ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

SISTEMA ACADÉMICO “AGUENA”

EXAMEN DE GRADO (COMPLEXIVO)

Previo a la obtención del Título de:

LICENCIADO EN SISTEMAS DE INFORMACIÓN

LUIS MARLON ORTIZ BARCIA

GUAYAQUIL – ECUADOR

AÑO: 2016

AGRADECIMIENTOS

Primeramente, a Dios que me ha dado la vida, fuerzas y recursos necesarios para llegar hasta donde he llegado, a mis padres que han sido pilar fundamental en mi vida y que, con mucho esfuerzo, dedicación y fe, invirtieron en mí y me han acompañado incondicionalmente, y a cada una de las personas que han colaborado con una palabra, incentivo o grano de arena para que yo me haya formado y haya logrado lo que he logrado. A todos ellos y a los que vendrán y estarán conmigo.

LUIS MARLON ORTIZ BARCIA

DEDICATORIA

El presente proyecto lo dedico a Dios sobre todas las cosas, a mis padres, la Lcda. Zully Barcia Piguave y al Sr. Luis Ortiz Holguín por ser mi mayor apoyo en esta y todas las etapas de mi vida. Dedico este proyecto también a todos y cada uno de mis amigos, compañeros y familiares que con una palabra de aliento y su grano de arena han sido de apoyo y empuje para continuar y avanzar a pesar de las adversidades, y finalmente a cada una de las autoridades y docentes que han sabido instruirme de muchas formas para luchar por mis sueños y alcanzar mis metas.

LUIS MARLON ORTIZ BARCIA

TRIBUNAL DE EVALUACIÓN

MSIG. Marjorie Chalén

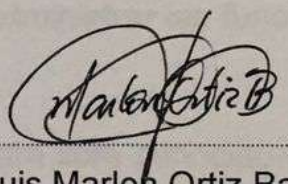
EVALUADOR 1

MSIG. Ronald Barriga

EVALUADOR 2

DECLARACIÓN EXPRESA

"La responsabilidad y la autoría del contenido de este Trabajo de Titulación, me corresponde exclusivamente; y doy mi consentimiento para que la ESPOL realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual"



Luis Marlon Ortiz Barcia

RESUMEN

La Academia de Guerra Naval requiere implementar una herramienta que les permita automatizar y mantener un registro histórico de todos los procesos y actividades estudiantiles de la academia. Así como también mejorar el portal que ellos utilizan en todas sus áreas que les permita no solo estar disponible para los estudiantes en un ambiente web, sino también así poder integrar todas las áreas con las tareas que se realizan a diario.

Dentro del proceso de levantamiento de información surgieron muchos problemas que tuvieron que ver con nóminas de estudiantes, llevar un control de todo el programa académico que se lleva en cada parcial, así como también administrar las funciones de todas las personas que laboran en la institución.

Se ha podido implementar una solución basada en un software libre conocido como Odoo. Esta solución fue la más apropiada ya que cuenta con una extensa gama de módulos y aplicativos que les permitan integrar cada una de sus funcionalidades e incluso mejorar los procesos que ellos realizaban manualmente.

ÍNDICE GENERAL

CAPÍTULO # 1

1. PLANTEAMIENTO DEL PROBLEMA.

La Academia de Guerra Naval ha venido realizando todos sus procesos de manera manual y a través de un sistema local base desarrollado previamente ya por un largo tiempo. Como institución siempre están buscando constantes mejoras para el manejo de todos sus procedimientos y actividades diarias. El entorno en el que ellos trabaja involucra no solo a estudiantes sino también a todo el personal administrativo, financiero, hasta gerencial, sin excluir obviamente a los docentes.

El sistema que ellos utilizan está basado en el lenguaje de programación Visual.net, el mismo que llevan utilizando ya por 5 años. Previo a este tenían un sistema de manejo de datos utilizando C++ que les permitía al menos tener un control de sus calificaciones y reportes necesarios para funciones de la institución.

Con el pasar el tiempo, se encontraban la barrera de la tecnología que día a día se va haciendo más grande si se trata de alcanzar o estar a la par en cuanto a las novedades y progresos de la misma. Estaban al tanto y pendientes de la búsqueda de una herramienta o solución que les permita migrar, mudar, integrar y administrar todas estas funcionalidades y requerimiento desde un solo ambiente.

Una página web, era necesaria para poder administrar todas las áreas de la institución ya que ofrece un ambiente fácil y accesible para trabajar no solo desde la institución sino desde cualquier punto de conexión a internet.

Un control de tareas y planificación como un Syllabus fue uno de los temas principales que surgieron ante la problemática, de que es un proceso engorroso llevar registros únicamente en papeles y al momento de buscarlo les tomaba mucho tiempo. Ante esta necesidad ellos también contaron con un sistema de registro de estudiantes y actividades al cual le llamaron “*LibAcademia*” y que les permitía tener almacenado en una base de datos el historial académico de los estudiantes.

Otros de los problemas que ellos consideraban era la integración de todas y cada una de las áreas que no se alcanzaban a controlar dentro de los sistemas con los que ellos trabajaban.

1.1 Objetivos

1.1.1 Objetivo general del proyecto

Desarrollar un sistema académico mediante herramientas de software libre como solución de manejo automático de datos administrativos, estudiantiles, planificación académica y reportes para la Academia de Guerra Naval de Guayaquil, para así disponer de información oportuna y precisa que les permita una toma de decisiones acertada para un buen desenvolvimiento institucional.

1.1.2 Objetivos específicos

Levantamiento de información y requerimientos para el nuevo sistema.

Concentrar y unificar los requerimientos de datos administrativos y académicos.

Desarrollar un sistema orientado a la web con los módulos requeridos por la institución.

Implementar la gestión multiusuario para que personal administrativo, docente y estudiantil pueda hacer uso de la página web.

Otorgar accesos a cada una de las personas que laboran en la institución.

Realizar el ingreso de los registros académicos que se usaban en el antiguo sistema. Entre estos se encuentran cada uno de los historiales de cursos, actividades, calificaciones, planificación académica, etc.

Proporcionar al personal administrativo y docente la capacitación necesaria en el uso del nuevo ambiente web.

Integrar los servicios del antiguo sistema con la nueva página web SIEAGUENA.

Establecer políticas y normativas generales acerca de los privilegios, accesos y diferentes tareas que involucren al sistema.

1.2 Alcance

El sistema será desarrollado en página web, utilizando herramientas de software libre y cubrirá los siguientes procesos:

- Matriculación de estudiantes.
- Registrar y controlar calificaciones, aprovechamiento y asistencias de los estudiantes.
- Generar reportes. Entre estos tenemos reportes de calificaciones, reportes de asistencias, actividades, administrativos, estadísticos, de productividad y certificados en general.
- Planificación académica y syllabus
- Calendario de actividades y eventos de la institución.
- Administración accesos y credenciales para cada uno de los usuarios finales de la institución

CAPÍTULO # 2

2. MARCO TEÓRICO.

2.1 Página web

Una página de Internet o página Web es un documento electrónico adaptado particularmente para el Web, que contiene información específica de un tema en particular y que es almacenado en algún sistema de cómputo que se encuentre conectado a la red mundial de información denominada Internet, de tal forma que este documento pueda ser consultado por cualquier persona que se conecte a esta red mundial de comunicaciones y que cuente con los permisos apropiados para hacerlo.

Una página Web tiene la característica peculiar de que el texto se combina con imágenes para hacer que el documento sea dinámico y permita que se puedan ejecutar diferentes acciones, una tras otra, a través de la selección de texto remarcado o de las imágenes, acción que nos puede conducir a otra sección dentro del documento, abrir otra página Web, iniciar un mensaje de correo electrónico o transportarnos a otro Sitio Web totalmente distinto a través de sus hipervínculos. Los documentos pueden ser elaborados por instituciones públicas o privadas, asociaciones, y por las propias personas en lo individual.

Todo, absolutamente todo lo que encuentras en Internet es una página web, desde el buscador Google, hasta el Facebook y el mismo Hotmail. Por su puesto hay otras cosas que no son páginas web por ejemplo el “MSN” o chats, programas de voz en línea, programas de telefonía IP o Skype.

En el siguiente proyecto se utilizará las tecnologías: HTML, CSS.

HTML. - es la "lengua materna" del navegador, con este lenguaje se definen las páginas web. Básicamente se trata de un conjunto de etiquetas que sirven para definir el texto y otros elementos que compondrán una página web.

El HTML es un lenguaje de marcación de elementos para la creación de documentos hipertexto, muy fácil de aprender, lo que permite que cualquier persona, aunque no haya programado en la vida, pueda enfrentarse a la tarea de crear una web. HTML

es fácil y podremos dominar el lenguaje. Así como la incorporación de otros lenguajes para definir el formato con el que se tienen que presentar las webs, como CSS.

Este lenguaje se escribe en un documento de texto, por eso necesitamos un editor de textos para escribir una página web. Así pues, el archivo donde está contenido el código HTML es un archivo de texto, con una peculiaridad, que tiene extensión .html o .htm (es indiferente cuál utilizar). De modo que cuando programemos en HTML lo haremos con un editor de textos, lo más sencillo posible y guardaremos nuestros trabajos con extensión .html, por ejemplo mipagina.html.

Por adelantar un poco cómo se utiliza el HTML diremos que el lenguaje consta de etiquetas que tienen esta forma <B> o <P>. Cada etiqueta significa una cosa, por ejemplo <B> significa que se escriba en negrita (bold) o <P> significa un párrafo, <A> es un enlace, etc. Casi todas las etiquetas tienen su correspondiente etiqueta de cierre, que indica que a partir de ese punto no debe de afectar la etiqueta. Por ejemplo </B> se utiliza para indicar que se deje de escribir en negrita.

CSS. - Hojas de estilo en cascada viene del inglés Cascading Style Sheets. CSS es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML). La idea del desarrollo de CSS es separar la estructura de un documento de su presentación.

Cuando se utiliza CSS, la etiqueta <H1> no debería proporcionar información sobre cómo será visualizado, solamente marca la estructura del documento. La información de estilo, separada en una hoja de estilo, específica cómo se ha de mostrar <H1>: color, fuente, alineación del texto, tamaño y otras características no visuales, como definir el volumen de un sintetizador de voz.

La información de estilo puede ser adjuntada como un documento separado o en el mismo documento HTML. En este último caso podrían definirse estilos generales en la cabecera del documento o en cada etiqueta particular mediante el atributo "style".

CSS proporciona tres caminos diferentes para aplicar las reglas de estilo a una página Web: Una hoja de estilo externa, es una hoja de estilo que está almacenada en un archivo diferente al archivo donde se almacena el código HTML de la página Web.

Esta es la manera de programar más potente, porque separa completamente las reglas de formateo para la página HTML de la estructura básica de la página.

Una hoja de estilo interna, es una hoja de estilo que está incrustada dentro de un documento HTML. De esta manera se separa la información del estilo del código HTML propiamente dicho. La única vez que se usa una hoja de estilo interna, es cuando se quiere proporcionar alguna característica a una página Web en un simple fichero.

Un estilo en línea es un método para insertar el lenguaje de estilo de página directamente dentro de una etiqueta HTML. Esta manera de proceder no es totalmente adecuada. El incrustar la descripción del formateo dentro del documento de la página Web, a nivel de código, se convierte en una manera larga, tediosa y poco elegante de resolver el problema de la programación de la página.

2.2 Sitio Web

Es un conjunto de archivos electrónicos y páginas Web referentes a un tema en particular, que incluye una página inicial de bienvenida, generalmente denominada home page, con un nombre de dominio y dirección específico en Internet; empleados por las instituciones públicas y privadas, organizaciones e individuos para comunicarse con el mundo entero.

Su Sitio Web no necesariamente debe localizarse en el sistema de cómputo de su propiedad. Los documentos que integran el Sitio Web pueden ubicarse en un equipo en otra localidad, inclusive en otro país. El único requisito es que el equipo en el que residan los documentos esté conectado a la red mundial de Internet.

Los Sitios Web pueden ser de diversos géneros, los sitios de negocios, servicio, comercio electrónico en línea, imagen corporativa, entretenimiento y sitios informativos.

Definición sitio web dinámico. - Un sitio Web dinámico es aquel que muestra su contenido obteniéndolo, normalmente, de una base de datos, empleando para ello lenguajes para la web como JSP, PHP o ASP. Mediante estos lenguajes el desarrollador web crea aplicaciones que acceden a la base de datos y muestran al usuario final la web, según el contenido de las tablas de la base de datos.

Por supuesto, dependiendo del tipo de web, este método de dinamismo se puede aprovechar para cualquier uso: foros, comunidad de usuarios, descargas, perfiles, comentarios, votos, libros de visita, compra de productos, catálogo de productos, etc.

Webs dinámicas vs estáticas. - La ventaja principal de las webs dinámicas frente a las estáticas es que, con las dinámicas, las secciones y posibilidades son casi infinitas. En una web dinámica podremos tener foros, encuestas, comunidad de usuarios, etc. algo imposible en una web estática. Además, las webs dinámicas requieren de muy pocos conocimientos por parte del usuario para gestionar su contenido. Otra de las grandes ventajas es que existen ya desarrollados y gratuitos numerosos CMS como Joomla que son sistemas de gestión de contenidos profesionales. Éstos permiten crear un sitio web completamente dinámico casi sin conocimientos de HTML ni de programación web. Permiten añadirles módulos de foros, galerías de imágenes y demás. El inconveniente principal de los sitios web dinámicos frente a los estáticos es que el desarrollo de este tipo de Web dinámicas es muchísimo más complicado qué el de una web estática. Para desarrollar una web dinámica se requieren conocimientos de programación (en PHP u otro lenguaje elegido como ASP, JSP, etc.), conocimientos de bases de datos (MySQL, PostgreSQL, etc.) y conocimientos de HTML. Otra desventaja es que las webs dinámicas suelen ser más lentas en su carga que las webs estáticas, pues requieren de acceso a base de datos y de compilación de los archivos PHP, ASP, JSP, etc. en el servidor para ser devueltos al usuario en HTML (entendible por el navegador).

2.3 Open Source (Código Abierto)

El código abierto es el software distribuido y desarrollado libremente. Se focaliza más en los beneficios prácticos (acceso al código fuente) que en cuestiones éticas o de libertad que tanto se destacan en el software libre. Para muchos el término «libre» hace referencia al hecho de adquirir un software de manera gratuita, pero más que eso, la libertad se refiere al poder modificar la fuente del programa sin restricciones de licencia, ya que muchas empresas de software encierran su código, ocultándolo y restringiéndose los derechos a sí misma.

La idea del código abierto se centra en la premisa de que, al compartir el código, el programa resultante tiende a ser de calidad superior al software propietario, es una

visión técnica. Por otro lado, el software libre tiene tendencias filosóficas e incluso morales: el software propietario, al no poder compartirse, es «antiético» dado que prohibir compartir entre seres humanos va en contra del sentido común. Ninguna adaptación ni cambios que no haya realizado previamente la empresa fabricante.

El código abierto ofrece:

- Acceso al código fuente: Para modificarlo, corregirlo o añadir más prestaciones.
- Gratuidad: El software puede obtenerse libremente.
- La posibilidad de evitar monopolios de software propietario: Para no depender de un único fabricante de software.
- Un modelo de avance: Por lo cual la información no se oculta.
- Al igual que el software libre, el código abierto tiene una serie de requisitos necesarios para que un programa pueda considerarse dentro de este movimiento, estos son:
- Libre redistribución: el software debe poder ser regalado o vendido libremente.
- Código fuente: el código fuente debe estar incluido u obtenerse libremente.
- Trabajos derivados: la redistribución de modificaciones debe estar permitida.
- Integridad del código fuente del autor: las licencias pueden requerir que las modificaciones sean redistribuidas solo como parches.
- La licencia no debe discriminar a ninguna persona o grupo: nadie puede dejarse fuera.
- Sin discriminación de áreas de iniciativa: los usuarios comerciales no pueden ser excluidos.
- Distribución de la licencia: deben aplicarse los mismos derechos a todo el que reciba el programa
- La licencia no debe ser específica de un producto: el programa no puede licenciarse solo como parte de una distribución mayor.
- La licencia no debe restringir otro software: la licencia no puede obligar a que algún otro software que sea distribuido con el software abierto deba también ser de código abierto.

- La licencia debe ser tecnológicamente neutral: no debe requerirse la aceptación de la licencia por medio de un acceso por clic de ratón o de otra forma específica del medio de soporte del software.

2.4 Programas en código abierto

- Sistemas operativos: los más conocidos, Ubuntu y Debian, basados en Linux. Android, de Google, para teléfonos inteligentes y tabletas.
- Programas: las suites ofimáticas Open Office y Libre Office, el navegador Firefox, el cliente de correo electrónico Thunderbird, el reproductor multimediaVLC o el editor de imágenes GIMP
- Los programas de Apache Software Foundation y Github.
- El lenguaje de programación de Apple y las plataformas de enseñanza como Moodle (un ejemplo es sloodle).

En este proyecto como parte de la solución estamos utilizando uno de los softwares de código abierto que está predominando en la actualidad, conocido como ODOO.

ODOO[1] es uno de los softwares de gestión empresarial más populares del mundo, con más de 2.000.000 de usuarios, y con presencia en más de 120 países. Actualmente, más de 1500 desarrolladores trabajan diariamente para mejorar y optimizar todos sus procesos, adaptando y ajustando el software a los cambios del entorno.

A día de hoy es el ERP tecnológicamente más avanzado del mercado, habiendo desarrollado e integrado en su software funcionalidades de CRM y E-Commerce, que permiten aprovechar al máximo todas las oportunidades de negocio Online.

Entre otras funcionalidades, el software de Odoo ofrece:

- Gestión Financiera
- Compras
- Ventas
- Almacén
- Proyectos
- Recursos Humanos
- Inteligencia de Negocio

- Portal Web
- E-Commerce
- CRM

2.5 Base de datos

Se define una base de datos como una serie de datos organizados y relacionados entre sí, los cuales son recolectados y explotados por los sistemas de información.

Entre las principales características de los sistemas de base de datos están:

- Independencia lógica y física de los datos.
- Redundancia mínima.
- Acceso concurrente por parte de múltiples usuarios.
- Integridad de los datos.
- Consultas complejas optimizadas.
- Seguridad de acceso y auditoría.
- Respaldo y recuperación.
- Acceso a través de lenguajes de programación estándar.

En el presente proyecto se utilizará el manejador de base de datos POSTGRESQL.

POSTGRESQL[3] es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más potente del mercado y en sus últimas versiones no tienen nada que envidiarles a otras bases de datos comerciales.

PostgreSQL utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando.

La última serie de producción es la 9.3. Sus características técnicas la hacen una de las bases de datos más potentes y robustas del mercado. Su desarrollo comenzó hace más de 16 años, y durante este tiempo, estabilidad, potencia, robustez, facilidad de administración e implementación de estándares han sido las características que más se han tenido en cuenta durante su desarrollo. PostgreSQL funciona muy bien

con grandes cantidades de datos y una alta concurrencia de usuarios accediendo a la vez a el sistema.

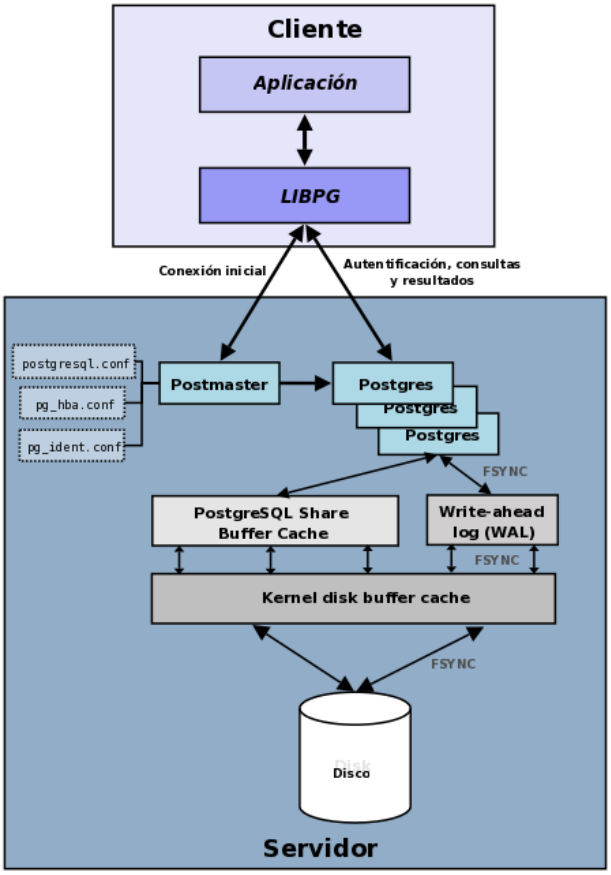


Figura 2.1 Componentes más importantes de PostgreSql

2.6 Lenguajes de programación

Los lenguajes de programación son herramientas que nos permiten crear programas y software. Entre ellos tenemos Delphi, Visual Basic, Pascal, Java, etc.

Los lenguajes de programación representan en forma simbólica y en manera de un texto los códigos que podrán ser leídos por una persona. Un lenguaje de programación es un lenguaje que puede ser utilizado para controlar el comportamiento de una máquina, particularmente una computadora. Consiste en un conjunto de reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos, respectivamente. Aunque muchas veces se usa lenguaje de

programación y lenguaje informático como si fuesen sinónimos, no tiene por qué ser así, ya que los lenguajes informáticos engloban a los lenguajes de programación y a otros más, como, por ejemplo, el HTML.

En el presente proyecto se implementará la solución utilizando uno de los lenguajes de programación web más modernos en la actualidad, PYTHON.

PYTHON es un lenguaje de programación poderoso y fácil de aprender. Cuenta con estructuras de datos eficientes y de alto nivel y un enfoque simple pero efectivo a la programación orientada a objetos. La elegante sintaxis de Python y su dinamismo, junto con su naturaleza interpretada, hacen de éste un lenguaje ideal para scripting y desarrollo rápido de aplicaciones en diversas áreas y sobre la mayoría de las plataformas. El intérprete de Python y la extensa biblioteca estándar están a libre disposición en forma binaria y de código fuente para las principales plataformas desde el sitio web de Python, https://www.python.org/, y puede distribuirse libremente. El mismo sitio contiene también distribuciones y enlaces de muchos módulos libres de Python de terceros, programas y herramientas, y documentación adicional. El intérprete de Python puede extenderse fácilmente con nuevas funcionalidades y tipos de datos implementados en C o C++ (u otros lenguajes accesibles desde C). Python también puede usarse como un lenguaje de extensiones para aplicaciones personalizables.

Python permite escribir programas compactos y legibles.

- Los programas en Python son típicamente más cortos que sus programas equivalentes en C, C++ o Java por varios motivos:
- Los tipos de datos de alto nivel permiten expresar operaciones complejas en una sola instrucción.
- La agrupación de instrucciones se hace por sangría en vez de llaves de apertura y cierre
- No es necesario declarar variables ni argumentos.

2.7 Metodologías de desarrollo de un sistema

Es un conjunto de actividades llevadas a cabo para desarrollar y poner en marcha un Sistema de Información. Son métodos que indican cómo hacer más eficiente el desarrollo de sistemas de información. Se suelen estructurar en fases de vida a los

sistemas para facilitar su planificación, desarrollo y mantenimiento. Las metodologías de desarrollo de sistemas deben definir: objetivos, fases, tareas, productos y responsables, necesarios para la correcta realización del proceso y su seguimiento.

Los principales objetivos de una metodología de desarrollo son:

- Definir actividades en el desarrollo del Proyecto de Sistema de Información.
- Unificar criterios en la organización para el desarrollo de Sistema de
- Información.
- Proporcionar puntos de control y revisión.
- Asegurar la uniformidad y calidad tanto del desarrollo como del sistema.
- Satisfacer las necesidades de los usuarios del sistema.
- Conseguir un mayor nivel de rendimiento y eficiencia del personal asignado
- al desarrollo.
- Ajustarse a los plazos y costes previstos en la planificación.
- Generar de forma adecuada la documentación asociada a los sistemas.
- Facilitar el mantenimiento posterior de los sistemas.

En el presente proyecto se está implementado la metodología XP.

¿Qué es XP (Programación Extrema)?

Nueva disciplina de desarrollo de software basada en la simplicidad, la comunicación, la retroalimentación y la refactorización de código.

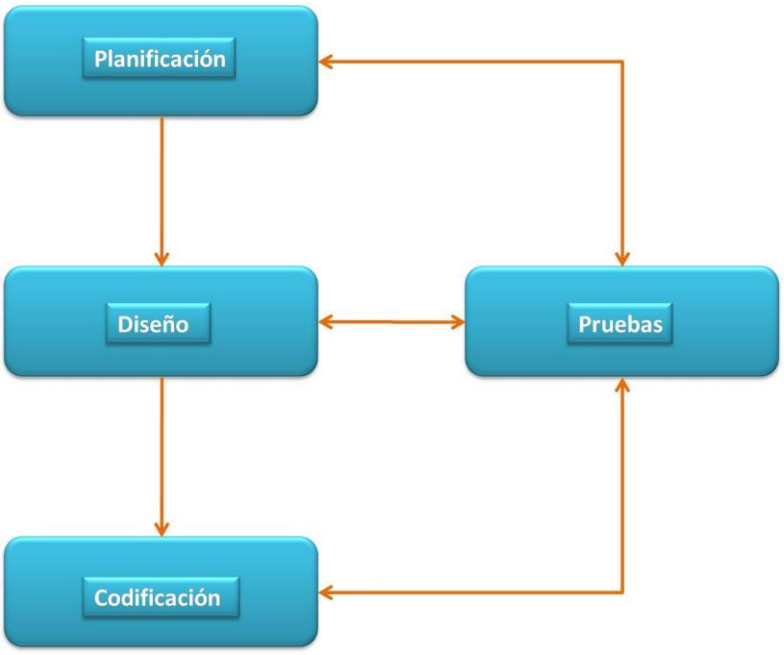


Figura 2.2 Fases de la Programación Extrema

CAPÍTULO # 3

3. IMPLEMENTACIÓN.

3.1 Propósito

El presente sistema a desarrollar tiene como propósito la integración dinámica de un ambiente web junto con el sistema académico que actualmente esta usando el personal de la institución. Se espera la integración de sistemas como gestión de las notas del aprovechamiento de los estudiantes, gestión de la asistencia y calificaciones de los estudiantes, gestión de las notas de conducta de los estudiantes, gestión de la información para consultas e impresiones referente a los estudiantes, planificación académica, calendarios y eventos, y administración del personal administrativo y docente del instituto.

3.2 Descripción general

El sistema manejará los siguientes roles de usuarios:

- Administrador
- Profesor
- Estudiante
- Responsable
- Usuario

El rol de administrador es el que tendrá privilegios de acceso para cada una de las caracteristicas, aplicaciones, menús, configuraciones y vistas dentro del sistema. No tiene accesos restringidos y es el que controla y administra el sistema en general.

El rol de profesor tiene accesos restringidos únicamente a módulos de planificaciones, estudiantes e historiales educativos.

El rol de estudiante tiene acceso limitado solo a su sesión y realizar consultas de tipo académico.

El rol de responsable es un rol especial que se le asigna a un trabajador que este a cargo de alguna área o tarea dentro de la institución. Sus privilegios se limitan al manejo del área que le fue asignada.

El rol de usuario es el mas básico de los roles. Se los identifica como a los usuarios ajenos al sistema que solo desean hacer consultas y no necesiten de realizar ningun cambio dentro del sistema.

3.3 Ambiente de instalación

Para realizar la instalación de nuestro sistema hay requerimientos mínimos que se deben cumplir como todo sistema para obtener un óptimo desempeño de la herramienta.

3.3.1 Software

- Espacio en disco: 1 Tb
- Memoria RAM: 4GB (mínimo)
- Procesado Dual Core +
- Servidor: A partir de Windows server 2008 aunque se recomienda Ubuntu server 12.0 en adelante.

Odoo proporciona a los instaladores envasados para Windows, distribuciones basados en deb (Debian Ubuntu, ...) y las distribuciones basadas en RPM (Fedora, CentOS, RHEL, ...) para las versiones de la comunidad y la empresa.

Estos paquetes configuran automáticamente para todas las dependencias (la versión de la Comunidad), pero pueden ser difíciles de mantener al día.

Paquetes oficiales de la Comunidad con todos los requisitos de dependencia pertinentes están disponibles en el servidor comunitario base. Ambos paquetes de la Comunidad sólo para empresas y se pueden descargar desde la página.

3.3.2 Hardware

Cualquier servidor será válido siempre y cuando pueda garantizar la conexión óptima de los usuarios a los servicios de Odoo. Por lo tanto, dicho servidor

tendrá que tener una infraestructura de red óptima para soportar la carga de usuarios.

3.4 Funcionalidades del sistema

- Registro, modificación o eliminación de usuarios al sistema.
- Registro, modificación o eliminación de un estudiante.
- Registro, actualización o eliminación de datos personales y socioeconómicos del estudiante en el sistema académico.
- Registro de las notas del estudiante.
- Registrará de las materias de estudio.
- Registrará de las novedades de la conducta del estudiante.
- Registrará de las novedades de la asistencia del estudiante.
- Realizará el cálculo de la nota promedio para los trimestres.
- Realizará el cálculo anual de los tres trimestres (promoción).
- Realizará consolidados de los promedios por periodos.
- Realizará un consolidado anual de las notas por periodos.
- Ingreso de planificaciones al syllabus.
- Facilitará la consulta de datos y notas del estudiante.
- Facilitará la impresión de reportes por curso y paralelo.
- Facilitará la impresión de los reportes.
- Generará e imprimirá la promoción anual del estudiante.
- Sesiones de acceso para cada usuario de acuerdo a su rol.
- Manipulación de calendarios y eventos: consultas, programación y planificación.
- Administración de un portal web con las funcionalidades e información disponible al público.
- Realizar sugerencias en cuanto al servicio, el personal e incluso el menú que está disponible dentro de la institución.

3.5 Menús

La lista de menú del sistema se encuentra detallada a continuación:

- **Planificación**
    - Parámetros y coeficientes
    - Docentes y evaluadores
    - Asignaturas - Syllabus
    - Curso
    - Oficiales alumnos
    - Registro de Alumnos
    - Registro de seminarios
    - Temas tesis, ensayos, monografías
- **Registro**
    - Directivos
    - Control de horas clases
    - Juegos de Guerra
    - Asignación de partidos JG
    - Trabajo de Dirección
    - Asignación de grupos TDD
- **Académico**
    - Calificación
    - Producto Integrador
    - AP Estadístico
    - Calificación Estadístico
    - Actitud Profesional
    - Actitud Profesional Coevaluación
    - Productividad
    - Documentos
- **Configuración**
    - Oficiales alumnos
        - Título académico
    - Docentes y evaluadores
        - Categoría
    - Actitud
        - Concepto

- Escuela
    - Grado
    - Nombre de Parámetros de Matriz
    - Promoción
    - Nato
    - Promoción del Curso
    - Religión
    - Nombre Curso
    - Especialidad
    - Formación eje
    - Títulos
    -

- **Reportes**
    - Actitud Profesional
    - Actitud Profesional Estudiantes
    - Actitud Profesional Profesores
    - Calificaciones
    - Calificaciones estudiantes
    - Calificaciones
    - Productividad
    - Productividad Estudiantes
    - Producto Integrador
    - Producto Integrador Estudiantes
    - Certificado
    - Asignaturas - Syllabus

3.6 Aplicaciones

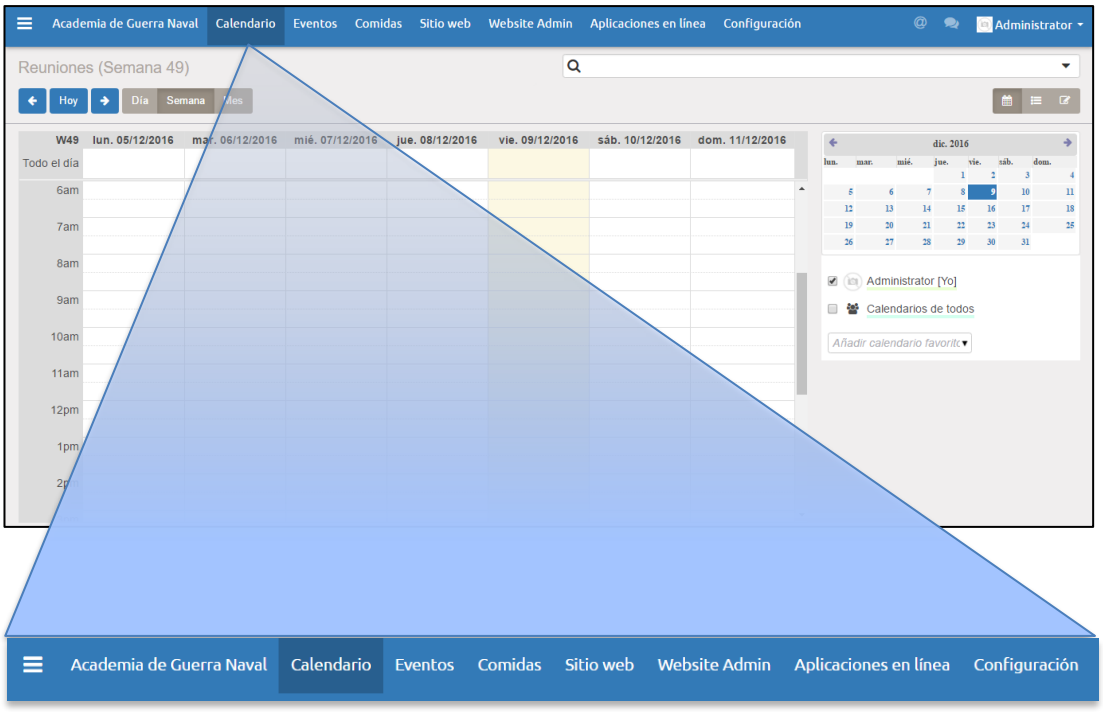


Figura 3.1 Estructura de las aplicaciones

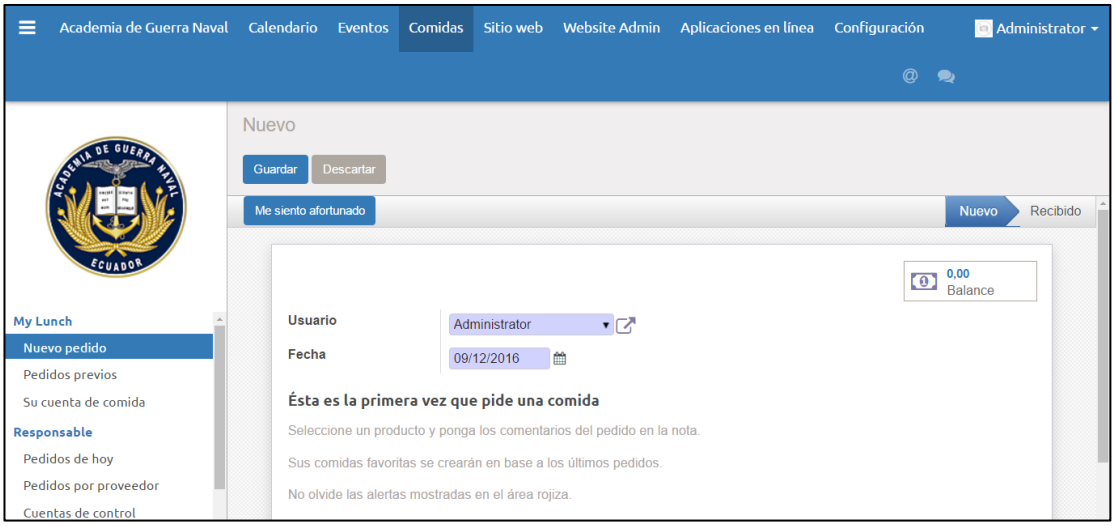


Figura 3.2 Aplicación del sistema

3.7 Vistas

3.7.1 Formulario

Vistas de formulario se utilizan para mostrar los datos de un solo registro. Su elemento raíz es **<form>.** Se componen de regular de **HTML** con componentes estructurales y semánticas adicionales.

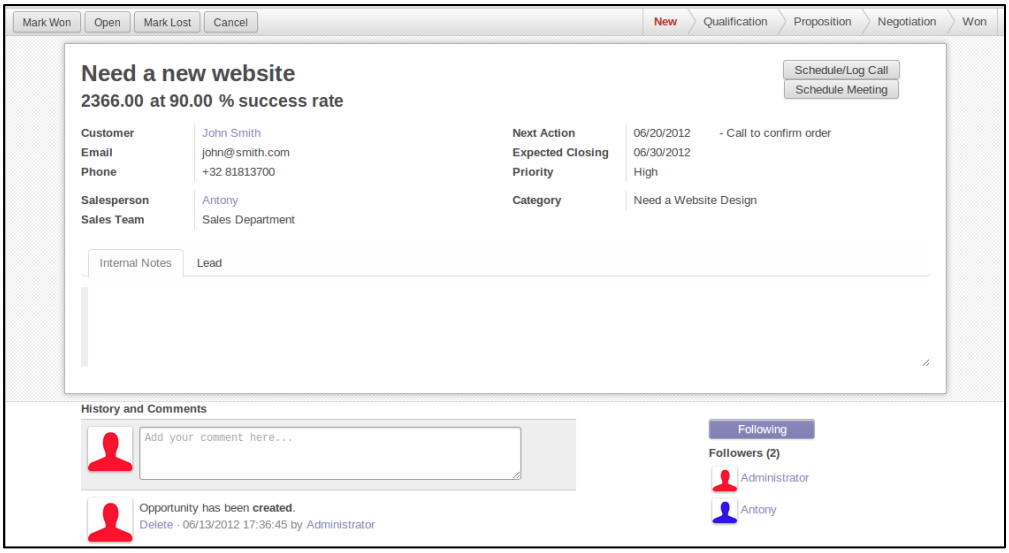


Figura 3.3: Vista Formulario.

Componentes estructurales

Los componentes estructurales proporcionan la estructura o características "visuales" con poca lógica. Se utilizan como elementos o conjuntos de elementos en las vistas de formulario.

Notebook

define una sección con pestañas. Cada ficha se define a través de un **page** elemento hijo. Las páginas pueden tener los siguientes atributos:

string (necesario) el título de la pestaña

accesskey un archivo HTML **accesskey**

attrs atributos dinámicos estándar basados en los valores de registro

group se utiliza para definir las disposiciones de columnas en las formas. De forma predeterminada, los grupos se definen las columnas 2 y la mayoría de los niños directos de grupos toman una sola columna. **Field** hijos directos de grupos mostrar una etiqueta por defecto, y la etiqueta y el campo mismo tener un colspan de 1 cada uno.

El número de columnas de un **group** se puede personalizar mediante el atributo **col**, el número de columnas tomadas por un elemento puede ser personalizado utilizando **colspan**.

Los hijos se colocan horizontalmente (trata de llenar la siguiente columna antes de cambiar de fila).

Los grupos pueden tener un atributo **string**, que se muestra como título del grupo

Newline sólo es útil dentro de los elementos **group** pone fin a la fila actual temprano e inmediatamente cambia a una nueva fila (sin llenar cualquier columna que queda de antemano)

Separator pequeño espacio horizontal, con un atributo **string** se comporta como un título de sección

Sheet se puede utilizar como un hijo directo **form** para un diseño de forma más estrecha y más sensible

Header combinado con **sheet**, proporciona una ubicación de ancho completo por encima de la propia hoja, que se utiliza generalmente para mostrar los botones de flujo de trabajo y widgets de estado

3.7.2 Kanban

La vista Kanban es una tabla Kanban visualización: muestra los registros como "tarjetas", a medio camino entre una vista de lista y no editable vista de formulario. Los registros pueden ser agrupadas en columnas para su uso en la visualización o manipulación del flujo de trabajo (por ejemplo, las tareas de gestión del trabajo o curso), o no agrupados (utilizados simplemente para visualizar los registros).

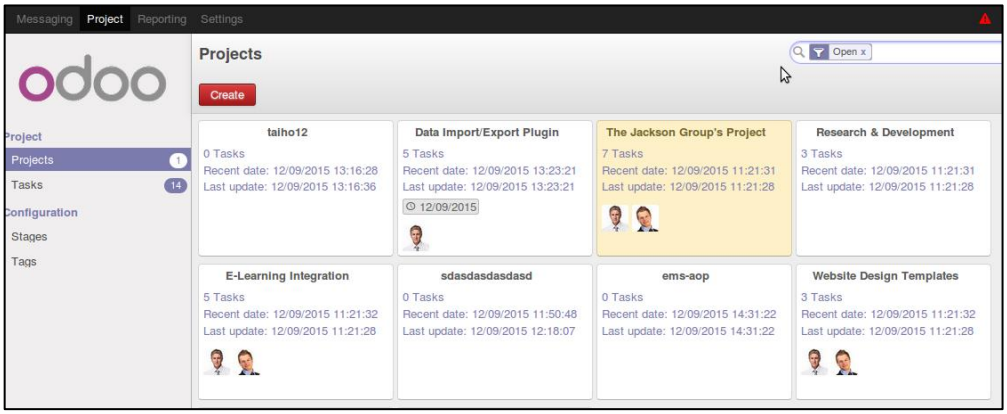


Figura 3.4: Vista Kanban.

El elemento raíz de la vista Kanban es <**kanban**>, puede utilizar los siguientes atributos:

default_group_by si la vista Kanban deben agruparse si no se especifica ninguna agrupación a través de la acción o de la búsqueda actual. Debe ser el nombre del campo agrupar por cuando se especifica lo contrario ninguna agrupación

default_order tarjetas orden de clasificación utilizados si el usuario no lo ha ordenado los registros (a través de la vista de lista)

class agrega clases HTML al elemento HTML raíz de la vista Kanban

quick_create si debe ser posible crear registros sin cambiar a la vista de formulario. De forma predeterminada, **quick_create** se habilita cuando se agrupa la vista Kanban, y los discapacitados cuando no.

Posibles elementos hijos de la vista son:

Field declara campos de agregado o para su uso en Kanban lógico. Si el campo es simplemente aparece en la vista Kanban, que no necesita ser pre-declarado.

Los posibles atributos son:

Name (necesario) el nombre del campo para ir a buscar

sum, avg, min, max,count muestra la agregación correspondiente en la parte superior de una columna de Kanban, el valor del campo es la etiqueta de la agregación (una cadena). Sólo una operación agregada por campo es compatible.

Templates define una lista de plantillas QWEB. Tarjetas definición puede ser dividido en varias plantillas para mayor claridad, pero las vistas Kanban debe definir al menos una plantilla de raíz kanban-box, que será rendido una vez para cada registro.

La vista Kanban utiliza mayormente estándar Qweb javascript y proporciona las siguientes variables de contexto:

Instance la corriente del cliente Web instancia

Widget la corriente **KanbanRecord**(), se puede utilizar para buscar un poco de meta-información. Estos métodos también están disponibles directamente en el contexto de la plantilla y no es necesario que se accede a través de widget

Record un objeto con todos los campos solicitados como sus atributos. Cada campo tiene dos atributos **value y raw_value**, el primero está formateada de acuerdo con los parámetros actuales del usuario, este último es el valor directo de un read()(a excepción de los campos de fecha y de fecha y hora que están formateados de acuerdo con la configuración regional del usuario)

3.7.3 Lista

El elemento raíz de puntos de vista lista es <**tree**>. Raíz de la vista de lista puede tener los siguientes atributos:

Editable por defecto, al seleccionar la fila de una vista de lista se abre la correspondiente vista de formulario. Los atributos editables hacen que la propia vista de lista editable en el lugar.

Los valores válidos son **top** y **bottom**, por lo que los nuevos registros aparecen respectivamente en la parte superior o inferior de la lista.

La arquitectura de la línea vista de formulario se deriva de la vista de lista. Mayoría de los atributos válidos en una vista de formulario campos 's y los botones están así aceptados por las vistas de lista, aunque puede que no tienen ningún significado si la vista de lista es no editable

default_order

anula el ordenamiento de la vista, en sustitución de orden por defecto del modelo. El valor es una lista separada por comas de los campos, se fijaron posteriormente por descordenar en orden inverso:

<árbol default_order = "secuencia, el nombre desc" >

Colors permite cambiar el color del texto de un registro basándose en los atributos del registro correspondiente.

Se define como una asignación de colores a las expresiones de Python. Los valores son de la forma:Para cada registro, los pares se prueban en orden, la expresión se evalúa para el registro y si el color correspondiente se aplica a la fila. Si no coincide con el color, utiliza el color del texto por defecto (negro). color:expr[;...]true

Color puede ser cualquiera válida unidad de color CSS.

Expr debe ser una expresión de Python evaluado con los atributos de registro actual como valores de contexto. Otros valores de contexto son uid (el id del usuario actual) y current_date(la fecha actual como una cadena de la formayyyy-MM-dd)

Fonts permite el cambio de estilo de fuente de un registro basándose en los atributos del registro correspondiente.

El formato es el mismo que para color, pero el color de cada par se sustituye por bold, itálico y subayado, la evaluación de la expresión de true aplicará el estilo correspondiente al texto de la fila. Contrariamente a **colors**, varios pares pueden igualar cada registro

create, edit,delete

permite deshabilitar la acción correspondiente en la vista estableciendo el atributo correspondiente a **false**

on_write sólo tiene sentido en una lista editable. Debe ser el nombre de un método en el modelo de la lista. El método será llamado con el id de un registro después de haber creado o editado ese registro (base de datos).

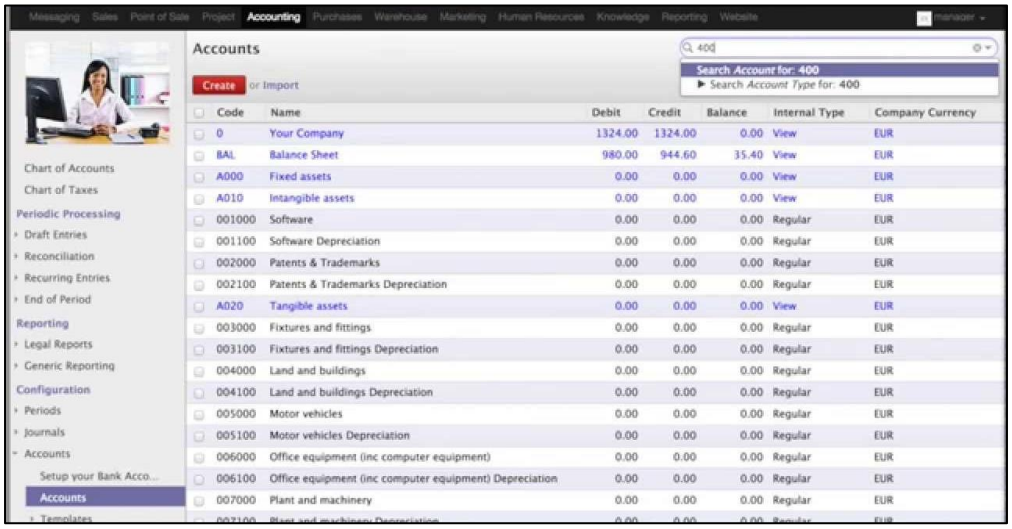


Figura 3.5: Vista lista.

El método debe devolver una lista de identificadores de otros registros para cargar o actualizar.

String traducible etiqueta alternativa para la vista en desuso desde la versión 8.0:no se visualiza más posibles elementos secundarios de la vista de lista son:

Button muestra un botón en una celda de lista

Icon icono que se usa para mostrar el botón

String si no hay icon, el texto del botón

si hay una icon, al de texto para el icono

Type tipo de botón, indica la forma en que afecta al hacer clic Odoo:

Workflow (defecto) envía una señal a un flujo de trabajo. El botón de **name** es la señal de flujo de trabajo, el historial de la fila se pasa como argumento a la señal

Object llamar a un método en el modelo de la lista. El botón de **name** es el método, que se llama con el ID de registro de la fila actual y el contexto actual.

Action carga un juego ejecutar una ir.actions, el botón de name es el identificador de la base de datos de la acción. El contexto se amplía con el modelo de la lista (tal como active_model), el historial de la fila actual (active_id) y todos los registros cargados actualmente en la lista (active_ids, puede ser sólo un subconjunto de los registros de base de datos que coinciden con el criterio de búsqueda)

Attrs atributos dinámicos basados en valores de registro.

Una asignación de atributos a los dominios, los dominios son evaluados en el contexto del registro de la fila actual, si el atributo da **True** correspondiente se encuentra en la célula.

Los posibles atributos son invisible (oculta el botón) y **readonly**(deshabilita el botón pero aun así lo muestra)

States abreviatura de invisible attrs: una lista de estados, separados por comas, requiere que el modelo tiene unstate_field y que se utiliza en la vista.

Context fusionado en el contexto de la vista cuando se realiza la llamada del botón.

Confirm mensaje de confirmación para mostrar (y para el usuario para aceptar) antes de realizar la llamada del botón

Field define una columna en la que el campo correspondiente debe mostrarse para cada registro. Puede utilizar los siguientes atributos:

Name el nombre del campo para que aparezca en el modelo actual. Un nombre dado sólo puede utilizarse una vez por visión

String el título de la columna del campo (por defecto, utiliza el stringdel campo del modelo)

Invisible obtiene y almacena el campo, pero no muestra la columna de la tabla. Necesario para campos que no deben mostrarse, pero son utilizados, por ejemplo, @colors

Groups enumera los grupos que deben ser capaces de ver el campo

Widget representaciones alternativas para la visualización de un campo. Los posibles valores de vista de lista son:

Progressbar muestra los campos como una barra de progreso.

Botón many2one reemplaza el valor del campo m2o por una marca de verificación si el campo está vacío, y una cruz si no es

3.7.4 Calendario

Figura 3.6: Vista calendario.

Vistas de la agenda de registros de visualización como eventos en un calendario diario, semanal o mensual. Su elemento raíz es <calendar>. Los atributos disponibles en la vista de calendario son:

date_start (necesario) Nombre del campo del registro de la celebración de la fecha de inicio del evento

date_stop Nombre del campo del registro de la celebración de la fecha de finalización del evento, si date_stoplos registros se convierten proporcionados móvil (a través de arrastrar y soltar) directamente en el calendario

date_delay alternativa a date_stop, proporciona la duración del evento en lugar de la fecha de finalización

Color nombre de un campo de registro a utilizar para la segmentación de color. Los registros en el mismo segmento de color se asignan el mismo color de resaltado en el calendario, los colores se asignan semi-aleatoria.

event_open_popup se abre el evento en un cuadro de diálogo en lugar de cambiar a la vista de formulario, activado por defecto

Quick_add permite la creación rápida de eventos de clic: sólo pide al usuario un name e intenta crear un nuevo evento con sólo eso y la hora del evento se ha hecho clic.

Display cadena de formato para la visualización de eventos, nombres de campo debe estar dentro de los soportes.

All**_day** nombre de un campo booleano en el expediente que indica si el evento correspondiente se marca como un día de duración (y la duración es irrelevante)

CAPÍTULO # 4

4. RESULTADOS Y PRUEBAS.

Dentro de las especificaciones y requerimientos por parte del cliente. Pudimos presentar un ambiente totalmente unificado que le permite asociar no solo a los usuarios sino también los proyectos, tareas y evento a lo largo del año estudiantil.

4.1 Resultados

4.1.1 Administración del Sitio Web

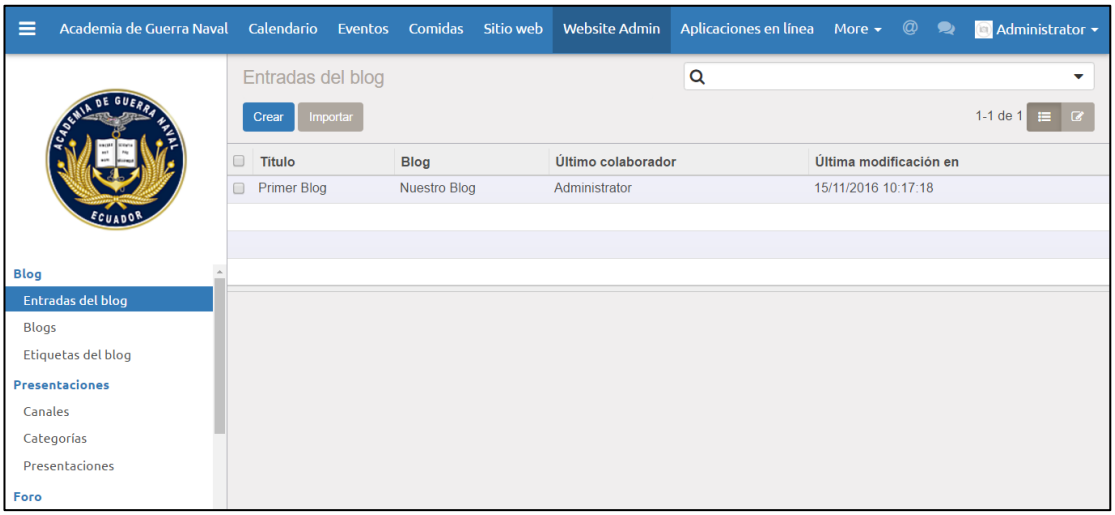


Figura 4.1: Vista preliminar del administrador del sitio web.

El Diseño es bastante sencillo, basado en los requerimientos del cliente el portal de administración de la página web es muy similar al de muchos creadores de páginas web online.

Las entradas, los menús, y las opciones generales y de administrador los encuentras en un solo lugar y/o portal.

4.1.2 Menús y áreas de trabajo dentro del módulo de AGUENA

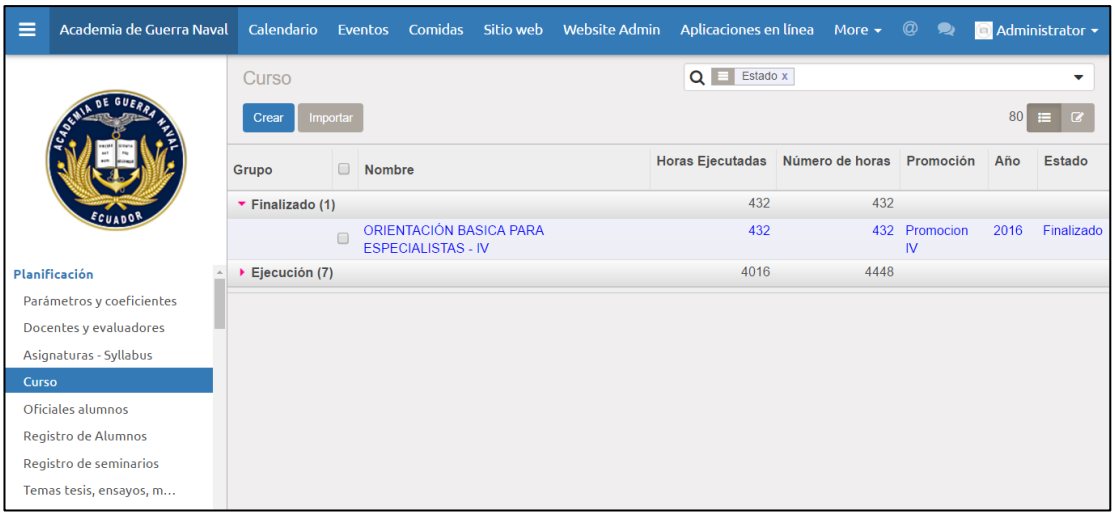


Figura 4.2: Vista preliminar de la opción Curso del menú

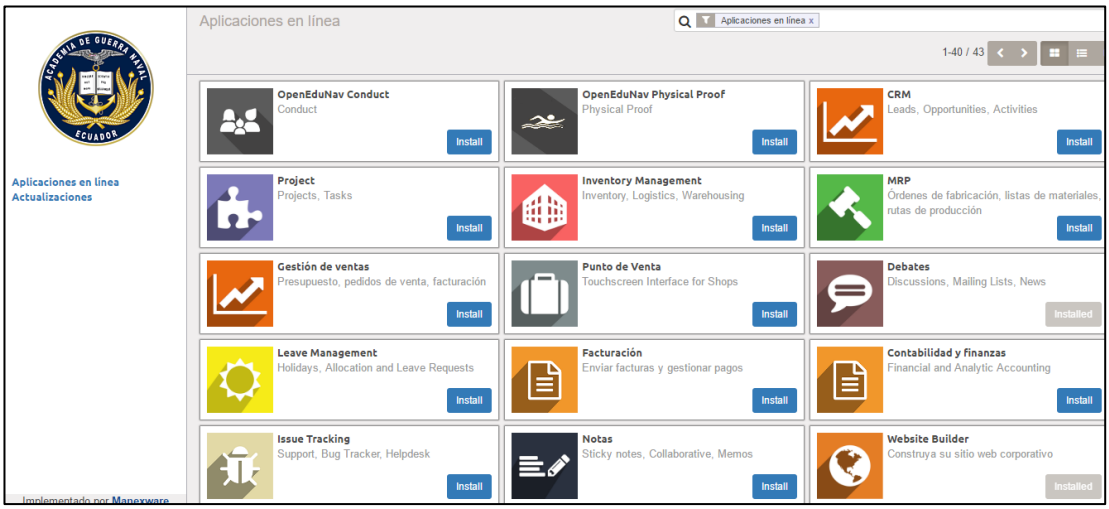


Figura 4.3: Vista preliminar de las aplicaciones que aparecen.

4.1.3 Reportes

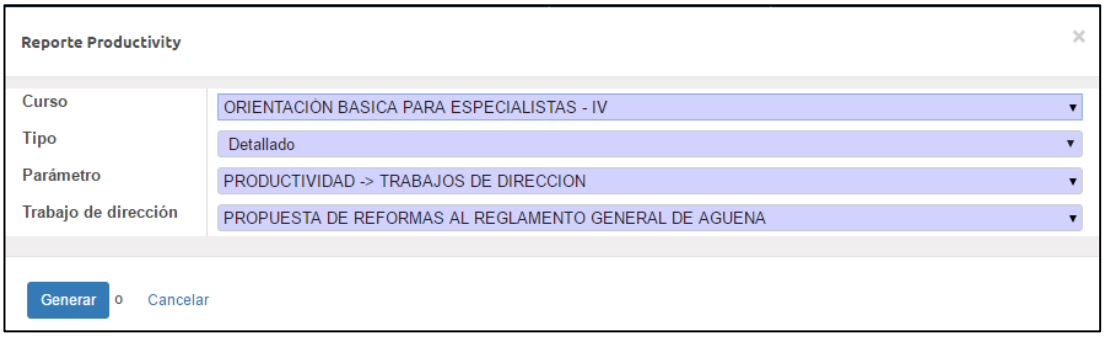
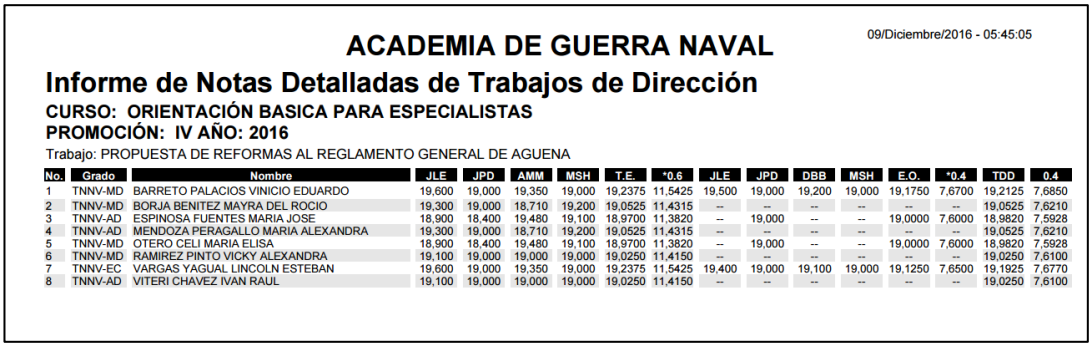


Figura 4.4: Cuadro de criterios y condiciones para reportes.



ACADEMIA DE GUERRA NAVAL

09/Diciembre/2016 - 05:45:05

Informe de Notas Detalladas de Trabajos de Dirección

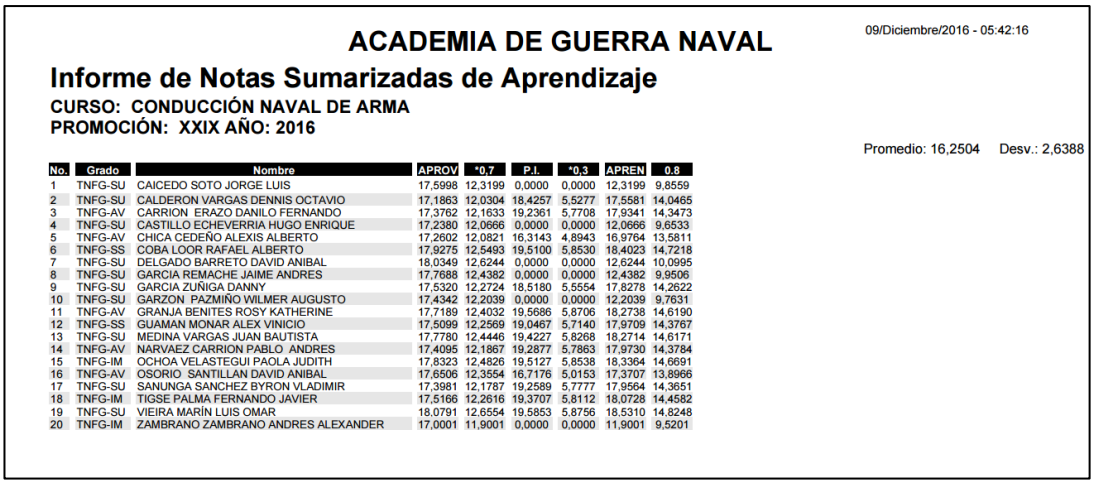
CURSO: ORIENTACIÓN BASICA PARA ESPECIALISTAS

PROMOCIÓN: IV AÑO: 2016

Trabajo: PROPUESTA DE REFORMAS AL REGLAMENTO GENERAL DE AGUENA

No.	Grado	Nombre	JLE	JPD	AMM	MSH	T.E.	*0.6	JLE	JPD	DBB	MSH	E.O.	*0.4	TDD	0.4
1	TNNV-MD	BARRETO PALACIOS VINICIO EDUARDO	19,600	19,000	19,350	19,000	19,2375	11,5425	19,500	19,000	19,200	19,000	19,1750	7,6700	19,2125	7,6850
2	TNNV-MD	BORJA BENITEZ MAYRA DEL ROCIO	19,300	19,000	18,710	19,200	19,0525	11,4315	--	--	--	--	--	--	19,0525	7,6210
3	TNNV-AD	ESPINOSA FUENTES MARIA JOSE	18,900	18,400	19,480	19,100	18,9700	11,3820	--	19,000	--	--	19,0000	7,6000	18,9820	7,5928
4	TNNV-AD	MENDOZA PERAGALLO MARIA ALEXANDRA	19,300	19,000	18,710	19,200	19,0525	11,4315	--	--	--	--	--	--	19,0525	7,6210
5	TNNV-MD	OTERO CELI MARIA ELISA	18,900	18,400	19,480	19,100	18,9700	11,3820	--	19,000	--	--	19,0000	7,6000	18,9820	7,5928
6	TNNV-MD	RAMIREZ PINTO VICKY ALEXANDRA	19,100	19,000	19,000	19,000	19,0250	11,4150	--	--	--	--	--	--	19,0250	7,6100
7	TNNV-EC	VARGAS YAGUAL LINCOLN ESTEBAN	19,600	19,000	19,350	19,000	19,2375	11,5425	19,400	19,000	19,100	19,000	19,1250	7,6500	19,1925	7,6770
8	TNNV-AD	VITERI CHAVEZ IVAN RAUL	19,100	19,000	19,000	19,000	19,0250	11,4150	--	--	--	--	--	--	19,0250	7,6100

Figura 4.5: Vista de un reporte detallado de Informe de notas.



ACADEMIA DE GUERRA NAVAL

09/Diciembre/2016 - 05:42:16

Informe de Notas Sumarizadas de Aprendizaje

CURSO: CONDUCCIÓN NAVAL DE ARMA

PROMOCIÓN: XXIX AÑO: 2016

Promedio: 16,2504 Desv.: 2,6388

No.	Grado	Nombre	APROV	*0,7	P.I.	*0,3	APREN	0.8
1	TNFG-SU	CAICEDO SOTO JORGE LUIS	17,5998	12,3199	0,0000	0,0000	12,3199	9,8559
2	TNFG-SU	CALDERON VARGAS DENNIS OCTAVIO	17,1863	12,0304	18,4257	5,5277	17,5581	14,0465
3	TNFG-AV	CARRION ERAZO DANILO FERNANDO	17,3762	12,1633	19,2361	5,7708	17,9341	14,3473
4	TNFG-SU	CASTILLO ECHEVERRIA HUGO ENRIQUE	17,2380	12,0666	0,0000	0,0000	12,0666	9,6533
5	TNFG-AV	CHICA CEDEÑO ALEXIS ALBERTO	17,2602	12,0821	16,3143	4,8943	16,9764	13,5811
6	TNFG-SS	COBA LOOR RAFAEL ALBERTO	17,9275	12,5493	19,5100	5,8530	18,4023	14,7218
7	TNFG-SU	DELGADO BARRETO DAVID ANIBAL	18,0349	12,6244	0,0000	0,0000	12,6244	10,0995
8	TNFG-SU	GARCIA REMACHE JAIME ANDRES	17,7688	12,4382	0,0000	0,0000	12,4382	9,9506
9	TNFG-SU	GARCIA ZUÑIGA DANNY	17,5320	12,2724	18,5180	5,5554	17,8278	14,2622
10	TNFG-SU	GARZON PAZMIÑO WILMER AUGUSTO	17,4342	12,2039	0,0000	0,0000	12,2039	9,7631
11	TNFG-AV	GRANJA BENITES ROSY KATHERINE	17,7189	12,4032	19,5686	5,8706	18,2738	14,6190
12	TNFG-SS	GUAMAN MONAR ALEX VINICIO	17,5099	12,2569	19,0467	5,7140	17,9709	14,3767
13	TNFG-SU	MEDINA VARGAS JUAN BAUTISTA	17,7780	12,4446	19,4227	5,8268	18,2714	14,6171
14	TNFG-AV	NARVAEZ CARRION PABLO ANDRES	17,4095	12,1867	19,2877	5,7863	17,9730	14,3784
15	TNFG-IM	OCHOA VELASTEGUI PAOLA JUDITH	17,8323	12,4826	19,5127	5,8538	18,3364	14,6691
16	TNFG-AV	OSORIO SANTILLAN DAVID ANIBAL	17,6506	12,3554	16,7176	5,0153	17,3707	13,8966
17	TNFG-SU	SANUNGA SANCHEZ BYRON VLADIMIR	17,3981	12,1787	19,2589	5,7777	17,9564	14,3651
18	TNFG-IM	TIGSE PALMA FERNANDO JAVIER	17,5166	12,2616	19,3707	5,8112	18,0728	14,4582
19	TNFG-SU	VIEIRA MARÍN LUIS OMAR	18,0791	12,6554	19,5853	5,8756	18,5310	14,8248
20	TNFG-IM	ZAMBRANO ZAMBRANO ANDRES ALEXANDER	17,0001	11,9001	0,0000	0,0000	11,9001	9,5201

Figura 4.6: Vista sumarizada del informe.

Actualmente se encuentra el sistema en un ambiente de prueba y parte del mismo ya se encuentra en producción, aunque el cliente no ha terminado de migrar su información de una plataforma a otro, ellos siguen usando el sistema anterior hasta que se dé culminado la construcción de nuevo.

4.2 Plan de entrega

Iteración primera: Se procederá a preparadas las funciones básicas relacionadas con la tarea de matricular los estudiantes y el cobro de especies.

Iteración segunda: Se procederá a preparar las funciones relacionadas con el proceso de la enseñanza-aprendizaje, como son notas, la asistencia, la conducta.

Iteración tercera: Se procederá a preparar las funciones restantes propuestas, como es el distributivo de horas y todo lo relacionado a los reportes, que den como resultado el producto a entregar.

4.3 PLAN DE ENTREGA INICIAL

4.3.1 Iteración 1:

Consta de las siguientes 5 historias de usuario:

1. Ingreso de usuario al sistema.
2. Crear curso.
3. Registro de representante.
4. Registro de estudiante.
5. Matricular estudiante.

4.3.2 HISTORIAS DE USUARIO

4.3.2.1 Ingreso de usuario al sistema.

“Un usuario autorizado ingresa a SIEAGUENA para la ejecución de las diferentes tareas inherentes a la labor educativa”.

TAREAS:

1.1. DESARROLLO DE UNA FUNCIONALIDAD DE USUARIOS.

1.2. DESARROLLO DE INTERFAZ PARA AÑADIR USUARIO.

1.3. DESARROLLO DE UNA OPCIÓN PARA REGISTRAR USUARIO.

1.4. DESARROLLO DE UNA OPCIÓN PARA MODIFICAR USUARIO.

1.5. DESARROLLO DE UNA OPCIÓN PARA ELIMINAR USUARIO.

4.3.2.2 Crear curso. “Secretaría crea un curso (año y paralelo) para matricular estudiantes.”

TAREAS:

2.1. DESARROLLO DE UNA FUNCIONALIDAD DE CURSOS.

2.2. DESARROLLO DE INTERFAZ PARA CREACIÓN DE CURSOS.

2.3. DESARROLLO DE UNA OPCIÓN PARA REGISTRAR CURSO.

2.4. DESARROLLO DE UNA OPCIÓN PARA MODIFICAR CURSO.

2.5. DESARROLLO DE UNA OPCIÓN PARA ELIMINAR CURSO.

4.3.2.3 Registro de representante.

“El representante se acerca al departamento de secretaría a matricular al estudiante a representar. Aquí se solicita primero sus datos personales.”

TAREAS:

3.1. DESARROLLO DE UNA FUNCIONALIDAD DE REPRESENTANTES.

3.2. DESARROLLO DE INTERFAZ PARA AÑADIR REPRESENTANTE.

3.3. DESARROLLO DE UNA OPCIÓN PARA REGISTRAR REPRESENTANTE.

3.4. DESARROLLO DE UNA OPCIÓN PARA MODIFICAR REPRESENTANTE.

3.5. DESARROLLO DE UNA OPCIÓN PARA ELIMINAR REPRESENTANTE.

4.3.2.4 4. Registro de estudiante. “Secretaría solicita los datos personales del estudiante a matricular.”

TAREAS:

4.1. DISEÑO DE INTERFAZ PARA EL MODO DE MOSTAR LOS ESTUDIANTES.

4.2. DESARROLLO DE UNA FUNCIONALIDAD DE ESTUDIANTES.

4.3. DESARROLLO DE INTERFAZ PARA AÑADIR ESTUDIANTE.

4.4. DESARROLLO DE UNA OPCIÓN PARA REGISTRAR ESTUDIANTE.

4.5. DESARROLLO DE UNA OPCIÓN PARA MODIFICAR ESTUDIANTE.

4.6. DESARROLLO DE UNA OPCIÓN PARA ELIMINAR ESTUDIANTE.

4.3.2.5 Matricular estudiante.

“Secretaría solicita los documentos requisitos e información personal del representante y estudiante, para matricular en un curso (año-paralelo) específico.”

TAREAS:

5.1. DISEÑO DE INTERFAZ PARA EL MODO DE MOSTRAR MATRÍCULAS.

5.2. DESARROLLO DE UNA FUNCIONALIDAD DE MATRÍCULAS.

5.3. DESARROLLO DE INTERFAZ PARA AÑADIR MATRÍCULA.

5.4. DESARROLLO DE UNA OPCIÓN PARA REGISTRAR MATRÍCULA.

5.5. DESARROLLO DE UNA OPCIÓN PARA MODIFICAR MATRÍCULA.

5.6. DESARROLLO DE UNA OPCIÓN PARA ELIMINAR MATRÍCULA.

PLAN DE ENTREGA INTERMEDIA

ITERACIÓN 2:

Consta de las siguientes 4 historias de usuario:

4.3.2.6 6. Cobrar especies.

7. Registro de datos socio-económicos.

8. Registro de notas.

9. Registro de novedades.

HISTORIAS DE USUARIO

4.3.2.7 6. Cobrar especies.

“Colecturía realiza el cobro de las especies por concepto de matriculación de cada estudiante.”

TAREAS:

6.1. DESARROLLO DE UNA FUNCIONALIDAD DE ESPECIES.

6.2. DESARROLLO DE INTERFAZ PARA AÑADIR ESPECIES.

6.3. DESARROLLO DE UNA OPCIÓN PARA REGISTRAR ESPECIE.

6.4. DESARROLLO DE UNA OPCIÓN PARA MODIFICAR ESPECIE.

6.5. DESARROLLO DE UNA OPCIÓN PARA ELIMINAR ESPECIE.

4.3.2.8 7. Registro de datos socio-económicos.

“En el departamento del COBE se solicita datos socio-económicos para ingresar al respectivo registro del estudiante.”

TAREAS:

7.1. DESARROLLO DE UNA FUNCIONALIDAD DE DATOS SOCIOECONÓMICOS.

7.2. DESARROLLO DE INTERFAZ PARA AÑADIR DATOS SOCIOECONÓMICOS.

7.3. DESARROLLO DE UNA OPCIÓN PARA REGISTRAR DATOSSOCIO-ECONÓMICOS.

7.4. DESARROLLO DE UNA OPCIÓN PARA MODIFICAR DATOS SOCIO-ECONÓMICOS.

7.5. DESARROLLO DE UNA OPCIÓN PARA ELIMINAR DATO SOCIO-ECONÓMICOS.

4.3.2.9 Registro de notas.

“Los docentes entregan en secretaría las notas por materia, curso (año/paralelo) para ser registradas en el SIAGUENA.”

TAREAS:

8.1. DESARROLLO DE UNA FUNCIONALIDAD DE NOTA GENERAL.

8.2. DESARROLLO DE INTERFAZ PARA AÑADIR NOTAS

GENERAL.

8.3. DESARROLLO DE UNA OPCIÓN PARA REGISTRAR NOTAS

GENERAL.

8.4. DESARROLLO DE UNA OPCIÓN PARA MODIFICAR NOTA

GENERAL.

8.5. DESARROLLO DE UNA OPCIÓN PARA ELIMINAR NOTA

GENERAL.

8.6. DESARROLLO DE INTERFAZ PARA INGRESO DE NOTAS POR

AÑO LECTIVO, CURSO (AÑO-PARALELO), MATERIA.

8.7. DESARROLLO DE UNA OPCIÓN PARA EL INGRESO Y/O

MODIFICACIÓN DE NOTAS POR AÑO LECTIVO, CURSO,

MATERIA.

4.3.2.10 Registro de novedades.

“Inspección registra novedades de asistencia y conducta de los estudiantes.”

TAREAS:

9.1. DESARROLLO DE UNA FUNCIONALIDAD DE NOVEDADES.

9.2. DESARROLLO DE INTERFAZ PARA AÑADIR NOVEDAD.

9.3. DESARROLLO DE UNA OPCIÓN PARA REGISTRAR NOVEDAD.

9.4. DESARROLLO DE UNA OPCIÓN PARA MODIFICAR NOVEDAD.

9.5. DESARROLLO DE UNA OPCIÓN PARA ELIMINAR NOVEDAD.

PLAN DE ENTREGA FINAL

ITERACIÓN 3:

Consta de las siguientes 6 historias de usuario:

10. Elaborar reportes de estudiantes.

11. Elaborar reportes trimestrales.

12. Elaborar promoción de año.

13. Registro de profesor.

14. Registrar distributivo de materias.

15. Registrar notas para graduación.

HISTORIAS DE USUARIO:

10. Elaborar reportes de estudiantes.

“Secretaría genera reportes de estudiantes por curso para el registro de notas, previo al ingreso en el AGUENA.”

“También genera reportes de estudiantes con parámetros de conducta para el registro de notas de conducta, previo al ingreso en AGUENA.”

TAREAS:

10.1. Diseño interfaz para selección de año lectivo y curso.

10.2. Desarrollo de una opción para generación de reportes.

4.3.2.11 Elaborar reportes trimestrales.

“Secretaría genera reportes trimestrales de los estudiantes, por curso, con las notas del aprovechamiento de cada materia.”.

TAREAS:

11.1. DISEÑO INTERFAZ PARA SELECCIÓN DE AÑO LECTIVO, TRIMESTRE, CURSO.

11.2. DESARROLLO DE PROGRAMA GENERADOR DE REPORTES.

4.3.2.12 Elaborar promoción de año.

“Secretaría genera reportes de los estudiantes al final del año lectivo, conel aprovechamiento trimestral y anual por cada materia y a la vez globalde todas las materias. También se incluye la conducta global.”

TAREAS:

12.1. DISEÑO INTERFAZ PARA SELECCIÓN DE AÑO LECTIVO, CURSO Y ESTUDIANTE.

12.2. DESARROLLO DE UNA OPCIÓN GENERADOR DE **PROMOCIÓN DE AÑO.**

4.3.2.13 profesor.”

TAREAS:

13.1. DESARROLLO DE UNA FUNCIONALIDAD DE PROFESOR.

13.2. DESARROLLO DE INTERFAZ PARA AÑADIR PROFESOR.

13.3. DESARROLLO DE UNA OPCIÓN PARA REGISTRAR PROFESOR.

13.4. DESARROLLO DE UNA OPCIÓN PARA MODIFICAR PROFESOR.

13.5. DESARROLLO DE UNA OPCIÓN PARA ELIMINAR PROFESOR.

4.3.2.14 Registrar distributivo de materias.

“Secretaría registra el distributivo de trabajo para el año lectivo, tomando en cuenta al profesor, las materias a dictar y en los cursos a impartir.”

TAREAS:

14.1. Desarrollo de una funcionalidad de distributivo.

14.2. Desarrollo de interfaz para añadir distributivo.

14.3. Desarrollo de programa para registrar distributivo.

14.4. Desarrollo de una opción para modificar distributivo.

14.5. Desarrollo de una opción para eliminar distributivo.

14.6. Desarrollo de interfaz para reporte de distributivo.

4.3.2.15 Registrar notas para graduación.

“Secretaría registra las notas para graduación de los estudiantes de 3ro de bachillerato. Las notas son las correspondientes a las promociones de 8vo, 9no, 10mo, 1ro, 2do, 3ro, exámenes de grado y trabajo comunitario.”

TAREAS:

15.1. desarrollo de una funcionalidad de notas para3ro de bachillerato.

15.2. desarrollo de interfaz para registrar y/o modificar notas.

15.3. desarrollo de una opción para registrar y/o modificar notas.

15.4. desarrollo de interfaz para el ingreso de parámetros para generación de reportes.

15.5. desarrollo de una opción para generar reportes.

CONCLUSIONES Y RECOMENDACIONES

El sistema SIEAGUENA se convierte en una herramienta de software importante porque ha permitido unificar los datos de los estudiantes y a la vez eliminar la repetición de los mismos. Ha permitido la integración de los usuarios en un entorno web y dar la facilidad de navegar y acceder a la información académica. Esto es una funcionalidad importante de la aplicación.

También el SIEAGUENA ha permitido definir los diferentes procesos académicos correspondientes a cada departamento, como son el proceso de planificación, de cobro de especies, la atención en el estudiante y el registro de novedades de conducta e inspección.

Además, el SIEAGUENA ha permitido identificar y definir a los usuarios del sistema, los perfiles que tendrá cada uno de ellos y a la vez sus limitaciones.

El sistema SIEAGUENA presenta la información disponible en el sistema a cada usuario respectivamente, para que pueda administrarlo, ya sea añadiendo, modificando, eliminado o solamente mostrándolo.

La metodología XP que se usó para el desarrollo del SIEAGUENA es la ideal, ya que permite crear versiones de la aplicación e ir probando y revisando junto con el usuario.

BIBLIOGRAFÍA

[1] Domatrix (2012, Abril). OpenERP(1era) [Online]. Disponible en: http://www.domatix.com/odoo-openerp/

[2] Postgresql, (2010, Octubre). Todo sobre PostgreSql[online]. Disponible en: http://www.postgresql.org.es/sobre_postgresql

[3] Odoo Community (2015, January) Documentación de Odoo[Online]. Disponible en:

https://www.odoo.com/documentation/8.0/reference/views.html

[4] Potencier. (2012). El tutorial Jobeet. Disponible en línea: http://www.librosWeb.es/jobeet_1_3. [30] Pressman R. (2002). Ingeniería del Software: Un enfoque práctico. Quinta Edición. Disponible en: http://fcqi.tij.uabc.mx/usuarios/luisgmo/data/capitulo%202%20pressman.pdf

[5] Revista Iberoamericana. (2000). Tic en Educación. Disponible en:

https://snt151.mail.live.com/default.aspx?id=64855#fid=flinbox.

[6] RIBALTA, A. (1993). Utilización de las técnicas de computación en el proceso docente. Varona (Ciudad de la Habana).

[7] Ricardo, C. (2001), Amenaza informática Disponible en:

https/www.el software.com.mx encontrado el 17 de abril de 2014

[8] Riera Barzallos, P. (1998). Ventajas de la informatización del servicio de consultas de los usuarios. El profesional de la información, 3. Recuperado:

http://www.elprofesionaldelainformacion.com/contenidos/1998/diciembre/ventajas_de_la_informatizacion_del_servicio_de_consultas_de_los_usuarios.html

ANEXOS

[1] Guido Van Rossum. (2016, Junio). El Tutorial de Python (1º) [Documento]. Disponible en: http://docs.python.org.ar/tutorial/pdfs/TutorialPython3.pdf