ramón. Metodologías Ágiles para Desarrollo de Software eXtreme Programming (xp), Mérida Julio 2009. <<http://www.metologa-agiles-desarrollo-software-xp>>. Accedida el 15 de Abril de 2015.

# ANEXOS

---

## ANEXO I: ESTÁNDARES DE PROGRAMACIÓN

En la programación se estableció un conjunto de estándares para hacer el código legible y fácil de mantener por cualquier miembro del equipo.

Algunas normas fueron:

- Escribir el código en minúscula, solo se empleó mayúscula en lo que correspondería a textos que visualiza el usuario en las pantallas del sistema.
- Se utilizó siempre el mismo formato de sangría.
- Al escribir objetos se utilizaron prefijos por ejemplo los nombres de botones con el prefijo btn.
- Para las variables también se utilizó prefijos, con el objetivo de identificar el tipo de dato que contiene. Por ejemplo una variable de tipo texto se codificó con el prefijo txt.
- Para las variables que corresponden a claves primarias de la base de datos se usó el prefijo cod; y las demás variables con un nombre consistente y similar al utilizado en la escuela.
- También se comentó el código cuando fue necesario.

Ejemplo de código donde se observa la tabulación y nombre de variables usados.

```
24 | </tr>
25 | <form action="procesaralumno.php" method="POST">
26 | <tr>
27 | <td>
28 | <table border="0" cellspacing="0" align="center">
29 | <tr>
30 | <td colspan=2>
31 | <br>Apellido/Apellidos
32 | <input style="width:450;" type="text" name="txtapellido" value="">
33 | <font color="#FF0000"> * </font>
34 | </td>
35 | </tr>
36 | <tr>
37 | <td colspan=2>
38 | Nombre/Nombres
39 | <input style="width:450;" type="text" name="txtnombre" value="">
40 | <font color="#FF0000"> * </font>
41 | </td>
42 | </tr>
43 | <tr>
44 | <td>
```

### Ejemplo de código donde se observa la utilización de prefijo para botones

```
155 <table border="0" cellspacing="0" align="center">
156 <tr>
157 <td>
158 <input type="submit" size="11" name="btnsubmit" value="Guardar y Seguir" >
159 </td>
160 <td>
161 <input type="reset" name="btnreset" value="Borrar">
162 </td>
163 <td>
164 <input type="submit" name="btnsalir" value="Salir" >
165 </td>
166 </tr>
167 </table>
```

### Ejemplo de prefijo de variable que contiene campo clave de un valor de la base de datos.

```
12 if (isset($_POST['btnmodificar']))
13 {
14     SESSION_start();
15     $apellidos=$_SESSION['apellidos'];echo $apellidos; echo "<br>";
16     $nombres=$_SESSION['nombres'];echo $nombres; echo "<br>";
17     $cod_individuo=$_SESSION['cod individuo'];echo $cod_individuo;echo "<br>";
18     include ("parametrosconexion.php");
19     $conexion = mysql_connect( $servidor, $usuario, $contrasenia );
20     $cod_codindrol=$_SESSION['cod ind rol'];echo $cod_codindrol;echo "<br>";
21     $codmat=$_SESSION ['cod materia'];echo $codmat;echo "<br>";
22     $codcurso=$_SESSION['cod curso'];echo $codcurso;echo "<br>";
23     $coddiv=$_SESSION['cod div'];echo $coddiv;echo "<br>";
24     $codturno=$_SESSION['cod turno'];echo $codturno;echo "<br>";
25     $materia = $_SESSION['materia'];echo $materia;echo "<br>";
```

### Ejemplo de comentarios realizados en el código fuente

```
81 -}
82 // Si hay errores registrados, continúa la rutina mostrando
83 // El formulario de errores
84 if (count($errores)>0)
85 {
86     //Dirección relativa de la página (formulario) donde se mostrarán los errores
87     $pagina_redireccionada = 'Location: formulario_errores.php?';
88     //Parámetros_GET
89     $parametros1 = "";
90     for( $j=0; $j<count($errores); $j++)
91     {
92         $parametros1 .= $j;
93         $parametros1 .= '=';
94         $parametros1 .= $errores[$j];
95         $parametros1 .= '&';
96     }
97     $parametros1 .= 'origen=Alumno.php';
98     //Función redirecciona
99     header($pagina_redireccionada.$parametros1);
100 }
```



## ANEXO II: HISTORIAS DE USUARIOS

Historia de Usuario 1	
<b>Nombre:</b> SOLICITUD INSCRIPCION DE ALUMNO	
<b>Usuario:</b> Alumno/Tutor/Preceptor	
<b>Prioridad:</b> Alta	<b>Prototipo:</b> 1
<b>Descripción:</b> se solicita la inscripción del alumno, si es la primera vez que se inscribe debe registrar sus datos personales y datos de su tutor.	
<b>Observaciones:</b>	

Tarea 1	
<b>Nombre tarea:</b> ALTAS, BAJAS, MODIFICACIONES DE CURSOS/DIVISIONES/TURNOS/ MATERIAS	
<b>Corresponde historia:</b> 1	<b>Tipo de tarea:</b> Desarrollo
<b>Programador responsable:</b> Equipo XP	
<b>Descripción:</b> se generan las altas, bajas y modificaciones de los datos de cursos, divisiones, turnos y materias necesarios para la inscripción del alumno.	

Tarea 2	
<b>Nombre tarea:</b> VERIFICAR EXISTENCIA DEL ALUMNO	
<b>Corresponde historia:</b> 1	<b>Tipo de tarea:</b> Desarrollo
<b>Programador responsable:</b> Equipo XP	
<b>Descripción:</b> Se accede a la base de datos ingresando el D.N.I. en caso que este registrado se le informa de su existencia y que ingrese en el sistema con su clave. En caso contrario le permite que se registre como alumno.	

Tarea 3	
<b>Nombre tarea:</b> REGISTRAR DATOS PERSONALES DEL ALUMNO	
<b>Corresponde historia:</b> 1	<b>Tipo de tarea:</b> Desarrollo
<b>Programador responsable:</b> Equipo XP	
<b>Descripción:</b> Se solicita que registre sus datos personales, nombre, apellido, DNI, cuil, sexo, estado civil, fecha de nacimiento, localidad, nacionalidad, dirección, y además el curso y el turno al que se inscribe.	

Tarea 4	
<b>Nombre tarea:</b> REGISTRAR CONTACTOS	
<b>Corresponde historia:</b> 1	<b>Tipo de tarea:</b> Desarrollo
<b>Programador responsable:</b> Equipo XP.	
<b>Descripción:</b> Se solicita ingresar el tipo de contacto, descripción y a quien pertenece.	



<b>Tarea 5</b>	
<b>Nombre tarea: REGISTRAR FICHA MEDICA</b>	
<b>Corresponde historia: 1</b>	<b>Tipo de tarea: Desarrollo</b>
<b>Programador responsable: Equipo XP</b>	
<b>Descripción: Se solicita ingresar el grupo, factor sanguíneo, cobertura y enfermedades.</b>	

<b>Tarea 6</b>	
<b>Nombre tarea: REGISTRAR DATOS DEL TUTOR</b>	
<b>Corresponde historia: 1</b>	<b>Tipo de tarea: Desarrollo</b>
<b>Programador responsable: Equipo XP</b>	
<b>Descripción: Se solicita ingresar datos personales del tutor.</b>	

<b>Tarea 7</b>	
<b>Nombre tarea: REGISTRAR FOTO DEL ALUMNO</b>	
<b>Corresponde historia: 1</b>	<b>Tipo de tarea: Desarrollo</b>
<b>Programador responsable: Equipo XP</b>	
<b>Descripción: Se solicita ingresar foto actualizada.</b>	

<b>Historia de Usuario 2</b>	
<b>Nombre: INSCRIPCION ALUMNO</b>	
<b>Usuario: Alumno/Tutor/Preceptor</b>	
<b>Prioridad: Alta</b>	<b>Prototipo: 1</b>
<b>Descripción: se inscribe o reinscribe al alumno, pudiendo modificar sus datos personales, y se le asigna el nuevo curso, división y turno. Se controla que la documentación requerida fue entregada y si no cumple con los requisitos se le da de baja.</b>	
<b>Observaciones:</b>	

<b>Tarea 1</b>	
<b>Nombre tarea: MODIFICAR DATOS PERSONALES</b>	
<b>Corresponde historia: 2</b>	<b>Tipo de tarea: Desarrollo</b>
<b>Programador responsable: Equipo XP</b>	
<b>Descripción: Permite modificar los datos personales y el sistema le asigna el nuevo curso, división y turno.</b>	

<b>Tarea 2</b>	
<b>Nombre tarea: ASIGNAR CURSO, DIVISION Y TURNO</b>	
<b>Corresponde historia: 2</b>	<b>Tipo de tarea: Desarrollo</b>
<b>Programador responsable: Equipo XP</b>	
<b>Descripción: Se solicita ingresar el curso, división y turno del alumno.</b>	



<b>Tarea 3</b>	
<b>Nombre tarea: REGISTRAR DOCUMENTACION REQUERIDA</b>	
<b>Corresponde historia: 2</b>	<b>Tipo de tarea: Desarrollo</b>
<b>Programador responsable: Equipo XP</b>	
<b>Descripción:</b> Permite modificar los datos personales, y el sistema le asigna el nuevo curso, división y turno.	

<b>Tarea 4</b>	
<b>Nombre tarea: REGISTRAR BAJA DEL ALUMNO</b>	
<b>Corresponde historia: 2</b>	<b>Tipo de tarea: Desarrollo</b>
<b>Programador responsable: Equipo XP</b>	
<b>Descripción:</b> Se le da de baja al alumno por no cumplir los requisitos.	

<b>Historia de Usuario 3</b>	
<b>Nombre: COMPLETAR INSCRIPCION</b>	
<b>Usuario:</b> Preceptor	
<b>Prioridad: Alta</b>	<b>Prototipo: 2</b>
<b>Descripción:</b> se finaliza la inscripción del alumno, controlando la documentación adeudada y en donde también se puede cambiar división y turno.	
<b>Observaciones:</b>	

<b>Tarea 1</b>	
<b>Nombre tarea: INFORMAR DOCUMENTACION ADEUDADA</b>	
<b>Corresponde historia: 3</b>	<b>Tipo de tarea: Desarrollo</b>
<b>Programador responsable: Equipo XP</b>	
<b>Descripción:</b> Se observa un listado de alumnos con la documentación no entregada.	

<b>Tarea 2</b>	
<b>Nombre tarea: CAMBIAR DIVISION Y TURNO</b>	
<b>Corresponde historia: 3</b>	<b>Tipo de tarea: Desarrollo</b>
<b>Programador responsable: Equipo XP</b>	
<b>Descripción:</b> Se registra el cambio de la división y del turno de un alumno determinado.	

<b>Historia de Usuario 4</b>	
<b>Nombre: ALTAS, BAJAS y MODIFICACIONES DEL PERSONAL</b>	
<b>Usuario:</b> Personal Directivo	
<b>Prioridad : Alta</b>	<b>Prototipo: 2</b>
<b>Descripción:</b> se registra altas, bajas y modificación del personal, en los diferentes cargos del establecimiento.	
<b>Observaciones:</b>	



<b>Tarea 1</b>	
<b>Nombre tarea: REGISTRAR DATOS DEL PERSONAL DEL ESTABLECIMIENTO</b>	
<b>Corresponde historia: 4</b>	<b>Tipo de tarea: Desarrollo</b>
<b>Programador responsable: Equipo XP</b>	
<b>Descripción:</b> Se registra datos del personal, cuando no existe, y si existe puede modificar sus datos personales.	

<b>Tarea 2</b>	
<b>Nombre tarea: REGISTRAR CONTACTOS</b>	
<b>Corresponde historia: 4</b>	<b>Tipo de tarea: Desarrollo</b>
<b>Programador responsable: Equipo XP</b>	
<b>Descripción:</b> Se solicita ingresar el tipo de contacto, descripción y a quien pertenece.	

<b>Tarea 3</b>	
<b>Nombre tarea: REGISTRAR EL CARGO A CUBRIR</b>	
<b>Corresponde historia: 4</b>	<b>Tipo de tarea: Desarrollo</b>
<b>Programador responsable: Equipo XP</b>	
<b>Descripción:</b> Se solicita ingresar el cargo a cubrir por parte del personal.	

<b>Tarea 4</b>	
<b>Nombre tarea: REGISTRAR FOTO</b>	
<b>Corresponde historia: 4</b>	<b>Tipo de tarea: Desarrollo</b>
<b>Programador responsable: Equipo XP</b>	
<b>Descripción:</b> Se solicita ingresar foto actualizada.	

<b>Tarea 5</b>	
<b>Nombre tarea: REGISTRAR EL ALTA DEL PERSONAL</b>	
<b>Corresponde historia: 4</b>	<b>Tipo de tarea: Desarrollo</b>
<b>Programador responsable: Equipo XP</b>	
<b>Descripción:</b> Se solicita ingresar situación de revista, fecha, curso, división, turno, ciclo lectivo y el área a cubrir.	

<b>Tarea 6</b>	
<b>Nombre tarea: REGISTRAR LA BAJA DEL PERSONAL</b>	
<b>Corresponde historia: 4</b>	<b>Tipo de tarea: Desarrollo</b>
<b>Programador responsable: Equipo XP</b>	
<b>Descripción:</b> Se solicita seleccione el cargo a dar de baja.	



Historia de Usuario 5	
<b>Nombre: REGISTRAR HORARIO</b>	
<b>Usuario:</b> Preceptor	
<b>Prioridad :</b> Alta	<b>Prototipo:</b> 2
<b>Descripción:</b> se selecciona turno curso y división, luego el día y las materias y se registra el horario, también se puede modificar el mismo.-	
<b>Observaciones:</b>	

Tarea 1	
<b>Nombre tarea: REGISTRAR HORARIO</b>	
<b>Corresponde historia:</b> 5	<b>Tipo de tarea:</b> Desarrollo
<b>Programador responsable:</b> Equipo XP	
<b>Descripción:</b> Se registra el horario por día.	

Tarea 2	
<b>Nombre tarea: MODIFICAR HORARIO</b>	
<b>Corresponde historia:</b> 5	<b>Tipo de tarea:</b> Desarrollo
<b>Programador responsable:</b> Equipo XP	
<b>Descripción:</b> Se registra la modificación del horario, seleccionando la materia y la hora.	

Tarea 3	
<b>Nombre tarea: INFORMAR HORARIO</b>	
<b>Corresponde historia:</b> 5	<b>Tipo de tarea:</b> Desarrollo
<b>Programador responsable:</b> Equipo XP	
<b>Descripción:</b> Se visualiza el horario del curso seleccionado.	

Historia de Usuario 6	
<b>Nombre: REGISTRAR NOTAS</b>	
<b>Usuario:</b> Docente/Preceptor	
<b>Prioridad:</b> Alta	<b>Prototipo:</b> 3
<b>Descripción:</b> se solicita el ingreso de la nota de la materia, para luego quedar registrada.	
<b>Observaciones:</b>	

Tarea 1	
<b>Nombre tarea: REGISTRAR NOTAS DE LA MATERIA</b>	
<b>Corresponde historia:</b> 6	<b>Tipo de tarea:</b> Desarrollo
<b>Programador responsable:</b> Equipo XP	
<b>Descripción:</b> Se selecciona la materia, luego el curso y se registra la nota del alumno en el trimestre.	



<b>Tarea 2</b>	
<b>Nombre tarea: INFORMAR CALIFICACIONES REGISTRADAS</b>	
<b>Corresponde historia: 6</b>	<b>Tipo de tarea: Desarrollo</b>
<b>Programador responsable: Equipo XP</b>	
<b>Descripción:</b> Luego de registrada la nota, se genera el informe, en donde el docente firma para dejar constancia.	

<b>Tarea 3</b>	
<b>Nombre tarea: REGISTRAR NOTAS DE CONDUCTA</b>	
<b>Corresponde historia: 6</b>	<b>Tipo de tarea: Desarrollo</b>
<b>Programador responsable: Equipo XP</b>	
<b>Descripción:</b> se solicita el ingreso de la nota de conducta del alumno.	

<b>Historia de Usuario 7</b>	
<b>Nombre: REGISTRAR NOVEDADES</b>	
<b>Usuario:</b> Preceptor	
<b>Prioridad: Alta</b>	<b>Prototipo: 3</b>
<b>Descripción:</b> se solicita el ingreso de la novedad, estableciendo el tiempo y la cantidad.	
<b>Observaciones:</b>	

<b>Tarea 1</b>	
<b>Nombre tarea: REGISTRAR NOVEDADES</b>	
<b>Corresponde historia: 7</b>	<b>Tipo de tarea: Desarrollo</b>
<b>Programador responsable: Equipo XP</b>	
<b>Descripción:</b> se selecciona la novedad, la fecha de inicio, y la fecha final, se ingresa la cantidad y si desea la observación, quedando registrada la novedad, para luego generar el comprobante de la novedad.	

<b>Tarea 2</b>	
<b>Nombre tarea: INFORMAR NOVEDADES</b>	
<b>Corresponde historia: 7</b>	<b>Tipo de tarea: Desarrollo</b>
<b>Programador responsable: Equipo XP</b>	
<b>Descripción:</b> Se elige la novedad del alumno a consultar, estableciendo la fecha de inicio y fecha final.	



Historia de Usuario 8	
<b>Nombre: REGISTRAR EXAMENES</b>	
<b>Usuario:</b> Docente	
<b>Prioridad:</b> Alta	<b>Prototipo:</b> 3
<b>Descripción:</b> se solicita ingrese el número de actas, del folio y las calificaciones quedando registrado el examen.	
<b>Observaciones:</b>	

Tarea 1	
<b>Nombre tarea: GENERAR MESA DE EXAMEN</b>	
<b>Corresponde historia:</b> 8	<b>Tipo de tarea:</b> Desarrollo
<b>Programador responsable:</b> Equipo XP	
<b>Descripción:</b> Se selecciona el turno, curso, división, la materia, el tipo de examen, la fecha, la hora y los vocales quedando registrado la mesa de examen.	

Tarea 2	
<b>Nombre tarea: INSCRIPCION DE EXAMEN</b>	
<b>Corresponde historia:</b> 8	<b>Tipo de tarea:</b> Desarrollo
<b>Programador responsable:</b> Equipo XP	
<b>Descripción:</b> selecciona la materia a rendir y se emite el permiso de examen.	

Tarea 3	
<b>Nombre tarea: REGISTRAR EXAMENES</b>	
<b>Corresponde historia:</b> 8	<b>Tipo de tarea:</b> Desarrollo
<b>Programador responsable:</b> Equipo XP	
<b>Descripción:</b> se ingresa el número de libro de actas, el número de folio, luego la asistencia y las calificaciones del alumno quedando registrado el examen.	

Historia de Usuario 9	
<b>Nombre: EMISION DE INFORMES CERTIFICADOS Y CONSTANCIAS</b>	
<b>Usuario:</b> Preceptor/Docente/Dirección	
<b>Prioridad:</b> Alta	<b>Prototipo:</b> 4
<b>Descripción:</b> se genera informes, certificados y constancias, en algunos casos se establece la fecha de inicio y de finalización.	
<b>Observaciones:</b>	



Tarea 1	
<b>Nombre tarea: EMITIR LIBRETA</b>	
<b>Corresponde historia: 9</b>	<b>Tipo de tarea: Desarrollo</b>
<b>Programador responsable: Equipo XP</b>	
<b>Descripción:</b> se genera la libreta de un alumno determinado, especificando curso, división, turno, notas de materias, de conducta, y asistencia.	

Tarea 2	
<b>Nombre tarea: EMITIR INFORME DE ALUMNOS POR CURSO</b>	
<b>Corresponde historia: 9</b>	<b>Tipo de tarea: Desarrollo</b>
<b>Programador responsable: Equipo XP</b>	
<b>Descripción:</b> se elabora el informe con los nombres y apellidos de los alumnos por curso, seleccionando el curso, la división y el turno.	

Tarea 3	
<b>Nombre tarea: EMITIR INFORME MESAS EXAMEN</b>	
<b>Corresponde historia: 9</b>	<b>Tipo de tarea: Desarrollo</b>
<b>Programador responsable: Equipo XP</b>	
<b>Descripción:</b> se genera el informe de mesa de examen, indicando fecha de inicio y de finalización.	

Tarea 4	
<b>Nombre tarea: EMITIR INFORME EXAMEN</b>	
<b>Corresponde historia: 9</b>	<b>Tipo de tarea: Desarrollo</b>
<b>Programador responsable: Equipo XP</b>	
<b>Descripción:</b> se crea el informe de examen, indicando fecha de inicio y de finalización.	

Tarea 5	
<b>Nombre tarea: EMITIR INFORME PREVIAS</b>	
<b>Corresponde historia: 9</b>	<b>Tipo de tarea: Desarrollo</b>
<b>Programador responsable: Equipo XP</b>	
<b>Descripción:</b> se origina el informe de previas, ingresando curso, división, turno y materias.	

Tarea 6	
<b>Nombre tarea: EMITIR CERTIFICACION DE SERVICIO</b>	
<b>Corresponde historia: 9</b>	<b>Tipo de tarea: Desarrollo</b>
<b>Programador responsable: Equipo XP</b>	
<b>Descripción:</b> se plasma la certificación de servicio, ingresando el D.N.I. del docente.	



<b>Tarea 7</b>	
<b>Nombre tarea: EMITIR CONSTANCIA DE DOCENTE</b>	
<b>Corresponde historia: 9</b>	<b>Tipo de tarea: Desarrollo</b>
<b>Programador responsable: Equipo XP</b>	
<b>Descripción: se produce la constancia del docente, ingresando el D.N.I. del individuo.</b>	

<b>Tarea 8</b>	
<b>Nombre tarea: EMITIR CONSTANCIA DE ALUMNO REGULAR</b>	
<b>Corresponde historia: 9</b>	<b>Tipo de tarea: Desarrollo</b>
<b>Programador responsable: Equipo XP</b>	
<b>Descripción: se elabora la constancia del alumno regular, ingresando el D.N.I. del individuo.</b>	

<b>Historia de Usuario 10</b>	
<b>Nombre: REGISTRAR COOPERADORA</b>	
<b>Usuario: Preceptor</b>	
<b>Prioridad : Alta</b>	<b>Prototipo: 4</b>
<b>Descripción: se consigna la cooperadora del alumno, ingresando su DNI, seleccionando el tipo de pago, el ciclo, el monto y si desea la observación.</b>	
<b>Observaciones:</b>	

<b>Tarea 1</b>	
<b>Nombre tarea: REGISTRAR PAGOS DE COOPERADORA</b>	
<b>Corresponde historia: 10</b>	<b>Tipo de tarea: Desarrollo</b>
<b>Programador responsable: Equipo XP</b>	
<b>Descripción: se asienta el pago de cooperadora de un alumno determinado.</b>	

<b>Tarea 2</b>	
<b>Nombre tarea: EMITIR INFORME DE PAGOS EXISTENTES</b>	
<b>Corresponde historia: 10</b>	<b>Tipo de tarea: Desarrollo</b>
<b>Programador responsable: Equipo XP</b>	
<b>Descripción: se genera el informe de pagos existentes de la cooperadora de alumno determinado, seleccionando fecha de inicio y de finalización.</b>	



**ANEXO III: DOCUMENTACIÓN UTILIZADA POR LA ESCUELA**

**Requisitos Ingreso a 1er. Año Ciclo Lectivo 2015**

- Fotocopia D.N.I del Alumno y Tutor
- Constancia de CUIL Alumno-Tutor
- Fotocopia de Partida de Nacimiento
- Certificado de Primaria Aprobada.
- Certificados Médicos: Buena Salud, Vista y oído, Odontológico.
- 1 Carpeta Espiral
- 1 Foto 4 x 4

Inscripción: 60

**Uniforme:** Pantalón de vestir o pollera Azul  
Camisa Blanca, Corbata Azul, Zapatos oscuros.

**CONSTANCIA DE PROFESOR O MAESTRO**

FORM. 866 C.

Se hace constar que .....  
es Profesor...Maestr... en este Establecimiento .....  
(táchese lo que no corresponda)

A pedido del interesado se extiende la presente constancia en .....  
....., a los ..... días del mes de .....  
..... del año 200....

para ser presentada ante las autoridades de .....

VERONESI • Santa Rosa 701 • Cordoba • Cód. 204 / 05/05

SELLO

Firma y aclaración de autoridad

**CONSTANCIA DE ALUMNO REGULAR**

ESCUELA SECUNDARIA N° 44 JORGE LUIS BORGES

Se hace constar que .....

DNIN° ..... alumno/a regular de ..... Año "....." de este establecimiento.

Se extiende la presente constancia en **Miraflores, Departamento Capayán, Catamarca** a los ..... Días  
mes de ..... Del año **20.....** para ser presentada ante las autoridades que lo requieran.

Sello del Establecimiento

Firma Autorizada



# Permiso de Exámen

Establecimiento \_\_\_\_\_

ALUMNO REGULAR

Permiso de Exámen N° \_\_\_\_\_

**CONDICION:** REGULAR - PREVIO - LIBRE - EQUIVALENCIA  
(Tache lo que no corresponda)

Conste por la presente que el alumno \_\_\_\_\_  
esta habilitado para rendir las asignaturas \_\_\_\_\_  
año de estudios, que se indican a continuación, lo que hizo en fechas señaladas.

N° de Orden	ASIGNATURAS	CURSO	FECHA	Calificación	Firma del Presidente de la Mesa Examinadora
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					

EMPORIO GRAFICO - Moja Ballejo 333 - nr 453401

\_\_\_\_\_ de \_\_\_\_\_ de \_\_\_\_\_  
lugar día mes año

(Firma manuscrita del Secretario)

SELLO

- NOTA:**
- 1) Para poder rendir exámen el alumno deberá presentar a la mesa examinadora este permiso y su documento de identidad.
  - 2) Los exámenes escritos deben ser hechos con tinta.



# ACTA VOLANTE DE EXAMENES

Form.884 EX.

ESTABLECIMIENTO \_\_\_\_\_

Exámenes de alumnos: \_\_\_\_\_ Espacio Curricular: \_\_\_\_\_  
Disciplina \_\_\_\_\_

Año	División	Turno

Libro de Actas N°: \_\_\_\_\_ Folio N°: \_\_\_\_\_

N° de Ciudadano	N° del Permiso	D.N.I. N°	APELLIDO Y NOMBRES	CALIFICACIONES			N° UNIDADES	
				Esc.	Oral	Definitiva	Esc.	Oral
1								
2								
3								
4								
5								
6								
7								
8								
9								
10								
11								
12								
13								
14								
15								
16								
17								
18								
19								
20								
21								
22								
23								
24								

VERONESI - Santa Rosa 701 - cód. 6 - 01/07

A continuación del último alumno deberá firmar el Secretario

Presidente \_\_\_\_\_  
Apellido y Nombres / Firma

Vocal \_\_\_\_\_  
Apellido y Nombres / Firma

Vocal \_\_\_\_\_  
Apellido y Nombres / Firma

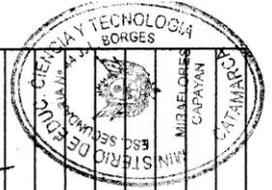
Total de alumnos	_____
Aprobados	_____
Aplazados	_____
Ausentes	_____

FECHA

D	M	A
I	E	N
A	S	O

PLANILLAS DE CALIFICACIONES CICLO SUPERIOR 2012-ESCUELA SECUNDARIA N°44 "JORGE LUIS BORGES"  
 ORIENTACIÓN: ECONOMÍA Y ADMINISTRACIÓN  
 ESPACIO CURRICULAR:  *Leng. And. his. Soc. (pa. de base)* DOCENTE:  *Rolando Leonor de* CURSO: 4to. DIV.: "A" TURNO MAÑANA

N°	APELLIDOS Y NOMBRES	1er. Trimestre			2do. Trimestre			3er. Trimestre			Calificación Anual		Examen Inst. Dic.	Examen Inst. Febrero	Calificación Definitiva	Observaciones
		1er. Trimestre	2do. Trimestre	3er. Trimestre	1er. Trimestre	2do. Trimestre	3er. Trimestre	Calificación Anual	Calificación Definitiva							
1	ABALOS, María del Pilar	6 (seis)	7 (siete)	7 (siete)	7 (siete)	7 (siete)			7 (siete)							
2	AGUIRRE, Ayelén Erika	7 (siete)	7 (siete)	8 (ocho)	8 (ocho)	8 (ocho)	8 (ocho)			8 (ocho)	A = 27.0% B = 6.0% C = -					
3	ARAGON, Isabel del C.	7 (siete)	8 (ocho)	9 (nueve)	8 (ocho)	8 (ocho)	8 (ocho)	8 (ocho)	8 (ocho)	8 (ocho)	8 (ocho)			8 (ocho)		
4	ARIZA, Samanta Nieves	6 (seis)	8 (ocho)	8 (ocho)	8 (ocho)	8 (ocho)			8 (ocho)							
5	ASTRADA, Micaela A.	6 (seis)	7 (siete)	7 (siete)	7 (siete)	7 (siete)			7 (siete)							
6	AVALOS, María Beatriz	9 (nueve)	9 (nueve)	9 (nueve)	9 (nueve)			9 (nueve)								
7	BAZAN, Santiago Noel	5 (cinco)	6 (seis)	7 (siete)	6 (seis)	6 (seis)	6 (seis)	6 (seis)	6 (seis)	6 (seis)	6 (seis)			6 (seis)		
8	BAZAN, Tania de los A.	6 (seis)	7 (siete)	7 (siete)	7 (siete)	7 (siete)			7 (siete)							
9	BERNEDE, Marcelo Dario	5 (cinco)	6 (seis)	7 (siete)	6 (seis)	6 (seis)	6 (seis)	6 (seis)	6 (seis)	6 (seis)	6 (seis)			6 (seis)		
10	CALCINA, Matias Exequiel	7 (siete)	7 (siete)	8 (ocho)	7 (siete)	7 (siete)	7 (siete)	7 (siete)	7 (siete)	7 (siete)	7 (siete)			7 (siete)		
11	CARRIZO, Alan Nahuel	8 (ocho)	9 (nueve)	9 (nueve)	9 (nueve)	9 (nueve)			9 (nueve)							
12	CORREA, Eugenia Janet	9 (nueve)	10 (diez)	10 (diez)	10 (diez)	10 (diez)			10 (diez)	A = 100% B = - C = -						
13	DIAZ, Emiliano	7 (siete)	7 (siete)	7 (siete)	7 (siete)			7 (siete)								
14	FERNANDEZ, Bruno E.	6 (seis)	8 (ocho)	9 (nueve)	8 (ocho)	8 (ocho)	8 (ocho)	8 (ocho)	8 (ocho)	8 (ocho)	8 (ocho)			8 (ocho)		
15	FIGUEROA, Mariana L.	6 (seis)	7 (siete)	7 (siete)	7 (siete)	7 (siete)			7 (siete)							
16	GOMEZ, Axel Agustin	5 (cinco)	7 (siete)	7 (siete)	7 (siete)	7 (siete)			7 (siete)							
17	GRANILLO, Dayana M.	7 (siete)	7 (siete)	7 (siete)	7 (siete)			7 (siete)								
18	HERRERA, Mercedes F.	6 (seis)	7 (siete)	8 (ocho)	7 (siete)	7 (siete)	7 (siete)	7 (siete)	7 (siete)	7 (siete)	7 (siete)			7 (siete)		
19	KURPIEL A., Juan Sebastian	7 (siete)	8 (ocho)	8 (ocho)	8 (ocho)	8 (ocho)			8 (ocho)	2° Trimestre						
20	LESCANO, Fiamma G.	9 (nueve)	9 (nueve)	9 (nueve)	9 (nueve)			9 (nueve)	A = 100% B = - C = -							
21	MALDONADO, Daniel Omar	7 (siete)	7 (siete)	7 (siete)	7 (siete)			7 (siete)								
22	MARTINEZ, Priscila Vanesa	6 (seis)	7 (siete)	7 (siete)	7 (siete)	7 (siete)			7 (siete)							
23	MOYA, Gisela Guadalupe	8 (ocho)	8 (ocho)	10 (diez)	8 (ocho)	8 (ocho)	8 (ocho)	8 (ocho)	8 (ocho)	8 (ocho)	8 (ocho)			8 (ocho)		
24	NARVAEZ, Carla Gisel	7 (siete)	8 (ocho)	8 (ocho)	8 (ocho)	8 (ocho)			8 (ocho)							
25	PACHECO, Gabriela Eugenia	6 (seis)	7 (siete)	8 (ocho)	6 (seis)	6 (seis)	6 (seis)	6 (seis)	6 (seis)	6 (seis)	6 (seis)			6 (seis)		
26	PALACIOS, Debora Aldana	6 (seis)	7 (siete)	7 (siete)	7 (siete)	7 (siete)			7 (siete)							
27	ROBLEDO, Paula Estefanía	6 (seis)	7 (siete)	7 (siete)	7 (siete)	7 (siete)			7 (siete)							
28	SAAVEDRA, Mariano A.	8 (ocho)	8 (ocho)	9 (nueve)	8 (ocho)	8 (ocho)	8 (ocho)	8 (ocho)	8 (ocho)	8 (ocho)	8 (ocho)			8 (ocho)		
29	SANDI, Carlos Alfredo	5 (cinco)	7 (siete)	7 (siete)	5 (cinco)	5 (cinco)	5 (cinco)	5 (cinco)	5 (cinco)	5 (cinco)	5 (cinco)			5 (cinco)		
30	SILVA, Noelia Soledad	7 (siete)	8 (ocho)	8 (ocho)	7 (siete)	7 (siete)	7 (siete)	7 (siete)	7 (siete)	7 (siete)	7 (siete)			7 (siete)		
31	SORIA, Cristian Adrian	5 (cinco)	6 (seis)	7 (siete)	5 (cinco)	5 (cinco)	5 (cinco)	5 (cinco)	5 (cinco)	5 (cinco)	5 (cinco)			5 (cinco)		
32	TOLEDO, Yenefir G.	6 (seis)	7 (siete)	7 (siete)	6 (seis)	6 (seis)	6 (seis)	6 (seis)	6 (seis)	6 (seis)	6 (seis)			6 (seis)		
33	RODRIGUEZ, José Luis	7 (siete)	8 (ocho)	9 (nueve)	7 (siete)	7 (siete)	7 (siete)	7 (siete)	7 (siete)	7 (siete)	7 (siete)			7 (siete)		
34																





CURSO	DIVIS.	TURNO	LIBRETA DE CALIFICACIONES DE: Repetto, Vahine										D.N.I. N°: 41.957.331			AÑO 2012			
N°	"A"	M.	Lengua y Lit.	Matemática	Educ. Física	Educ. Art. Húmana	Historia	Geografía	Biología	Física-Química	Tecnología	Ética y Ciudad.	Lengua ext.	JUSTIFICADAS	INJUSTIFICADAS	AMONESTACIONES	CONDUCTA	FIRMA AUTORIDAD COMPETENTE	FIRMA DEL PADRE, MADRE, TUTOR O ENCARGADO
1er. Trimestre			8	9	7	6	6	6	6	6	10	8	6	5	1	-	B	Dr. AMAR OMAR SANTIUCHI DI. RECTOR ESC. SECUNDARIA N° 44 "JORGE-LUIS BORGES"	
2do. Trimestre																			
3er. Trimestre																			
Inst. de Acompañamiento Compensación Diciembre																			
Inst. de Acompañamiento Compensación Febrero																			
Examen																			
Calificación Definitiva																			
PROMEDIO FINAL																			



Grado o Curso: 3º División: B Turno: /- Año: 2017

Nº de Orden	APELLIDO Y NOMBRE	MARZO		ABRIL		MAYO		JUNIO	
		Rec. Nº	\$						
01	AGUIRRE, Hemesio P.		30		31	18/04	32		
02	BRANDA, Brian J.Q.	252	30						
03	ARREGUER, Eber Alexis	236	30						
04	ARREGUER, Maria F.								
05	MUSTOS, Mauricio H.								
06	BRAND, Luisa Anahi	253	30	29/04	31		32		32
07	CORDOBA, Rolando G.								
08	PAVIA, Ivair Viviana	233	30	20/04	31		32		32
09	TESCANO, Federico Yamil								
10	PROTECO, Cristian H.		30		31		32		32
11	PERAZOZA, Melba E.								
12	PEREZ, Maria Belen	246	30		32		32		32
13	SANTILLAN, BRAND, Fernando	23/02	25		31		32		32
14	SIGAMPA, Jose Ignacio	292	30						
15	TULA Franco Emiliano	232	30	27/04	31		32		32
16	YAPURA, Dalila P.								
17	PROTECO, Hernan Federico.								





ACTA DE APERTURA DEL LIBRO

AÑO: 2012 CURSO: 4to GRADO: DIVISION: A TURNO: Mañana DATOS

N° de Orden	Apellido y Nombre	Edad	Fecha de Nacimiento	Nacionalidad	N° de Matrícula	N° de Cédula y Policia
1	ABALOS Maria del Pilar		17/08/96	Argentina	39.541.469	
2	AGUIRRE Mylen Cecilia		01/06/97	Argentina	40.433.712	
3	ARAGON Isabel del C.		30/03/96	Argentina	39.541.454	
4	ARIZA					FALTA DATOS
5	ASTRADA Nicoola Antonela		27/07/96	Argentina	39.541.456	
6	AVALOS Maria Beatriz		16/12/95	Argentina	39.015.293	
7	BARAN Santiago Noel		02/09/94	Argentina	39.998.614	
8	BARAN Tania de los P.		01/11/96	Argentina	40.598.918	
9	BERNARDI Marcelo Darío		08/11/96	Argentina	39.541.488	
10	CAICINA Matias Ezequiel		05/05/96	Argentina	39.540.018	
11	CARRIZO Dian Nahuel		05/05/97	Argentina	40.380.635	
12	CORREA Eugenia Janet		03/10/96	Argentina	39.016.679	
13	DINE Jose Emiliano		08/03/95	Argentina	38.175.183	
14	FERNANDEZ BRUNO E.		17/05/96	Argentina	38.753.736	
15	FIGUEROA Mariana Leticia		15/02/96	Argentina	38.753.722	
16	GOMEZ					FALTA DATOS
17	GRANILLO Dariana Mariel		16/03/96	Argentina	38.224.789	
18	HERRERA Mercedes F.		24/09/96	Argentina	39.016.871	
19	KURPIEL P. Juan Sebastian		14/03/97	Argentina	40.241.627	
20	LEZANO Fiamma Guadalupe		30/06/97	Argentina	40.433.719	
21	MAIDORADO Daniel Omar		25/04/91	Argentina	35.501.006	
22	MARTINEZ Priscila Vanesa		01/04/97	Argentina	39.998.617	
23	MOYA Gisela Guadalupe		06/02/97	Argentina	39.998.603	
24	NAPUAE Carla Gisel		27/07/96	Argentina	39.541.460	
25	PACHECO Gabriela Eugenia		14/06/97	Argentina	39.805.895	
26	PALACIOS Debora Aldora		20/06/97	Argentina	40.433.709	
27	RODRIGUEZ Paula Estefana		29/06/96	Argentina	38.753.750	
28	RODRIGUEZ Eduardo E.		27/01/95	Argentina	40.521.257	
29	SABUERA Mariana P.		31/08/96	Argentina	39.016.356	
30	SANDI Carlos Alfredo		05/03/97	Argentina	39.997.257	
31	SILVA Noelia Soledad		06/06/97	Argentina	40.433.705	
32	SORIA Cristian Adrian		07/04/95	Argentina	37.642.547	
33	TOLEDO Yennifer Guadalupe		13/11/96	Argentina	39.541.489	
34						
35						
36						
37						
38						
39						
40						
41						
42						
43						
44						
45						





D A T O S D E L

N° de Orden	NUMERO DE CERTIFICADOS				NOMBRE DEL PADRE O TUTOR
	Antivar.	Triple	Poliom	B. C. G.	
1					
2					
3					Francisca J. Piagon
4					HERREERA Graciela
5					Monica P. Ferrandez
6					LORA Lorena Beatriz
7					PREQUEZ, Verónica
8					
9					
10					
11					Marcelo Miranda
12					Rosa Correa
13					Felipe Diaz
14					Rita B. Polomague
15					Ramon J. Figueroa
16					Lorena Prequeza
17					
18					Karel Herrera
19					Sebastian E. KURPIEL
20					MELLSWEG, Liliana.
21					
22					Norma Gladis Barzan
23					
24					Cynthia Homi Parvaz
25					
26					Maria Rosa Acosta
27					Nancy Paola Robledo
28					
29					Blanca Liliana Saavedra
30					Maria Sara Armas
31					
32					COMPERO, Mista Graciela
33					Marta Edith Ojeda
34					Luisa Carole Sanjilten
35					
36					
37					
38					
39					
40					
41					
42					
43					
44					
45					



PADRE O TUTOR

Nacionalidad	Profesión	Firma del Padre o Tutor	Anotaciones Particulares	Ingresó	Egresó
		x <del>[Firma]</del>			
	Amo de Casa	x <del>[Firma]</del>			
	Amo de Casa	x <del>[Firma]</del>	* Sonomij Moira (Corresponde ASTONDA)		
Argentina	Amo de Casa	x <del>[Firma]</del>			
Argentina	Empleada	x <del>[Firma]</del>			
	Empleada	x <del>[Firma]</del>			
Argentina	Amo de Casa	x <del>[Firma]</del>			
Argentina	Empleada	x <del>[Firma]</del>			
	Amo de Casa	x <del>[Firma]</del>			
	Empleado	x <del>[Firma]</del>	154792784.		
Argentina	Amo de Casa	x <del>[Firma]</del>			
Argentina	Empleado	x <del>[Firma]</del>		4484714	
	Amo de Casa	x <del>[Firma]</del>			
	Amo de Casa	x <del>[Firma]</del>			
Argentina	Amo de Casa	x <del>[Firma]</del>			
Argentina	Amo de Casa	x <del>[Firma]</del>			
Argentina	Amo de Casa	x <del>[Firma]</del>			
Argentina	Amo de Casa	x <del>[Firma]</del>	27-21984245-7		
	Amo de Casa	x <del>[Firma]</del>			
Argentina	Amo de Casa	x <del>[Firma]</del>			
Argentina	Amo de Casa	x <del>[Firma]</del>			
Argentina	Amo de Casa	x <del>[Firma]</del>			



Folio N° 62

Esc. Sec. N° 44 Jorge Luis Borges  
ESTABLECIMIENTO:

**ACTA DE EXAMENES DE ALUMNOS**

de Lengua Extranjera Inglés I  
correspondiente al 4to año de estudios.  
A los 31 días del mes de Julio de 2012

reunida la comisión Examinadora de la asignatura mencionada, con asistencia de sus miembros.  
Señores: Prof. Carina Ortiz, Prof. Virginia Sala, Prof. Pablo Padovani  
procedió a cumplir con el resultado que se consigna a continuación:

NOTA: Sera considerada como fraudulenta toda enmienda o raspadura que no haya sido salvada expresamente con la firma de todos los miembros de la mesa.  
El número de alumnos examinados, aprobados y desaprobados, se anotará con letras y números.

N° de Orden	Permiso de Examen N°	Tipo y N° de Documento	APELLIDO Y NOMBRES	Exámen Escrito	Exámen Oral	Calificación Definitiva
1		36051317	Santellan, Juan Alexis	4,50 (cuatro y medio)	- - -	4,50 (cuatro y medio)
2		38153737	Yapura, Erika Marina	6 (seis)	- - -	6 (seis)
3		37642590	Pacheco, Emilce Braceli	6 (seis)	- - -	6 (seis)
4		38016732	Nieva, Celeste Beatriz	4 (cuatro)	- - -	4 (cuatro)
5	- -	- -	Moreno, Carlos Exequiel	- FUSEANTE -	- - -	- - -
6		38753725	Arreguez, Elias Joel	7 (siete)	- - -	7 (siete)
7		30224774	Medina, Lucas Gabriel	3 (tres)	- - -	3 (tres)
8						
9						
10						
11						
12						
13						
14						
15						
16						
17						
18						
19						
20						
21						
22						
23						
24						
25						
26						

Se hace constar que sobre un total de siete alumnos, resultaron: tres aprobados, Tres desaprobados y un ausentes