

Sistema de Matrícula Online

***ONLINE***

***ENROLLMENT***

***SYSTEM***

OES

**UNIVERSIDAD NACIONAL FEDERICO VILLARREAL**

**FACULTAD DE INGENIERÍA ELECTRÓNICA E INFORMÁTICA**

**INTEGRANTES:**

* **ECHEANDIA PINTO, CHRISTIAN**
* **GARGATE JAVIER, JOSSELIN**
* **GONZALES TELLO, FRANCIS**
* **HERNANDEZ MAVILA, PETER**
* **HURTADO MARINO, DANIEL**
* **JAVE FARFAN, YENIFFER**

**CURSO:**

**PROYECTOS INFORMATICOS**

**DOCENTE:**

**ING. EDWARD FLORES M.**

**CICLO:**

**IX**

**2012 - I**

# INDICE

Índice………………………………………………………………………………………… 1

Introducción……………………………………………………………………………… 4

Objetivos………………………………………………………………………………...… 5

Objetivo General………………………………………………………………. 5

Objetivo Específico……………………………………………………………. 5

Descripción………………………………………………………………………………… 5

Justificación……………………………………………………………………………….. 6

Beneficios…………………………………….…………………………………………….. 6

Equipo de Trabajo………..…………………………………………………………….. 7

Alcance………..…………………………………………………………………………….. 9

Contemplados…………………………………………………………………… 9

No contemplados………………………………………………………………. 9

Factibilidad…………………………………………………………………………………. 10

Factibilidad Técnica………………………………….……….………………. 10

Factibilidad Operacional………..………………….………………………. 10

Factibilidad Económica………………………………….….………………. 10

Cronograma ……………………………………………………………………………….. 11

Itinerario del Diagrama de Gantt……………………………………….. 11

Diagrama de Gantt……………………………………………………………. 12

Escala de Tiempo………………………………………………………………. 13

Costos………………………………………………………………………………. 14

Análisis y diseño del Sistema del OES …………………………………..…….. 15

Casos de Uso de Negocio ………………………………..………...……… 17

Casos de Uso de Sistema ………………………………….………………. 18

Diagrama de Actividades (Administrador)…………………….……. 19

Diagrama de Actividades (Proceso Ingresantes)……………… 20

Diagrama de Actividades (Proceso Regular)……………………. 21

Diferencias entre el Sistema de Matrícula actual y el OES………… 22

Diagramas de flujo (Proceso actual)………………………………… 22

Diagramas de flujo (Proceso mediante OES)……………………. 25

Tabla de diferencias entre el proceso de matrícula

actual y el proceso mediante el OES………………………………… 27

Diseño de la base de datos………………………………………………………… 28

SCRIPT generado por el phpMyadmin……………………………… 32

Software utilizados en el OES ……………………………………..…………….. 35

GitHub …………………………………..…………………….. …………..…… 35

WampServer ……………………………….. ………………………..………. 37

NetBeans ………………………………………………………………..………. 39

Apache………………………………………………………………….…….…… 41

JavaScript ………………………………………………………………………… 43

JQuery ……………………………………………………………………………… 44

Ajax ………………………………………………………………………….……… 44

Git …………………………………………………………………………………… 45

Firebug …………………………………………………………….……………… 49

Google Developers Tools ……………………………………………….… 49

Imágenes del OES ………………………………………………………………….….. 51

Bienvenida al OES ……………………………………………………….. ……. 51

Datos del Alumno ……………………………………………………………. 52

Presentación del Voucher de Pago …………………………………… 53

Matrícula Online ……………………………………………………………… 54

Cursos a Matricularse ……………………………………………………… 55

Confirmación de Matrícula ………………………………………………. 56

Documentos a Descargar ………………………………………………….. 57

Anexos……………………………………………………………………………………….. 58

Entrevista con la Srta. Irma (Encargada de la OSA)……………. 58

Glosario de términos………………………………………………………... 60

**INTRODUCCIÓN**

*En esta oportunidad, presentaremos la última fase del desarrollo del Sistema de Matrícula Online para la Facultad de Ingeniería Electrónica e Informática (FIEI) de la Universidad Nacional Federico Villarreal, de acuerdo al cronograma planteado inicialmente.*

*Como último entregable, en este documento se hará una recapitulación de todo lo visto en los 3 entregables desarrollados anteriormente detallando desde el objetivo del OES hasta los software que se han utilizado para llevar a cabo este proyecto.*

*El enfoque es detallar el sistema y no presentar un marco teórico de lo que es cada uno de los temas presentados desarrollando así una propuesta concreta de un sistema que se quiere poner en marcha.*

*El sistema se analiza y expone de forma general pero a la vez se consideran muchos detalles y aspectos específicos del funcionamiento. Muchos de los beneficios de este sistema son indiscutibles, por lo que se afirma que es una gran opción para cualquier institución educativa que desee automatizar su sistema de matrículas.*

1. **OBJETIVOS**
2. **OBJETIVO GENERAL:**

Implementar una solución que permitan procesar de forma rápida y eficiente el proceso de matrícula de la Facultad de Ingeniería Electrónica e Informática de la Universidad Nacional Federico Villarreal considerando necesario para dicho proceso el desarrollo de un sistema de gestión de matrícula.

1. **OBJETIVOS ESPECÍFICOS:**

* Ofrecer mediante el sistema de matrícula: rapidez, seguridad de datos e información a tiempo real de la UNFV – FIEI.
* Proponer un sistema de Gestión de Matricula orientado a mejorar la atención de los alumnos.
* Brindar la información oportuna y confiable en el momento que se requiera.

1. **DESCRIPCIÓN**

El OES (Online Enrollment System) es un proyecto que se viene desarrollando por un grupo de estudiantes de ingeniería informática, de la facultad de ingeniería electrónica e informática de la UNFV que busca implantar un sistema de matrícula online para dicha facultad.

El proyecto está comprendido por 5 fases definidos detalladamente en el cronograma de actividades adjuntos en este documento:

**Planificación:** esta es la fase inicial del proyecto el cual nos servirá como guía durante el proceso del mismo. Para ello desarrollaremos puntos esenciales como el alcance, cronograma, costos y factibilidad del proyecto detallados en el actual documento.

**Análisis:** en esta fase de análisis, nos encargaremos del levantamiento de información acerca del proceso de matrícula actual la cual nos ayudará en el proceso de análisis y diseño del sistema.

**Diseño:** Con la información obtenida en la fase de análisis, procederemos a desarrollar el entorno gráfico de la aplicación, en este caso la creación de las páginas web con las cuales interactuarán los estudiantes a la hora de matricularse.

**Desarrollo:** en la fase desarrollo, el administrador de base de datos y el programador cumplirán con las funciones encargadas por el Jefe de proyecto para la construcción lógica del sistema.

**Instalación:** Finalmente se procederá a alojar el sistema en un servidor web y proceder a las respectivas pruebas con el usuario y en caso de que haya alguna falla en el sistema se dará paso a proceder con los cambios necesarios, finalizando con la documentación y la capacitación a los usuarios finales.

1. **JUSTIFICACION**

Nuestra Facultad tiene ya casi 15 años de formación, y teniendo a su cargo carreras como la nuestra (Ing. informática) que van a la vanguardia con la tecnología debería contar con un sistema de matrícula online como en todas las grandes universidades en el Perú y en el extranjero.

El sistema tradicional de matrícula resulta prácticamente obsoleto con todas las herramientas que existen hoy en día, es por ello que este proyecto busca implantar un sistema de matrícula online en el cual beneficiará tanto a los estudiantes como a los responsables de la matrícula en cuanto a tiempo se refiere, por ejemplo los estudiantes podrán matricularse desde la comunidad de su casa, una cabina de internet o en donde se encuentre, sin la necesidad de ir a la facultad y esperar su turno formando largas colas.

1. **BENEFICIOS**

Entre los beneficios de implementar el sistema de tenemos:

* Facilidad de matricularse desde casa o desde cualquier sitio sin tener que hacer colas en la Facultad.
* El sistema te ofrece las asignaturas que puedes cursar, y tú solo tienes que indicarle cuáles te interesan, por lo que te evitas sorpresas posteriores.
* Posee facilidad de uso, ya que es un medio guiado, con sencillas pantallas que te ayudarán paso a paso, permitiéndote retroceder y cambiar cualquier dato antes de finalizar tu matrícula.
* Agilidad en el proceso de matrícula.
* Evitar gastos innecesarios en material de impresión o carpetas.

1. **EQUIPO DE TRABAJO**

Son las personas o el recurso humano con las capacidades y conocimientos necesarios para desarrollar el proyecto informático y que será dirigido por el jefe de proyecto.

Cabe mencionar que es muy importante que el equipo de trabajo conozca muy bien con anterioridad la planificación del proyecto, para desarrollarlo y entregarlo con el tiempo suficiente para cumplir con los plazos.

1. **JEFE DE PROYECTO:**

Hurtado Marino, Daniel André

Es la persona encargada de planificar, dirigir, controlar y coordinar el proyecto informático hasta su término e implantación.

Su misión está basada en 2 aspectos. Por un lado está el aspecto **técnico** en donde se deben concretar los objetivos, gestionar los recursos, etc., y por otro lado está la **gestión** de los recursos humanos, materiales, la relación con el receptor del proyecto informático.

1. **ANALISTA FUNCIONAL:**

Echeandía Pinto, Christian

Jave Farfán, Yeniffer Tatiana

Son los encargados del análisis del proyecto, tienen una visión más general que los demás componentes. Sus funciones son:

* Estudio de la situación actual.
* Estudio de los objetivos.
* Estudio de viabilidad.
* Estudio del impacto hardware/software.
* Definición de los requisitos del proyecto.
* Definición de circuitos.
* Identificación de necesidades de información e intercambio.
* Identificación de procesos.
* Elaboración de la documentación funcional.
* Diseño del modelo de entidades.
* Diseño de flujo de datos.
* Diseño lógico de ficheros y tablas.
* Diseño de conversaciones.

1. **PROGRAMADOR:**

Gonzales Tello, Francis Angelino

Gargate Javier, Josselin

La labor a realizar es la construcción de los prototipos, codificación de las interfaces del sistema y colaborara en todo momento con el analista, en las diversas funciones que realice.

Ellos se encargaran de escribir, depurar y mantener el código fuente del sistema, es decir, del conjunto de instrucciones que ejecutará el hardware de la computadora para realizar una tarea determinada.

Sus funciones son:

* Estructuración lógica del programa.
* Codificación.
* Elaboración de pruebas unitarias.
* Colaboración en pruebas de conjunto.

1. **ADMINISTRADOR DE BASE DE DATOS:**

Hernández Mavila, Peter

La tarea del DBA es crear la base de datos en sí y poner en vigor los controles técnicos necesarios. El DBA se encargará también de garantizar el funcionamiento adecuado del sistema y de proporcionar otros servicios de índole técnica relacionados.

La responsabilidad general del DBA es facilitar el desarrollo y el uso de la Base de Datos dentro de las guías de acción definidas anteriormente.

El DBA es responsable primordialmente de:

* Administrar la estructura de la Base de Datos
* Administrar la actividad de los datos
* Administrar el Sistema Manejador de Base de Datos
* Establecer el Diccionario de Datos
* Asegurar la confiabilidad de la Base de Datos
* Confirmar la seguridad de la Base de Datos

1. **ALCANCE**

A continuación se detalla la información que ha sido recabada de las diferentes reuniones que se han celebrado con los estudiantes y administrativos de la UNFV-FIEI, de acuerdo a los testimonios de los estudiantes de la UNFV-FIEI es demasiado tedioso realizar los diferentes pasos del proceso de matrícula actual por motivo de tiempo y dinero.

Es por ello que nuestra solución mejorará el actual proceso por un proceso online, permitiendo que los estudiantes acudan a la UNFV-FIEI solo a recoger su constancia de matrícula.

* **CONTEMPLADOS**

1. Crear el O.E.S. (Online Enrollment System) como una contribución que ayudará a la FIEI a agilizar el proceso de matrícula a los estudiantes.
2. Compatibilidad con cualquier navegador.
3. Compatibilidad para cualquier sistema operativo
4. Solo a un plan curricular 2001.
5. Elegir turno.
6. Garantizar la integridad del alumno en el proceso de matrícula online.
7. El uso adecuado de los recursos informáticos que se disponen en la facultad.
8. Servir como modelo para que otras facultades implementen este nivel de servicios.

* **NO CONTEMPLADOS**

1. Sistema multi – idioma.
2. Verificación de pago con el banco
3. Pago online
4. Verificación multicurrícula.
5. Actualizar datos del alumno.
6. Asesores para el proceso de matrícula online.
7. Reserva de matrícula.
8. **FACTIBILIDAD**
9. **FACTIBILIDAD TÉCNICA**

Utilizaremos PHP y MySql que son software libre y muy portables es decir es sumamente sencillo poder migrar el sistema de un servidor a otro si ese fuese el caso además como IDE utilizaremos Netbeans que también es una herramienta de software libre esto implica que el gasto en cuanto a licencias para el desarrollo del software es cero además el sistema completo funcionará en internet esto significa que para acceder simplemente necesitamos usar un navegador de internet lo cual nos permite acceder desde cualquier sistema operativo e incluso desde dispositivos móviles.

1. **FACTIBILIDAD OPERACIONAL**

Es completamente posible el uso de nuestro sistema pues no tiene procesos complejos para los administradores del sistema ni para las personas que lo utilizarán. Lo que pretendemos con nuestro software es sustituir el proceso arcaico que se viene utilizando actualmente como es el proceso presencial de matrícula el cual posee muchas desventajas. Al implantar nuestro sistema los alumnos y encargados de este proceso tendrán la información disponible online 24/7.

1. **FACTIBILIDAD ECONÓMICA**

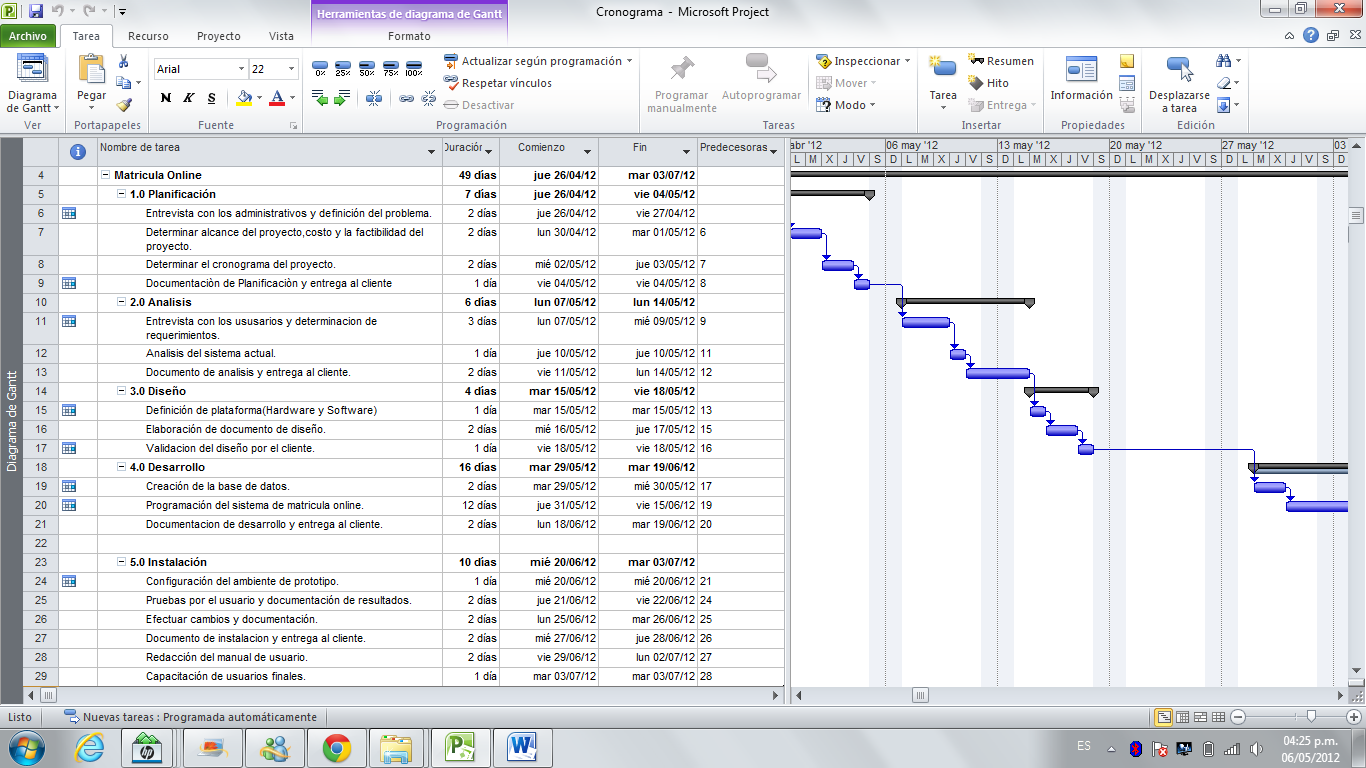
Para determinar la factibilidad económica hemos realizado el análisis de costo-beneficio.

Con este sistema se obtiene un excelente rendimiento y ahorro, debido a que no es necesario gastar en costosos recursos de hardware ni en licencias de software debido a que todo estará basado en internet. Entre los cuantificables o tangibles se tienen el mantenimiento del servidor y el pago que reciben las personas que trabajan en el desarrollo del sistema.

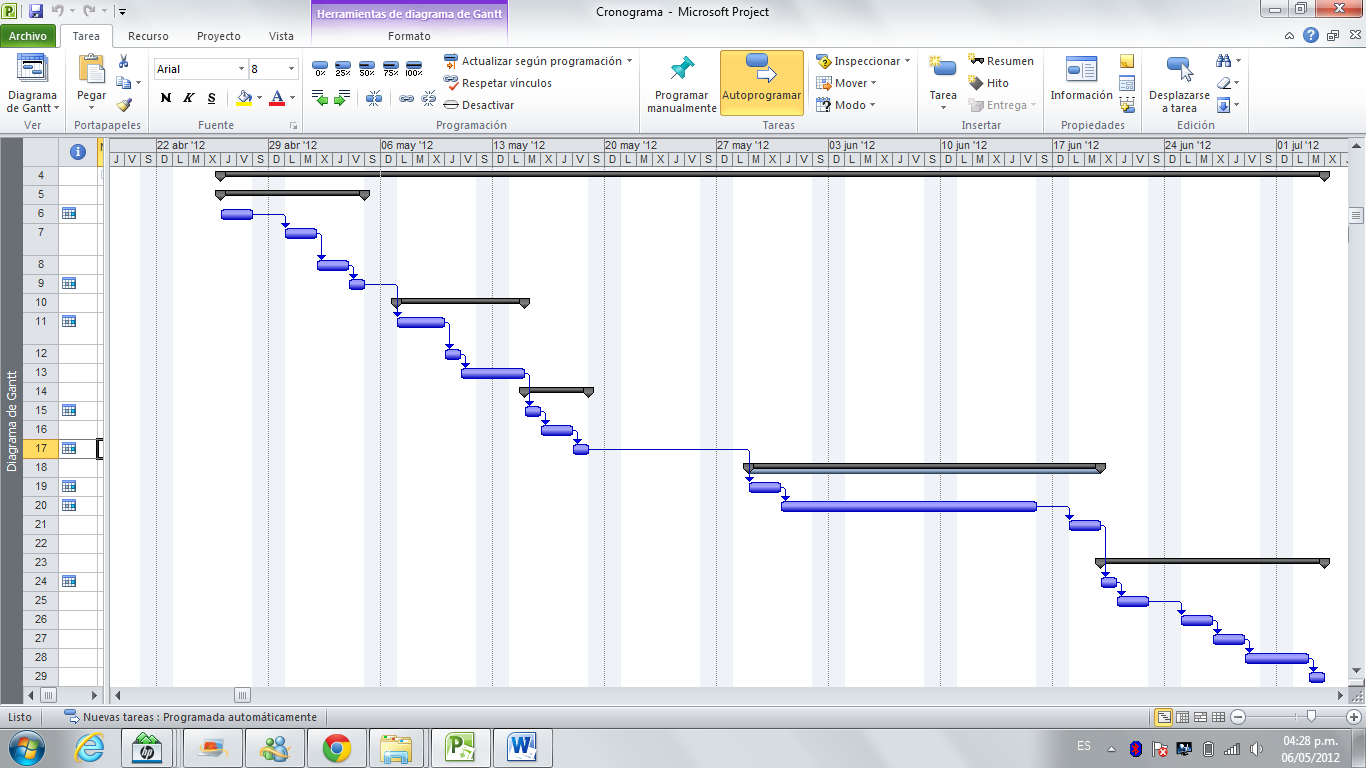
1. **CRONOGRAMA**

Para la realización del cronograma se ha utilizad una popular herramienta gráfica cuyo objetivo es mostrar el tiempo de dedicación previsto para diferentes tareas o actividades a lo largo de un tiempo total determinado, aquella herramienta es llamada Diagrama de Gant y se muestra a continuación:

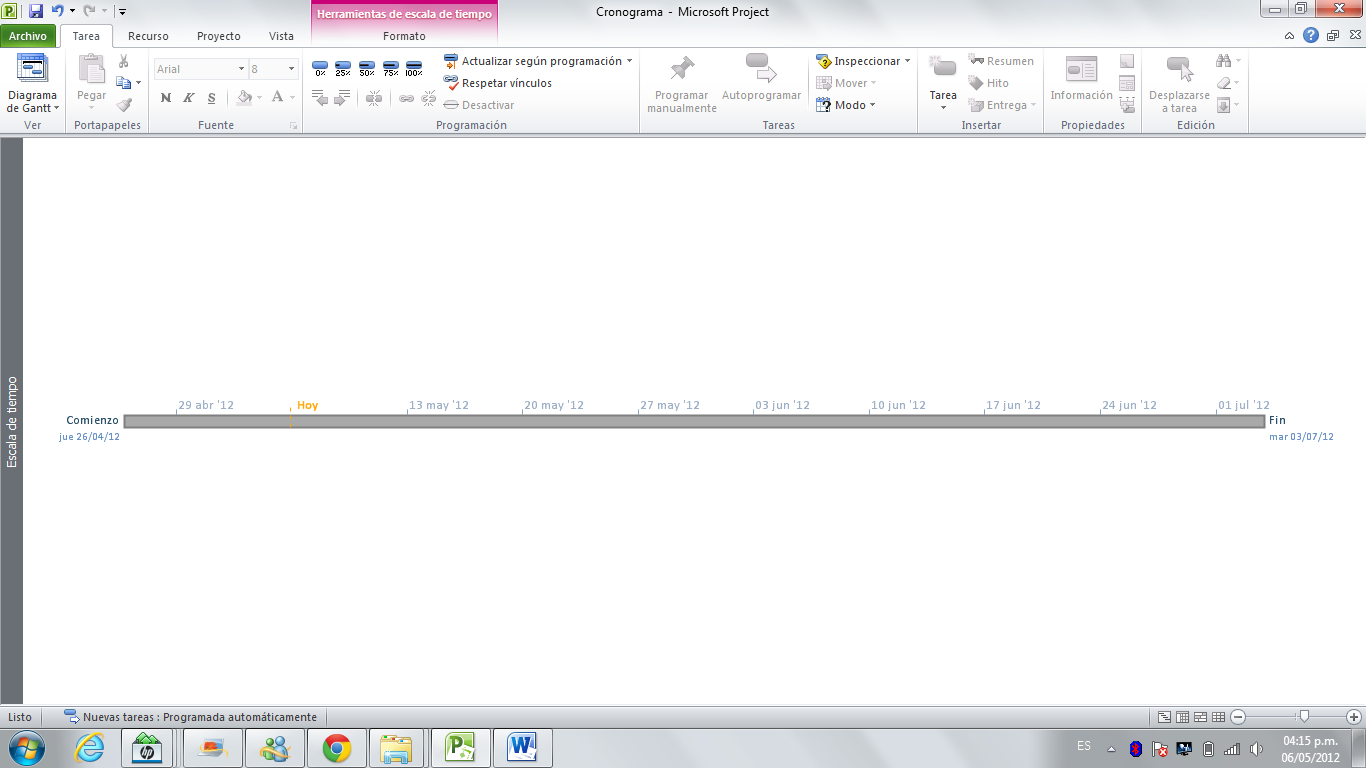
* **ITINERARIO DEL DIAGRAMA DE GANTT**



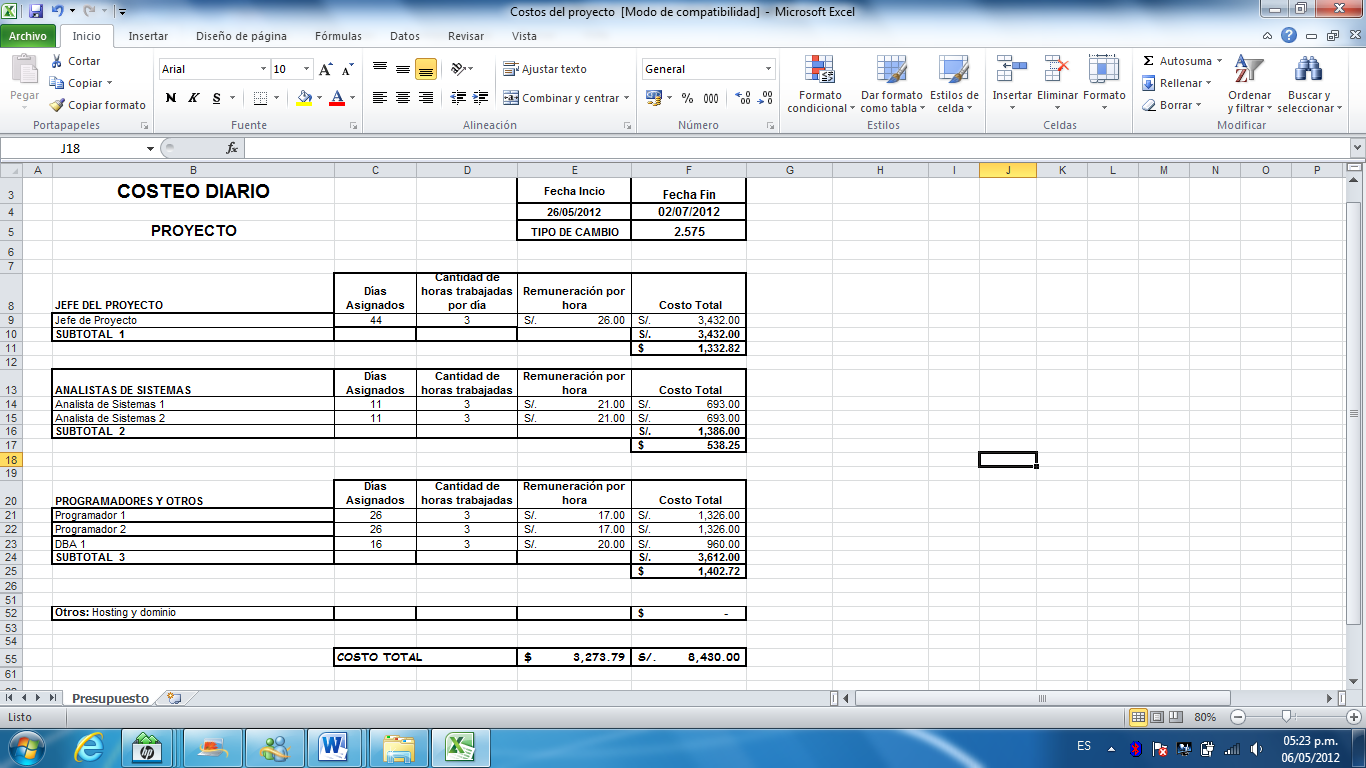
* **DIAGRAMA DE GANTT**



* **ESCALA DE TIEMPO**



1. **COSTOS**



1. **ANÁLISIS Y DISEÑO DEL SISTEMA DE MATRÍCULA ONLINE**

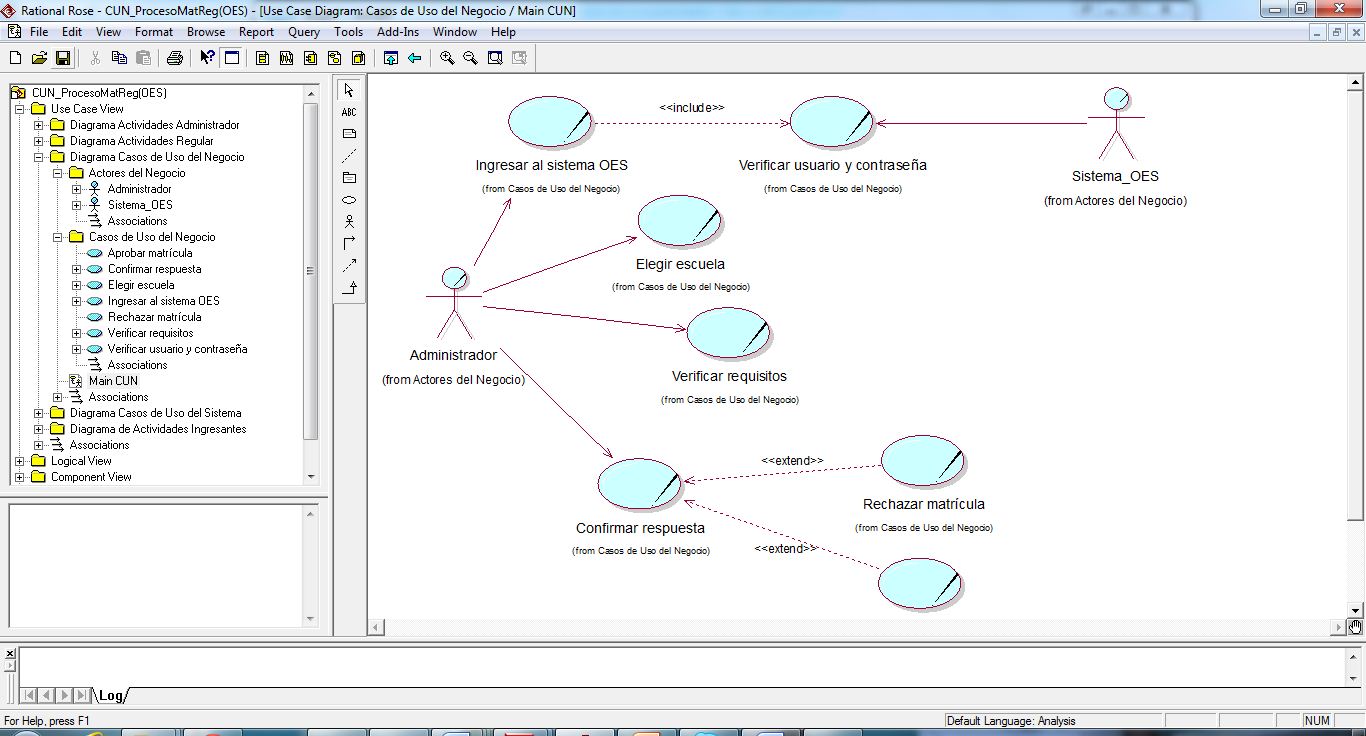
**El análisis de un sistema de información se realiza una vez entendido el problema, se procede a determinar lo que se debe hacer para lograr una óptima solución, especificando las funciones con base en el estado actual de la organización. De esta forma es necesario adelantar las siguientes actividades: profundizar y ampliar la definición del problema, analizar las áreas problema, documentos utilizados, esquema organizacional y funcional, las comunicaciones y sus flujos, el sistema de control y evaluación, análisis y evaluación del sistema actual, determinar los recursos humanos, técnicos y económicos necesarios para desarrollar el sistema.**

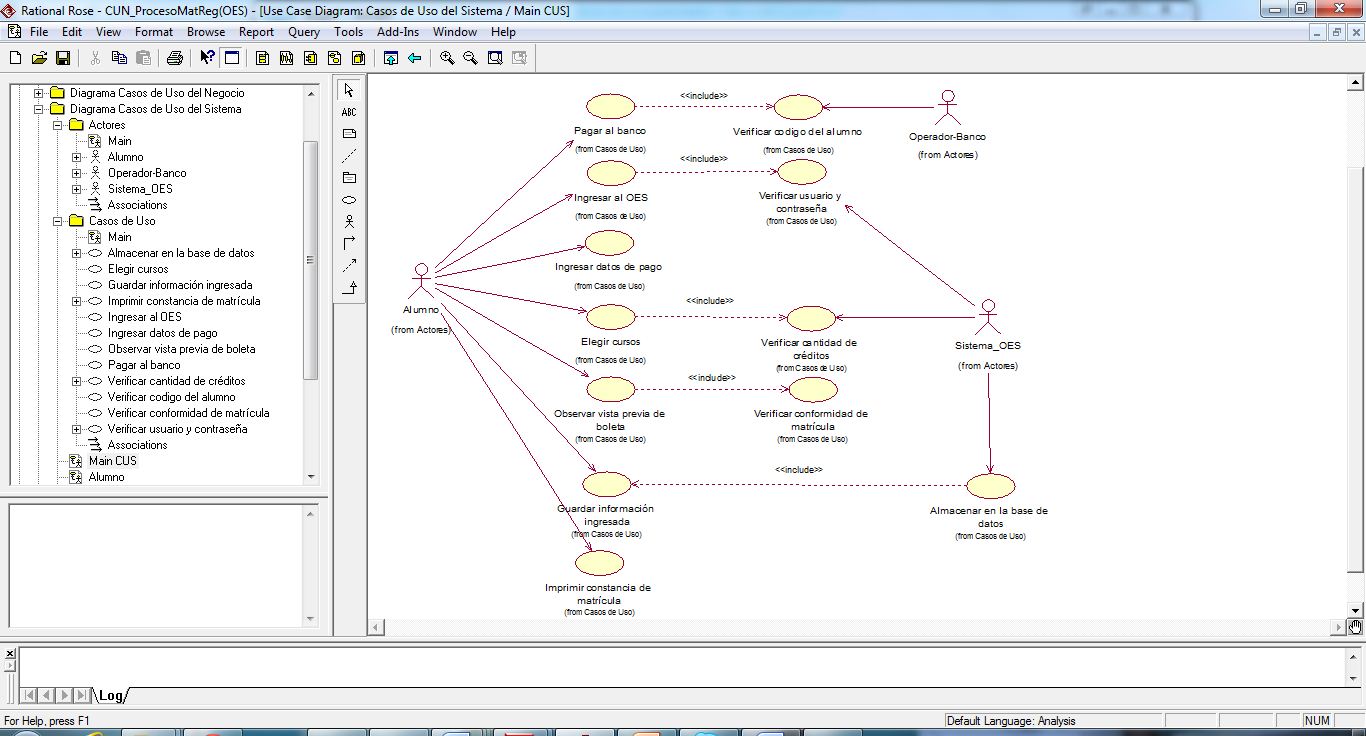
**Así el diseño de sistema de información comprende las siguientes actividades: esquemas físicos de organización de datos, modelo estructural de los datos, directorio de datos, procedimientos lógicos y cálculos requeridos, determinación de los controles necesarios, diseños de formato para captura de datos, diseño de formato para salida de información tanto en pantalla como por impresora, determinación del software y hardware necesarios para el desarrollo del sistema.**

**Características de UML:**

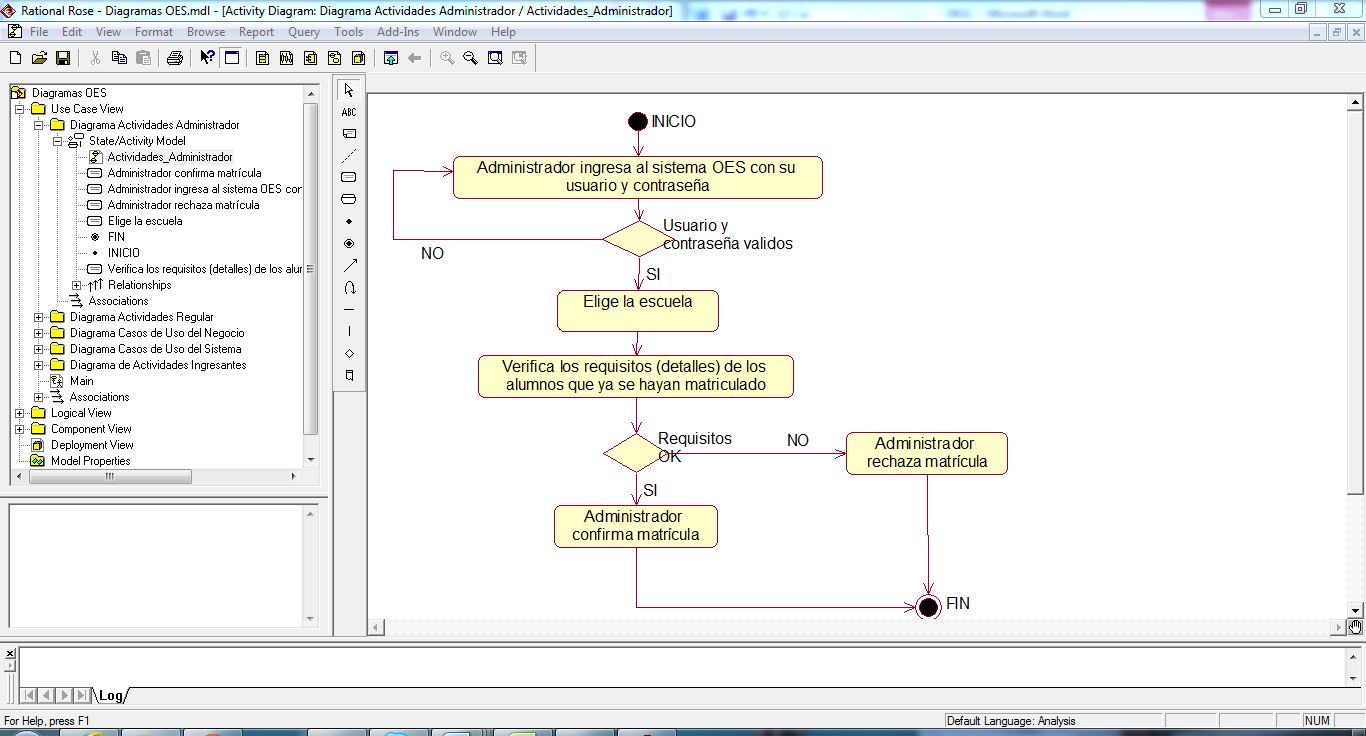
Es un lenguaje para visualizar, especificar, construir modificar y/o ser base de documentación.

* Definición de estereotipos: un estereotipo es una nueva clase de elemento de modelado que debe basarse en ciertas clases ya existentes en el metamodelo y constituye un mecanismo de extensión del modelo.
* Responsabilidades.
* Mecanismos de extensibilidad: estereotipos, valores etiquetados y restricciones.
* Tareas y procesos.
* Distribución y concurrencia (para modelar por ejemplo ActiveX/ DCOM y CORBA).
* Patrones/Colaboraciones.
* Diagramas de actividad (para reingeniería de proceso de negocios)
* Clara separación de tipo, clase e instancia.
* Refinamiento (para manejar relaciones entre niveles de abstracción).
* Interfaces y componentes.

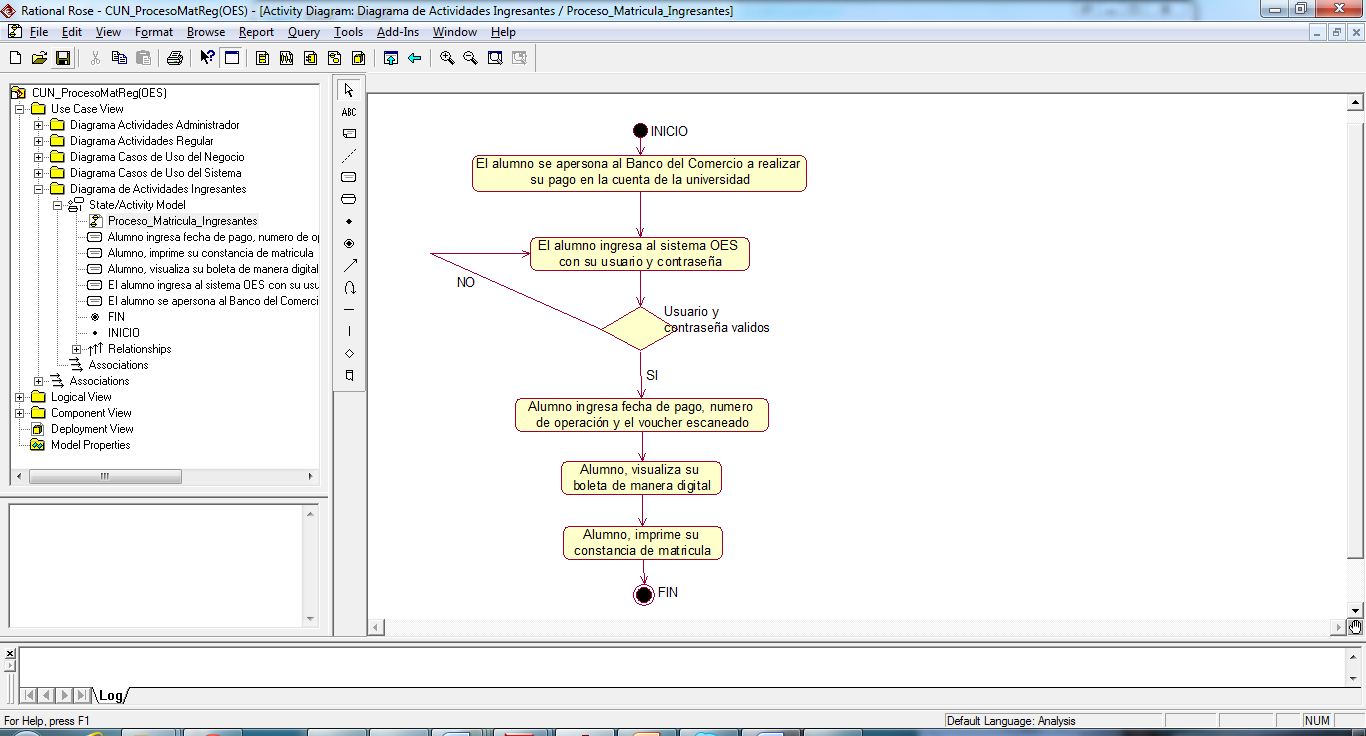
1. **Casos de Uso de Negocio**
2. **Casos de Uso de Sistema**



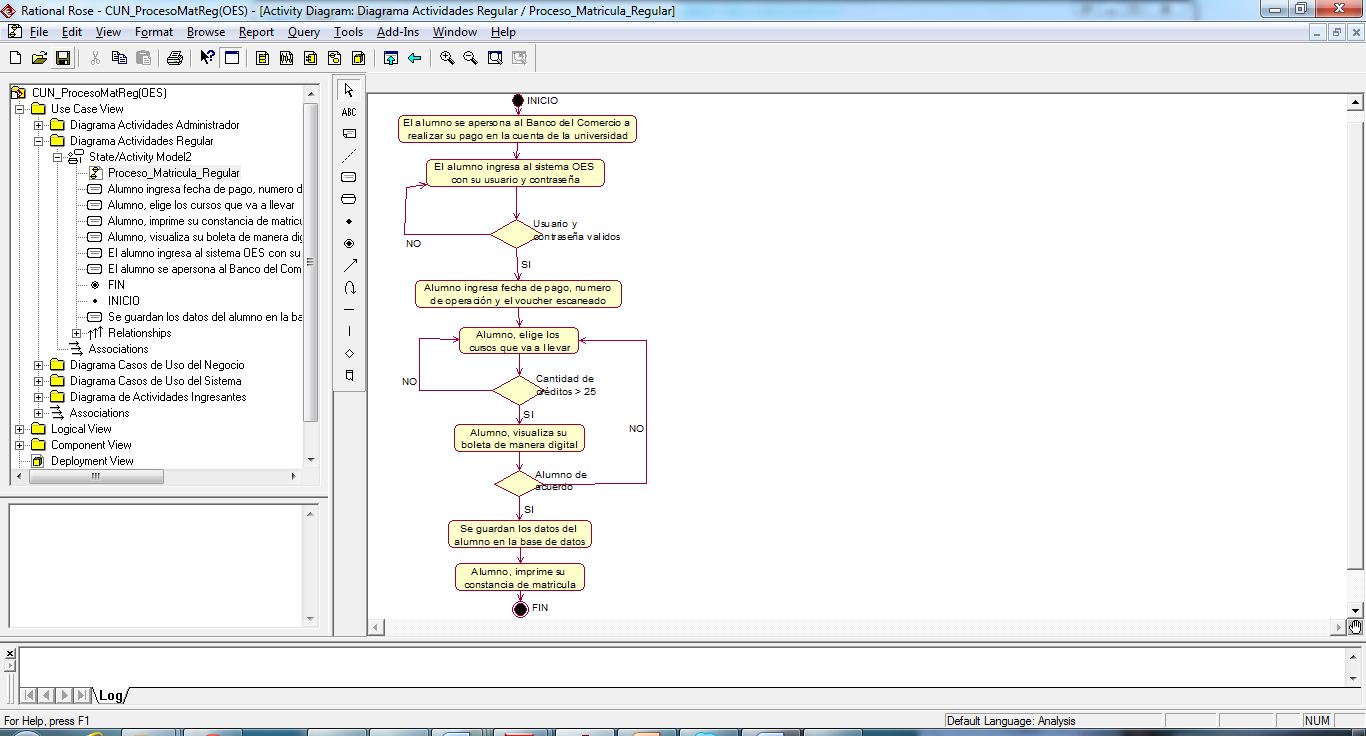
1. **Diagrama de Actividades (Administrador)**



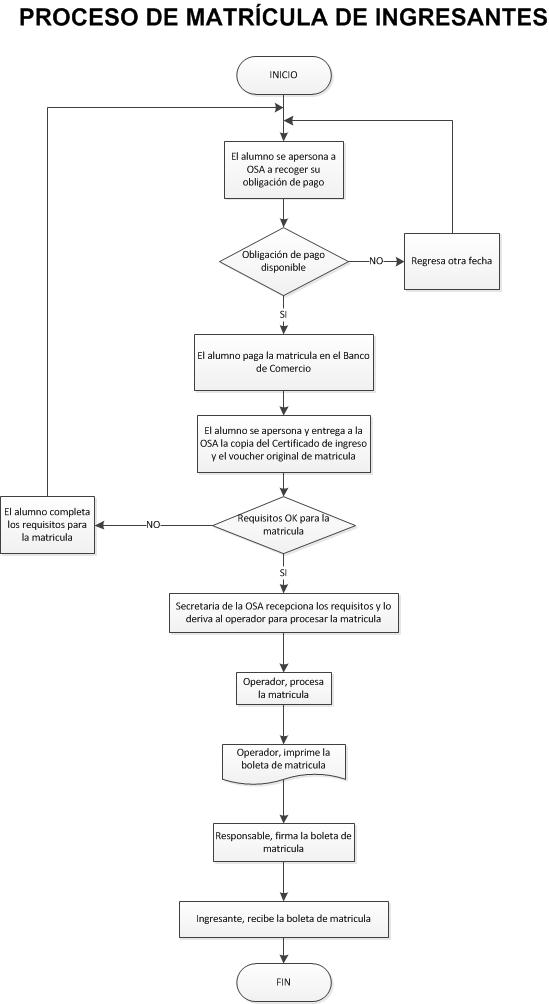
1. **Diagrama de Actividades (Proceso Ingresantes)**

****

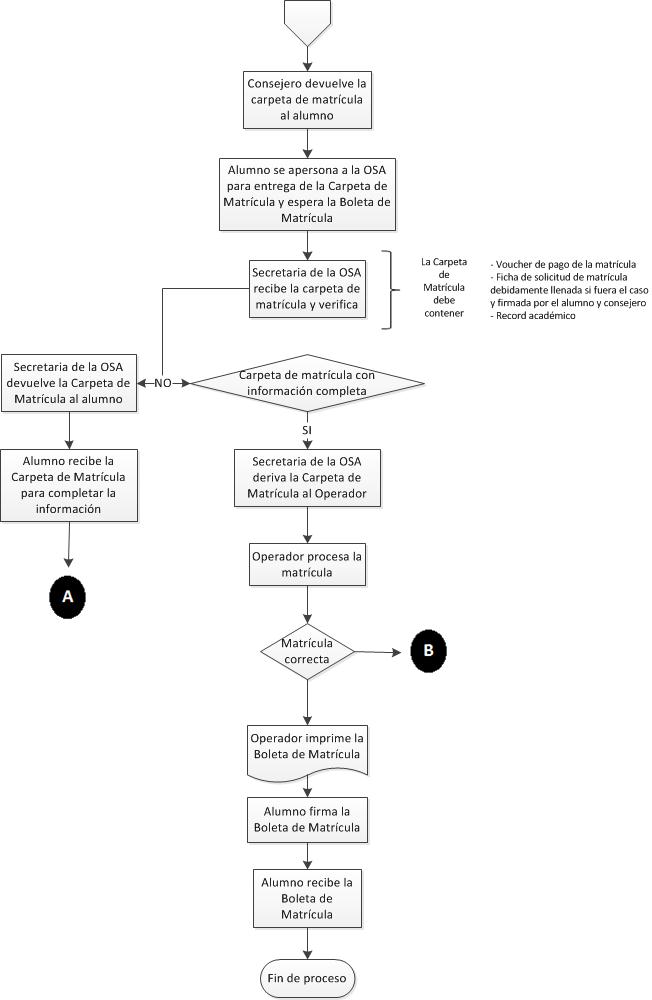
1. **Diagrama de Actividades (Proceso Regular)**

****

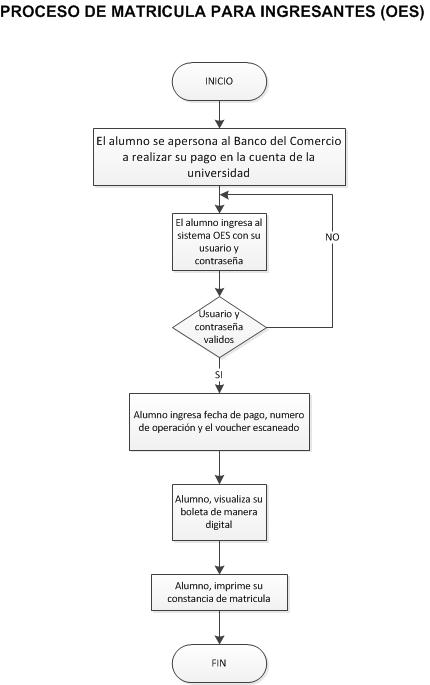
1. **DIFERENCIAS ENTRE EL SISTEMA DE MATRÍCULA ACTUAL Y EL OES**
2. **Diagramas de flujo (Proceso Actual)**

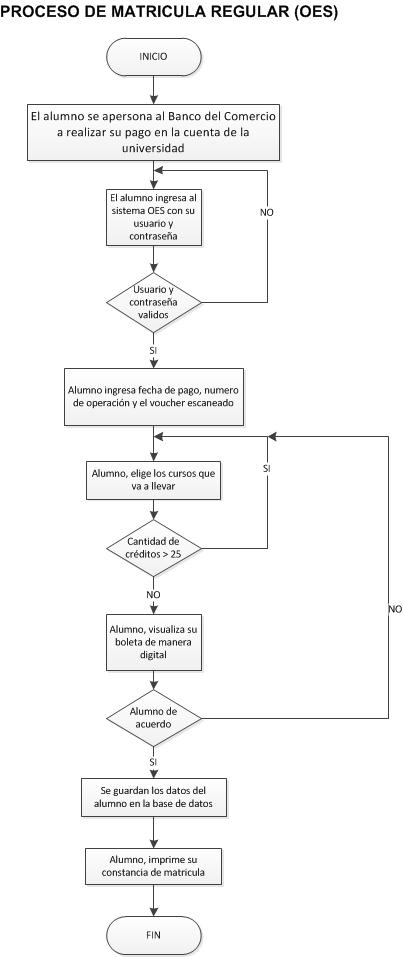
****

****

****

1. **Diagramas de flujo (Proceso mediante OES)**

****

****

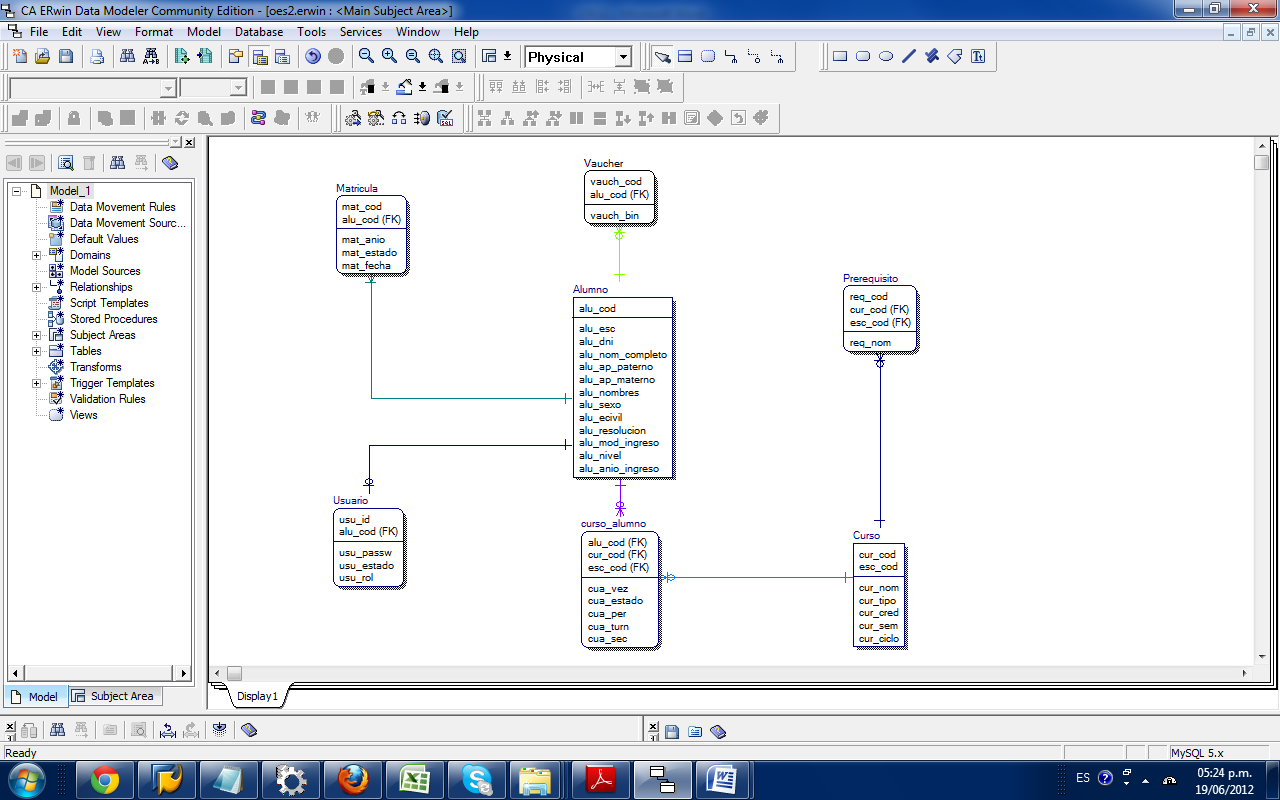
1. **Tabla de diferencias entre el proceso de matrícula actual y OES**

|  |  |
| --- | --- |
| SISTEMA DE MATRÍCULA ACTUAL | SISTEMA DE MATRÍCULA ONLINE |
| En el proceso actual, el estudiante tiene que estar presente en la facultad para recoger la carpeta de matrícula y posteriormente proceder a matricularse formando en ciertos casos, largas colas en la OSA. | Gracias al OES, el alumno se apersonará a nuestra facultad para recoger su carpeta de matrícula pudiendo realizar el respectivo proceso desde la comodidad de su hogar mediante una PC y acceso a internet o en la misma facultad, mediante las computadoras de los laboratorios de la FIEI. |
| El proceso actual nos conlleva a mucha pérdida de tiempo en un proceso repetitivo para los administrativos encargados de la matrícula y en el caso de los alumnos, esperar pacientemente a ser atendidos. | El OES, implementado con interfaces amigables para los usuarios, nos permitirá sistematizar este proceso, haciéndolo de esta manera más simple para los administrativos y para el alumno sin restricciones de horario, ya que estará disponible 24/7. |
| El alumno llena la ficha de solicitud de matrícula, verificando: record académico y plan curricular. | El alumno visualiza los cursos que debe y puede llevar con la opción de desmarcar aquellos en los que no desee matricularse si fuera el caso, dependiendo de sus prerrequisitos. |
| El alumno debe apersonarse con su carpeta a un consejero de matrícula, quien verifica la conformidad de la misma. | Gracias al OES, la labor del consejero de matrícula ya no será necesaria y verificar la conformidad de la matrícula será responsabilidad única del alumno. |
| El alumno se apersona a la OSA para entregar carpeta de matrícula. | El alumno será capaz de verificar la matrícula gracias a una vista previa de su boleta con los cursos a los cuales se inscribió. |
| El operador imprime y le entrega la boleta de matrícula al alumno. | El alumno, luego de verificar la conformidad de su matrícula, imprime su constancia, la cual deberá ser cambiada por la original boleta de matrícula posteriormente a este proceso. |

1. **DISEÑO DE LA BASE DE DATOS**

La base de datos será desarrollada en Mysql 5.5 ya que es un motor de gestión de base de datos gratuito y la facultad no pagaría ningún tipo de licenciamiento.

El diagrama entidad-relación se muestra a continuación:

**Diagrama. a –** Diagrama Entidad-Relación para el sistema propuesto

El diccionario de datos se define a continuación:

|  |  |  |  |
| --- | --- | --- | --- |
| Matricula | | | |
| CAMPO | DESCRIPCIÓN | TIPO | TAMAÑO |
| alu\_cod | Indica el código del alumno. | CHAR | 10 |
| mat\_anio | Indica el año en el que se está matriculándose el alumno. | CHAR | 4 |
| mat\_estado | Indica el estado en el que se encuentra la matricula:  No matriculado / En proceso / Rechazado / Matriculado | VARCHAR | 45 |
| mat\_fecha | Indica la fecha en la que el estudiante está realizando su matrícula online. | TIME STAMP | - |

|  |  |  |  |
| --- | --- | --- | --- |
| Usuario | | | |
| CAMPO | DESCRIPCIÓN | TIPO | TAMAÑO |
| usu\_id | Indica el código de usuario. | VARCHAR | 10 |
| usu\_passw | Indica la contraseña del usuario. | VARCHAR | 128 |
| usu\_estado | Indica el estado en el que se encuentra el usuario. (\*) | VARCHAR | 45 |
| usu\_rol | Indica el rol del usuario. (\*\*) | VARCHAR | 5 |

(\*) E: Estudiante / G: Egresado

(\*\*) ALUM: Alumno / ADMIN: Administrador

|  |  |  |  |
| --- | --- | --- | --- |
| Alumno | | | |
| CAMPO | DESCRIPCIÓN | TIPO | TAMAÑO |
| alu\_esc | Indica el código de escuela del alumno. | CHAR | 2 |
| alu\_cod | Indica el código del alumno. | CHAR | 10 |
| alu\_dni | Indica el nro de dni del alumno. | CHAR | 8 |
| alu\_nom\_completo | Indica los nombres y apellidos del alumno. | VARCHAR | 200 |
| alu\_ap\_paterno | Indica el apellido paterno del alumno. | VARCHAR | 45 |
| alu\_ap\_materno | Indica el apellido materno del alumno. | VARCHAR | 45 |
| alu\_nombres | Indica los nombres y apellidos del alumno. | VARCHAR | 100 |
| alu\_sexo | Indica el sexo del alumno. (\*) | CHAR | 1 |
| alu\_ecivil | Indica el estado civil del alumno. (\*\*) | VARCHAR | 45 |
| alu\_resolucion | Indica la resolución con la que ingreso el alumno. | VARCHAR | 60 |
| alu\_mod\_ingreso | Indica la modalidad de ingreso del alumno. | VARCHAR | 60 |
| alu\_nivel | Indica el año académico en el que se encuentra cursando el alumno. (\*\*\*) | CHAR | 1 |
| alu\_anio\_ingreso | Indica el año de ingreso del alumno | CHAR | 4 |

(\*) M: Masculino / F: Femenino.

(\*\*) S: Soltero / C: Casado / V: Viudo / D: Divorciado.

(\*\*\*) 1: 1er año / 2: 2do año / 3: 3er año / 4: 4to año / 5: 5to año

|  |  |  |  |
| --- | --- | --- | --- |
| curso | | | |
| CAMPO | DESCRIPCIÓN | TIPO | TAMAÑO |
| cur\_cod | Indica el código del curso. | CHAR | 6 |
| esc\_cod | Indica el código de la escuela | CHAR | 2 |
| cur\_nom | Indica el nombre del curso. | VARCHAR | 45 |
| cur\_tipo | Indica el tipo de curso. (\*) | CHAR | 1 |
| cur\_cred | Indica la cantidad de créditos que tiene el curso. | INT | - |
| cur\_sem | Indica el semestre a que pertenece el curso. (\*\*) | VARCHAR | 2 |
| cur\_ciclo | Indica el ciclo al cual pertenece el curso. | VARCHAR | 4 |

(\*) R: Regular / E: Electivo

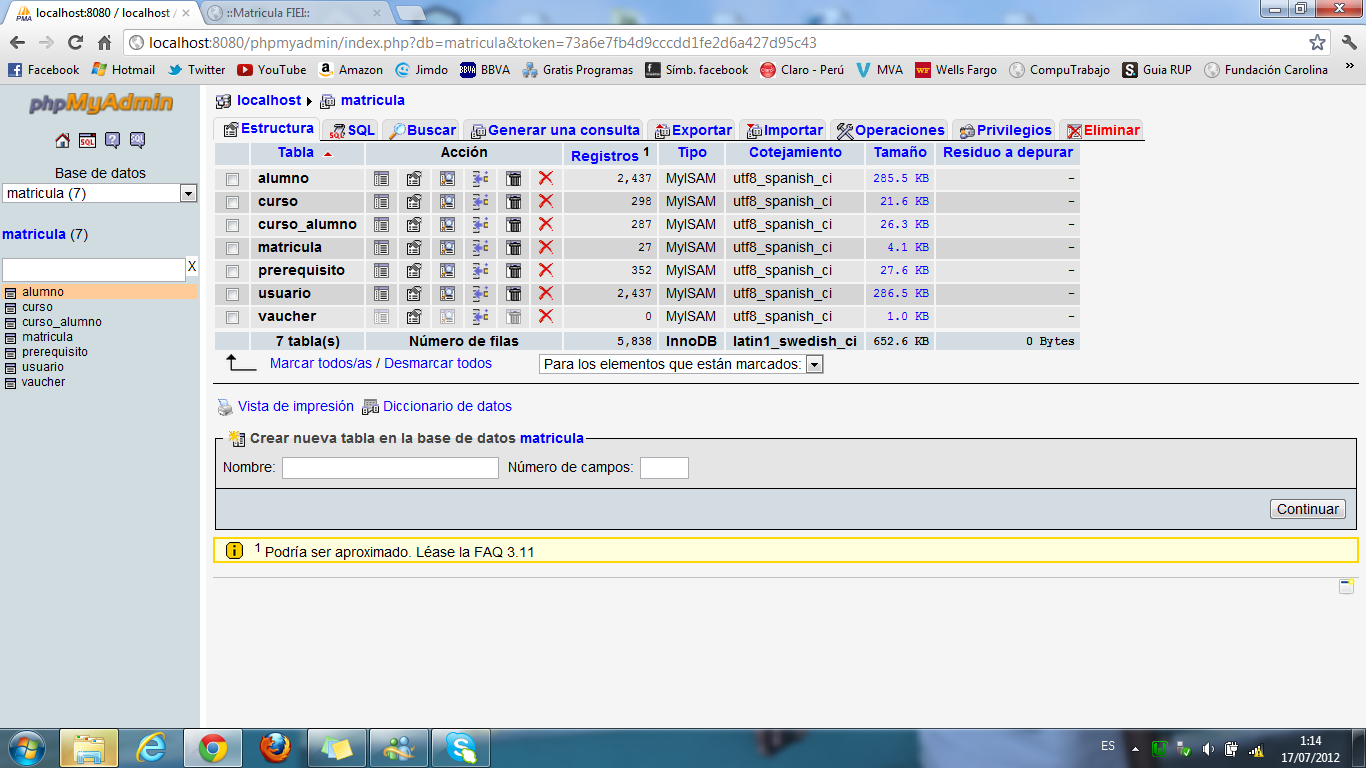
(\*\*) I: 1er semestre / II: 2do semestre

|  |  |  |  |
| --- | --- | --- | --- |
| curso-alumno | | | |
| CAMPO | DESCRIPCIÓN | TIPO | TAMAÑO |
| cur\_cod | Indica el código del curso. | CHAR | 6 |
| alu\_cod | Indica el código del alumno. | CHAR | 45 |
| esc\_cod | Indica el código de la escuela | CHAR | 2 |
| cua\_estado | Indica el estado del curso (\*) | CHAR | 1 |
| cua\_per | Periodo en que se lleva el curso | VARCHAR | 6 |
| cua\_turn | Turno del estudiante | VARCHAR | 1 |
| cua\_sec | Seccion del estudiante | VARCHAR | 1 |

(\*) A: Aprobado / D: Desaprobado

|  |  |  |  |
| --- | --- | --- | --- |
| Vaucher | | | |
| CAMPO | DESCRIPCIÓN | TIPO | TAMAÑO |
| vauch\_cod | Indica el código de la escuela. | CHAR | 2 |
| alu\_cod | Indica el nombre de la escuela | CHAR | 10 |
| Vauch\_bin | contenido | BLOB | Large Binary |

|  |  |  |  |
| --- | --- | --- | --- |
| Prerrequisito | | | |
| CAMPO | DESCRIPCIÓN | TIPO | TAMAÑO |
| req\_cod | Indica el código del requisito. | CHAR | 2 |
| req\_cod | Indica el código del requisito. | CHAR | 2 |
| req\_nom | Indica el nombre del requisito. | VARCHAR | 60 |



**SCRIPT generado por el phpMyadmin:**

-- phpMyAdmin SQL Dump

-- version 3.3.9

-- http://www.phpmyadmin.net

--

-- Servidor: localhost

-- Tiempo de generaciÃ³n: 28-05-2012 a las 06:04:04

-- VersiÃ³n del servidor: 5.5.8

-- VersiÃ³n de PHP: 5.3.5

SET SQL\_MODE="NO\_AUTO\_VALUE\_ON\_ZERO";

/\*!40101 SET @OLD\_CHARACTER\_SET\_CLIENT=@@CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET @OLD\_CHARACTER\_SET\_RESULTS=@@CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET @OLD\_COLLATION\_CONNECTION=@@COLLATION\_CONNECTION \*/;

/\*!40101 SET NAMES utf8 \*/;

--

-- Base de datos: `matricula`

--

-- --------------------------------------------------------

--

-- Estructura de tabla para la tabla `alumno`

--

CREATE TABLE IF NOT EXISTS `alumno` (

`alu\_cod` varchar(10) COLLATE utf8\_spanish\_ci NOT NULL,

`alu\_esc` varchar(2) COLLATE utf8\_spanish\_ci DEFAULT NULL,

`alu\_dni` varchar(45) COLLATE utf8\_spanish\_ci DEFAULT NULL,

`alu\_nom\_completo` varchar(200) COLLATE utf8\_spanish\_ci DEFAULT NULL,

`alu\_ap\_paterno` varchar(45) COLLATE utf8\_spanish\_ci DEFAULT NULL,

`alu\_ap\_materno` varchar(45) COLLATE utf8\_spanish\_ci DEFAULT NULL,

`alu\_nombres` varchar(100) COLLATE utf8\_spanish\_ci DEFAULT NULL,

`alu\_sexo` varchar(1) COLLATE utf8\_spanish\_ci DEFAULT NULL,

`alu\_ecivil` varchar(1) COLLATE utf8\_spanish\_ci DEFAULT NULL,

`alu\_resolucion` varchar(40) COLLATE utf8\_spanish\_ci DEFAULT NULL,

`alu\_mod\_ingreso` varchar(60) COLLATE utf8\_spanish\_ci DEFAULT NULL,

`alu\_nivel` varchar(1) COLLATE utf8\_spanish\_ci DEFAULT NULL,

`alu\_anio\_ingreso` varchar(4) COLLATE utf8\_spanish\_ci DEFAULT NULL,

PRIMARY KEY (`alu\_cod`)

) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8\_spanish\_ci;

-- --------------------------------------------------------

--

-- Estructura de tabla para la tabla `curso`

--

CREATE TABLE IF NOT EXISTS `curso` (

`cur\_cod` varchar(6) COLLATE utf8\_spanish\_ci NOT NULL,

`cur\_nom` varchar(60) COLLATE utf8\_spanish\_ci DEFAULT NULL,

`cur\_tipo` varchar(1) COLLATE utf8\_spanish\_ci DEFAULT NULL,

`cur\_cred` int(11) DEFAULT NULL,

`cur\_sem` varchar(2) COLLATE utf8\_spanish\_ci DEFAULT NULL,

`cur\_ciclo` varchar(4) COLLATE utf8\_spanish\_ci DEFAULT NULL,

`esc\_cod` varchar(2) COLLATE utf8\_spanish\_ci NOT NULL,

PRIMARY KEY (`cur\_cod`,`esc\_cod`)

) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8\_spanish\_ci;

-- --------------------------------------------------------

--

-- Estructura de tabla para la tabla `curso\_alumno`

--

CREATE TABLE IF NOT EXISTS `curso\_alumno` (

`alu\_cod` varchar(10) COLLATE utf8\_spanish\_ci NOT NULL,

`cur\_cod` varchar(6) COLLATE utf8\_spanish\_ci NOT NULL,

`cua\_estado` varchar(1) COLLATE utf8\_spanish\_ci DEFAULT NULL,

`esc\_cod` varchar(2) COLLATE utf8\_spanish\_ci NOT NULL,

`cua\_per` varchar(6) COLLATE utf8\_spanish\_ci DEFAULT NULL,

`cua\_turn` varchar(1) COLLATE utf8\_spanish\_ci DEFAULT NULL,

`cua\_sec` varchar(1) COLLATE utf8\_spanish\_ci DEFAULT NULL,

PRIMARY KEY (`alu\_cod`,`cur\_cod`,`esc\_cod`),

KEY `R\_3` (`cur\_cod`,`esc\_cod`)

) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8\_spanish\_ci;

-- --------------------------------------------------------

--

-- Estructura de tabla para la tabla `matricula`

--

CREATE TABLE IF NOT EXISTS `matricula` (

`mat\_cod` varchar(10) COLLATE utf8\_spanish\_ci NOT NULL,

`mat\_anio` varchar(4) COLLATE utf8\_spanish\_ci DEFAULT NULL,

`mat\_estado` varchar(45) COLLATE utf8\_spanish\_ci DEFAULT NULL,

`mat\_fecha` timestamp NULL DEFAULT NULL,

`alu\_cod` varchar(10) COLLATE utf8\_spanish\_ci NOT NULL,

PRIMARY KEY (`mat\_cod`,`alu\_cod`),

KEY `R\_4` (`alu\_cod`)

) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8\_spanish\_ci;

-- --------------------------------------------------------

--

-- Estructura de tabla para la tabla `prerequisito`

--

CREATE TABLE IF NOT EXISTS `prerequisito` (

`req\_cod` varchar(6) COLLATE utf8\_spanish\_ci NOT NULL,

`req\_nom` varchar(60) COLLATE utf8\_spanish\_ci DEFAULT NULL,

`cur\_cod` varchar(6) COLLATE utf8\_spanish\_ci NOT NULL,

`esc\_cod` varchar(2) COLLATE utf8\_spanish\_ci NOT NULL,

PRIMARY KEY (`req\_cod`,`cur\_cod`,`esc\_cod`),

KEY `R\_8` (`cur\_cod`,`esc\_cod`)

) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8\_spanish\_ci;

-- --------------------------------------------------------

--

-- Estructura de tabla para la tabla `usuario`

--

CREATE TABLE IF NOT EXISTS `usuario` (

`usu\_id` varchar(10) COLLATE utf8\_spanish\_ci NOT NULL,

`usu\_passw` varchar(128) COLLATE utf8\_spanish\_ci DEFAULT NULL,

`usu\_estado` varchar(1) COLLATE utf8\_spanish\_ci DEFAULT NULL,

`usu\_rol` varchar(5) COLLATE utf8\_spanish\_ci DEFAULT NULL,

`alu\_cod` varchar(10) COLLATE utf8\_spanish\_ci NOT NULL,

PRIMARY KEY (`usu\_id`,`alu\_cod`),

KEY `R\_11` (`alu\_cod`)

) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8\_spanish\_ci;

-- --------------------------------------------------------

--

-- Estructura de tabla para la tabla `vaucher`

--

CREATE TABLE IF NOT EXISTS `vaucher` (

`vauch\_cod` varchar(30) COLLATE utf8\_spanish\_ci NOT NULL,

`vauch\_date` date DEFAULT NULL,

`alu\_cod` varchar(10) COLLATE utf8\_spanish\_ci NOT NULL,

PRIMARY KEY (`vauch\_cod`,`alu\_cod`),

KEY `R\_10` (`alu\_cod`)

) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8\_spanish\_ci;

1. **SOFTWARE UTILIZADOS EN EL OES**
2. **GITHUB**

**Git** es uno de los [sistemas de control de versiones](http://www.genbetadev.com/sistemas-de-control-de-versiones/kit-basico-de-herramientas-para-desarrollar-en-equipo-i-control-de-versiones) más populares entre los desarrolladores. Y parte culpa de su popularidad la tiene **GitHub**, un excelente servicio de alojamiento de repositorios de software con este sistema, que lejos de quedarse en esta funcionalidad, ofrece hoy en día un conjunto de características muy útiles para el trabajo en equipo.

No en vano, es el servicio **elegido por proyectos de software libre** como jQuery, reddit, Sparkle, curl, Ruby on Rails, node.js, ClickToFlash, Erlang/OTP, CakePHP, Redis, y otros muchos. Además, algunas de las grandes empresas de Internet, como [Facebook](https://github.com/facebook), alojan ahí sus desarrollos públicos, tales como el SDK, librerías, ejemplos, etc.

**Ventajas**

Aunque Git es un sistema **descentralizado** (no necesita de un servidor donde subir los cambios, sino que los desarrolladores pueden enviarse los mismos los unos a los otros desde sus equipos), es muy habitual encontrarnos con que un equipo de trabajo funciona mejor disponiendo de uno, ya que así evitamos tener que descargar los cambios de cada ordenador perteneciente a un miembro del equipo de manera individual, debiendo éste permanecer encendido para poder efectuar esas operaciones. De modo que una de las ventajas evidentes de GitHub, y que fue el servicio casi único con el que nació, es **alojar el repositorio de código**en el que trabaja todo el equipo.

Al alojar el código, podemos decidir si éste será público o su visibilidad estará limitada a unos usuarios determinados, que son los únicos que tendrán permiso para acceder al mismo y subir cambios.

Pero además de servir de alojamiento para el código, GitHub posee un estupendo **visor de código** mediante el cual, a través del navegador, podremos consultar en un instante el contenido de un determinado fichero, con su **resaltado de sintaxis**correspondiente para el lenguaje en el que esté escrito. Este navegador es realmente rápido, y gracias a él podremos hacer pequeñas consultas o copiar porciones de código que nos puedan hacer falta en un momento determinado sin necesidad de bajarse todo el repositorio. Por supuesto, se puede navegar por cualquiera de las versiones del mismo, de modo que podemos ver el contenido de un fichero antes de que se dieran varias actualizaciones de cambios. Seleccionar un determinado commit, tag o rama es cuestión de pocos clicks.

**Características de GitHub útiles para el trabajo en equipo**

Pero hoy en día GitHub es mucho más que un servicio de alojamiento de código. Además de éste, se ofrecen varias pequeñas herramientas en línea muy útiles para el trabajo en equipo. Entre ellas, caben destacar:

* Un wiki que funciona con [gollum](https://github.com/github/gollum), el cual opera con Git para el mantenimiento de las distintas versiones de las páginas.
* Un [sistema de seguimiento de problemas](https://github.com/rails/rails/issues), que al estilo del clásico sistema de tickets, permiten a los miembros de tu equipo (o a cualquier usuario de GitHub si tu repositorio es público) abrir un ticket detallando un problema que tenga con tu software o una sugerencia que desee hacer al mismo.
* Una [herramienta de revisión de código](https://github.com/rails/rails/commit/0495736339df48a625fd58f4304b7b497b02d859#comments), donde se pueden añadir anotaciones en cualquier punto de un fichero (ej: “Esto es mejor que lo extraigamos a una nueva clase”), y debatir sobre determinados cambios realizados en un commit específico.
* Un [visor de ramas](https://github.com/rails/rails/branches) donde se pueden comparar los progresos realizados en las distintas ramas de nuestro repositorio.



**Costo**

GitHub es totalmente gratuito para alojar código open source, siendo estos repositorios de visibilidad pública forzosamente. Sin embargo, su [plan de precios](https://github.com/plans) para los repositorios privados y con colaboradores externos es más que razonable, abarcando desde 5 repositorios privados y hasta un colaborador por 7 dólares al mes, hasta 125 repositorios privados y equipos de personas ilimitados por 200 dólares al mes.

1. **WAMPSERVER**

**WampServer** es la continuación del anterior proyecto que se llamaba **Wamp**. Este software es gratuito y es utilizado para **crear un** [servidor local](http://bitsbeta.com/xampp-servidor-local-para-blogs).

El programa sirve para **instalar un servidor** (lo mismo que se hace en los sitios en Internet) en el disco duro de nuestro equipo. A partir de ahí hasta es posible funcionar como hospedaje para otros usuarios que quieran poner sus sitios webs en nuestro equipo.

Pero la mayoría usa **WampServer** para hacer funcionar sus blogs en el PC y hacer pruebas de funcionamiento antes de subirl el blog definitivamente al **servidor u hospedaje web**.

**WampServer** es una muy buena opción para todos los niveles de usuarios. Los usuarios principiantes lo van a encontrar muy fácil de usar por su sistema de menú e interfaz gráfica.

Los usuarios avanzados encuentran en **WampServer** un servidor muy potente que integra: Apache, MySQL y PHP.

**Funcionalidades de WampServer**

WampServer tiene  funcionalidades que lo hacen  muy completo y fácil de usar. Con un click izquierdo sobre el icono de WampServer, usted será capaz de:

* Gestionar sus servicios de Apache y MySQL,
* Cambiar de línea / fuera de línea (dar acceso a todos o sólo local)
* Instalar y cambiar de Apache, MySQL y PHP emisiones
* Gestión de la configuración de sus servidores
* Acceder a sus registros
* Acceder a sus archivos de configuración
* Crear alias

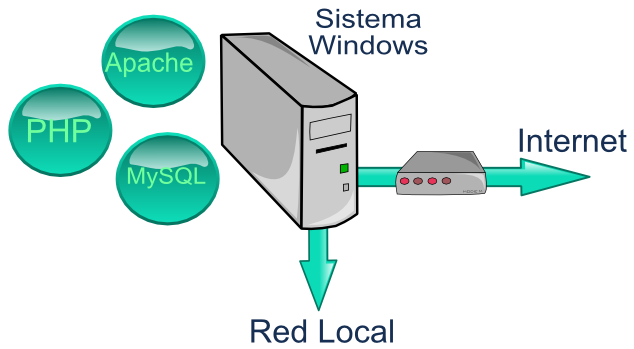
Con un clic derecho:

* WampServer cambiar el idioma del menú
* Acceder a la página principal

**La terna Apache + PHP + MySQL es la base para instalar infinidad de aplicaciones web**

Instalar y configurar un servidor Apache, un servidor MySQL y el lenguaje PHP, así como configurarlo para que interrelacionen entre ellos y el servidor funcione perfectamente, es una tarea compleja que solo pueden acometer informáticos profesionales. Para simplificar la tarea de instalar Apache + PHP + MySQL en Windows y acercar al gran público la posibilidad de disfrutar de estos servicios, existen los llamados paquetes WAMP que instalan y configuran automáticamente dichas aplicaciones para Windows y que proporcionan:

* Servidor Web Apache
* Base de datos MySQL
* Lenguaje de programación PHP
* Accesos para el arranque y la parada de los servicios
* Facilidades para la configuración de los servicios
* Otros servicios



Un Servidor WAMP ofrece Apache + PHP + MySQL en un sistema Windows

1. **NETBEANS**

NetBeans es un proyecto exitoso de código abierto con una gran base de usuarios, una comunidad en constante crecimiento. Sun MicroSystems fundó el proyecto de código abierto NetBeans en junio 2000 y continúa siendo el patrocinador principal de los proyectos.

Al día de hoy hay disponibles dos productos: el NetBeans IDE y NetBeans Platform.

NetBeans IDE es un entorno de desarrollo - una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Está escrito en Java - pero puede servir para cualquier otro lenguaje de programación. Existe además un número importante de módulos para extender el NetBeans IDE. NetBeans IDE es un producto libre y gratuito sin restricciones de uso.

También está disponible NetBeans Platform; una base modular y extensible usada como estructura de integración para crear grandes aplicaciones de escritorio. Empresas independientes asociadas, especializadas en desarrollo de software, proporcionan extensiones adicionales que se integran fácilmente en la plataforma y que pueden también utilizarse para desarrollar sus propias herramientas y soluciones.

Ambos productos son de código abierto y gratuito para uso tanto comercial como no comercial. El código fuente está disponible para su reutilización de acuerdo con la Common Development and Distribution License (CDDL) v1.0 and the GNU General Public License (GPL) v2.

La plataforma NetBeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados módulos. Un módulo es un archivo Java que contiene clases de java escritas para interactuar con las APIs de NetBeans y un archivo especial (manifest file) que lo identifica como módulo. Las aplicaciones construidas a partir de módulos pueden ser extendidas agregándole nuevos módulos. Debido a que los módulos pueden ser desarrollados independientemente, las aplicaciones basadas en la plataforma NetBeans pueden ser extendidas fácilmente por otros desarrolladores de software.

**Netbeans IDE**

El IDE Netbeans es un [entorno de desarrollo integrado](http://es.wikipedia.org/wiki/Entorno_de_desarrollo_integrado) - una herramienta para programadores pensada para escribir, compilar, [depurar](http://es.wikipedia.org/wiki/Depuraci%C3%B3n_de_programas) y ejecutar programas. Está escrito en Java - pero puede servir para cualquier otro lenguaje de programación. Existe además un número importante de módulos para extender el IDE Netbeans. El IDE Netbeans es un producto libre y gratuito sin restricciones de uso.

El Netbeans IDE es un [IDE](http://es.wikipedia.org/wiki/Entorno_de_desarrollo_integrado) de código abierto escrito completamente en Java usando la plataforma Netbeans. El Netbeans IDE soporta el desarrollo de todos los tipos de aplicación Java (J2SE, web, EJB y aplicaciones móviles). Entre sus características se encuentra un sistema de proyectos basado en Ant, control de versiones y refactoring.

Netbeans IDE 6.5, la cual fue publicada el 19 de noviembre de 2008, extiende las características existentes del Java EE (incluyendo Soporte a Persistencia, EJB 3 y JAX-WS). Adicionalmente, el Netbeans Enterprise Pack soporta el desarrollo de Aplicaciones empresariales con Java EE 5, incluyendo herramientas de desarrollo visuales de SOA, herramientas de esquemas XML, orientación a web servicies (for BPEL), y modelado UML. El Netbeans C/C++ Pack soporta proyectos de C/C++, mientras el [PHP](http://es.wikipedia.org/wiki/PHP) Pack, soporta PHP 5.

Modularidad. Todas las funciones del IDE son provistas por módulos. Cada módulo provee una función bien definida, tales como el soporte de Java, edición, o soporte para el sistema de control de versiones. Netbeans contiene todos los módulos necesarios para el desarrollo de aplicaciones Java en una sola descarga, permitiéndole al usuario comenzar a trabajar inmediatamente.

[Sun Studio](http://es.wikipedia.org/wiki/Sun_Studio), [Sun Java Studio Enterprise](http://es.wikipedia.org/w/index.php?title=Sun_Java_Studio_Enterprise&action=edit&redlink=1" \o "Sun Java Studio Enterprise (aún no redactado)), y [Sun Java Studio Creator](http://es.wikipedia.org/w/index.php?title=Sun_Java_Studio_Creator&action=edit&redlink=1" \o "Sun Java Studio Creator (aún no redactado)) de [Sun Microsystems](http://es.wikipedia.org/wiki/Sun_Microsystems" \o "Sun Microsystems) han sido todos basados en el IDE Netbeans.

Desde julio de 2006, NetBeans IDE es licenciado bajo la Common Development and Distribution License ([CDDL](http://es.wikipedia.org/wiki/CDDL)), una licencia basada en la [Mozilla Public License](http://es.wikipedia.org/wiki/Mozilla_Public_License) (MPL).

1. **APACHE**

El [servidor](http://informatica-sinlimites.blogspot.com/2011/01/que-es-un-servidor.html) HTTP Apache es un servidor web HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP/1 y la noción de sitio virtual. Cuando comenzó su desarrollo en 1995 se basó inicialmente en código del popular NCSA HTTPd 1.3, pero más tarde fue reescrito por completo. Su nombre se debe a que Behelendorf quería que tuviese la connotación de algo que es firme y enérgico pero no agresivo, y la tribu Apache fue la última en rendirse al que pronto se convertiría en gobierno de EEUU, y en esos momentos la preocupación de su grupo era que llegasen las empresas y "civilizasen" el paisaje que habían creado los primeros ingenieros de internet. Además Apache consistía solamente en un conjunto de parches a aplicar al servidor de NCSA. Era, en inglés, a patchy server (un servidor "parcheado").

El servidor Apache se desarrolla dentro del proyecto HTTP Server (httpd) de la Apache Software Foundation.

Apache presenta entre otras características altamente configurables, bases de datos de autenticación y negociado de contenido, pero fue criticado por la falta de una interfaz gráfica que ayude en su configuración.

Apache tiene amplia aceptación en la red: desde 1996, Apache, es el servidor HTTP más usado. Alcanzó su máxima cuota de mercado en 2005 siendo el servidor empleado en el 70% de los sitios web en el mundo, sin embargo ha sufrido un descenso en su cuota de mercado en los últimos años.

La mayoría de las vulnerabilidades de la seguridad descubiertas y resueltas tan sólo pueden ser aprovechadas por usuarios locales y no remotamente. Sin embargo, algunas se pueden accionar remotamente en ciertas situaciones, o explotar por los usuarios locales malévolos en las disposiciones de recibimiento compartidas que utilizan PHP como módulo de Apache.

**Uso del apache**

Apache es usado principalmente para enviar páginas web estáticas y dinámicas en la World Wide Web. Muchas aplicaciones web están diseñadas asumiendo como ambiente de implantación a Apache, o que utilizarán características propias de este servidor web.

Apache es el componente de servidor web en la popular plataforma de aplicaciones [LAMP](http://es.wikipedia.org/wiki/LAMP), junto a [MySQL](http://es.wikipedia.org/wiki/MySQL" \o "MySQL) y los lenguajes de programación [PHP](http://es.wikipedia.org/wiki/PHP)/[Perl](http://es.wikipedia.org/wiki/Perl)/[Python](http://es.wikipedia.org/wiki/Python" \o "Python) (y ahora también [Ruby](http://es.wikipedia.org/wiki/Ruby)).

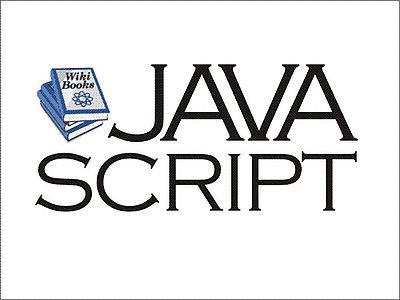
Este servidor web es redistribuido como parte de varios paquetes propietarios de software, incluyendo la base de datos [Oracle](http://es.wikipedia.org/wiki/Oracle) y el IBM [WebSphere](http://es.wikipedia.org/wiki/WebSphere" \o "WebSphere) application server. Mac OS X integra apache como parte de su propio servidor web y como soporte de su servidor de aplicaciones [WebObjects](http://es.wikipedia.org/w/index.php?title=WebObjects&action=edit&redlink=1" \o "WebObjects (aún no redactado)). Es soportado de alguna manera por [Borland](http://es.wikipedia.org/wiki/Borland" \o "Borland) en las herramientas de desarrollo Kylix y Delphi. Apache es incluido con Novell NetWare 6.5, donde es el servidor web por defecto, y en muchas distribuciones Linux.

Apache es usado para muchas otras tareas donde el contenido necesita ser puesto a disposición en una forma segura y confiable. Un ejemplo es al momento de compartir archivos desde una computadora personal hacia Internet. Un usuario que tiene Apache instalado en su escritorio puede colocar arbitrariamente archivos en la raíz de documentos de Apache, desde donde pueden ser compartidos.

Los programadores de aplicaciones web a veces utilizan una versión local de Apache con el fin de previsualizar y probar código mientras éste es desarrollado.

Microsoft [Internet Information Services](http://es.wikipedia.org/wiki/Internet_Information_Services) (IIS) es el principal competidor de Apache, así como Sun Java System Web Server de [Sun Microsystems](http://es.wikipedia.org/wiki/Sun_Microsystems" \o "Sun Microsystems) y un anfitrión de otras aplicaciones como Zeus Web Server. Algunos de los más grandes sitios web del mundo están ejecutándose sobre Apache. La capa frontal (front end) del motor de búsqueda [Google](http://es.wikipedia.org/wiki/Google) está basada en una versión modificada de Apache, denominada Google Web Server (GWS). Muchos proyectos de [Wikimedia](http://es.wikipedia.org/wiki/Wikimedia" \o "Wikimedia) también se ejecutan sobre servidores web Apache.

1. **JAVASCRIPT**

JavaScript es un [lenguaje de programación interpretado](http://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n_interpretado). Se define como [orientado a objetos](http://es.wikipedia.org/wiki/Programaci%C3%B3n_orientada_a_objetos), [basado en prototipos](http://es.wikipedia.org/wiki/Programaci%C3%B3n_basada_en_prototipos), [imperativo](http://es.wikipedia.org/wiki/Programaci%C3%B3n_imperativa), débilmente tipado y dinámico.

Se utiliza principalmente en su forma del [lado del cliente](http://es.wikipedia.org/wiki/Lado_del_cliente), implementado como parte de un [navegador web](http://es.wikipedia.org/wiki/Navegador_web) permitiendo mejoras en la [interfaz de usuario](http://es.wikipedia.org/wiki/Interfaz_de_usuario) y [páginas web](http://es.wikipedia.org/wiki/P%C3%A1gina_web) dinámicas, en bases de datos locales al navegador, aunque existe una forma de JavaScript del [lado del servidor](http://es.wikipedia.org/w/index.php?title=Lado_del_servidor&action=edit&redlink=1) (Server-side JavaScript o SSJS). Su uso en [aplicaciones](http://es.wikipedia.org/wiki/Aplicaci%C3%B3n_inform%C3%A1tica) externas a la [web](http://es.wikipedia.org/wiki/World_Wide_Web), por ejemplo en documentos [PDF](http://es.wikipedia.org/wiki/PDF), aplicaciones de escritorio (mayoritariamente [widgets](http://es.wikipedia.org/wiki/Widget" \o "Widget)) es también significativo.

JavaScript se diseñó con una sintaxis similar al [C](http://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n_C), aunque adopta nombres y convenciones del [lenguaje de programación Java](http://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n_Java). Sin embargo [Java](http://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n_Java) y JavaScript no están relacionados y tienen semánticas y propósitos diferentes.

Todos los navegadores modernos interpretan el código JavaScript integrado en las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del [Document Object Model](http://es.wikipedia.org/wiki/Document_Object_Model" \o "Document Object Model) (DOM).

Tradicionalmente se venía utilizando en páginas web [HTML](http://es.wikipedia.org/wiki/HTML) para realizar operaciones y únicamente en el marco de la [aplicación cliente](http://es.wikipedia.org/wiki/Cliente_(inform%C3%A1tica)), sin acceso a funciones del [servidor](http://es.wikipedia.org/wiki/Servidor). JavaScript se interpreta en el [agente de usuario](http://es.wikipedia.org/wiki/Agentes_de_usuario), al mismo tiempo que las sentencias van descargándose junto con el código [HTML](http://es.wikipedia.org/wiki/HTML).

1. **JQUERY**

jQuery es una [biblioteca](http://es.wikipedia.org/wiki/Biblioteca_(inform%C3%A1tica)) de [JavaScript](http://es.wikipedia.org/wiki/JavaScript), que permite simplificar la manera de interactuar con los documentos [HTML](http://es.wikipedia.org/wiki/HTML), manipular el árbol [DOM](http://es.wikipedia.org/wiki/Document_Object_Model), manejar eventos, desarrollar animaciones y agregar interacción con la técnica [AJAX](http://es.wikipedia.org/wiki/AJAX) a páginas web.

jQuery es [software libre y de código abierto](http://es.wikipedia.org/wiki/Software_libre_y_de_c%C3%B3digo_abierto), posee un doble licenciamiento bajo la [Licencia MIT](http://es.wikipedia.org/wiki/MIT_License) y la [Licencia Pública General de GNU](http://es.wikipedia.org/wiki/GNU_General_Public_License) v2, permitiendo su uso en proyectos [libres](http://es.wikipedia.org/wiki/Software_libre) y [privativos](http://es.wikipedia.org/wiki/Software_propietario). jQuery, al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio.

Las empresas [Microsoft](http://es.wikipedia.org/wiki/Microsoft) y [Nokia](http://es.wikipedia.org/wiki/Nokia) anunciaron que incluirán la biblioteca en sus plataformas. Microsoft la añadirá en su [IDE](http://es.wikipedia.org/wiki/Entorno_de_desarrollo_integrado) [Visual Studio](http://es.wikipedia.org/wiki/Microsoft_Visual_Studio)[3](http://es.wikipedia.org/wiki/JQuery#cite_note-2) y la usará junto con los frameworks [ASP.NET AJAX](http://es.wikipedia.org/wiki/ASP.NET_AJAX) y [ASP.NET MVC](http://es.wikipedia.org/w/index.php?title=ASP.NET_MVC&action=edit&redlink=1), mientras que Nokia los integrará con su plataforma [Web Run-Time](http://es.wikipedia.org/w/index.php?title=Web_Run-Time&action=edit&redlink=1).

1. **AJAX**

Ajax ([JavaScript](http://es.wikipedia.org/wiki/JavaScript) asíncrono y [XML](http://es.wikipedia.org/wiki/XML)), es una técnica de [desarrollo web](http://es.wikipedia.org/wiki/Desarrollo_web) para crear aplicaciones interactivas. Estas aplicaciones se ejecutan en el [cliente](http://es.wikipedia.org/wiki/Cliente_(inform%C3%A1tica)), es decir, en el [navegador](http://es.wikipedia.org/wiki/Navegador_web) de los usuarios mientras se mantiene la comunicación [asíncrona](http://es.wikipedia.org/wiki/As%C3%ADncrono) con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad y [usabilidad](http://es.wikipedia.org/wiki/Usabilidad) en las aplicaciones.

Ajax es una tecnología asíncrona, en el sentido de que los datos adicionales se solicitan al servidor y se cargan en segundo plano sin interferir con la visualización ni el comportamiento de la página. [JavaScript](http://es.wikipedia.org/wiki/JavaScript) es el lenguaje interpretado en el que normalmente se efectúan las funciones de llamada de Ajax mientras que el acceso a los datos se realiza mediante [XMLHttpRequest](http://es.wikipedia.org/wiki/XMLHttpRequest" \o "XMLHttpRequest), objeto disponible en los navegadores actuales. En cualquier caso, no es necesario que el contenido asíncrono esté formateado en [XML](http://es.wikipedia.org/wiki/XML).

Ajax es una técnica válida para múltiples plataformas y utilizable en muchos sistemas operativos y navegadores dado que está basado en estándares abiertos como JavaScript y [Document Object Model](http://es.wikipedia.org/wiki/Document_Object_Model" \o "Document Object Model)(DOM).

1. **GIT**

En [el desarrollo de software](http://en.wikipedia.org/wiki/Software_development) , Git es una [revisión de control distribuido](http://en.wikipedia.org/wiki/Distributed_revision_control) y la [gestión del código fuente](http://en.wikipedia.org/wiki/Source_code_management) del sistema con un énfasis en la velocidad. Git fue inicialmente diseñado y desarrollado por [Linus Torvalds](http://en.wikipedia.org/wiki/Linus_Torvalds" \o "Linus Torvalds) para el [kernel de Linux](http://en.wikipedia.org/wiki/Linux_kernel" \o "El kernel de Linux) de desarrollo. Cada Git [directorio de trabajo](http://en.wikipedia.org/wiki/Working_directory) es un hecho y derecho [del repositorio](http://en.wikipedia.org/wiki/Repository_(version_control)) con la historia completa y todas las capacidades de seguimiento de la revisión, no depende de acceso a la red o un servidor central. Git es [software libre](http://en.wikipedia.org/wiki/Free_software) distribuido bajo los términos de la [GNU General Public License](http://en.wikipedia.org/wiki/GNU_General_Public_License) versión 2.

**Diseño**

El diseño de Git se inspiró en [BitKeeper](http://en.wikipedia.org/wiki/BitKeeper" \o "BitKeeper) y [monótona](http://en.wikipedia.org/wiki/Monotone_(software)) . Git fue diseñado originalmente como un motor de sistema de bajo nivel de control de versiones en la parte superior de los cuales los demás podrían escribir los extremos delanteros, como [Cogito](http://en.wikipedia.org/wiki/Cogito_(software)) o [StGIT](http://en.wikipedia.org/w/index.php?title=StGIT&action=edit&redlink=1" \o "StGIT (página no existe)) . Sin embargo, el central del proyecto Git se ha convertido en un sistema de control de la revisión completa que se puede utilizar directamente.

**Terminología y estructura**

Un repositorio Git - datos y metadatos - está completamente contenida dentro de su directorio, por lo normal a nivel del sistema de copia (o renombrar o eliminar) de un repositorio Git es toda una operación segura. La copia resultante es a la vez independiente y consciente de la original.

**Datos**

* árbol de trabajo: Los archivos y directorios - las cosas que normalmente se consideran "bajo control de código fuente, pero el check out" de que los cambios del usuario en un editor o de otro tipo, viven en el directorio del repositorio superior, pero fuera de su git directorio /..

Durante una sesión de trabajo normal, se puede optar por solicitar a Git que tenga en cuenta para más adelante todas o algunas de las diferencias entre cómo los archivos se encuentran ahora y cómo depósito permanente de Git se acuerda de ellos. El lugar que se acuerda de ellos se llama "el área de ensayo" (o índice) y recordando que se denomina "estadificación". También se puede eliminar o excluir a los cambios seleccionados de estadificación si se adapta. La estadificación diffs es fácil, y es similar en concepto a la creación de un fichero de parche. La estadificación es el precursor de la perpetración de esos cambios como un registro permanente en el repositorio Git.

* . gitignore: Se trata de un archivo de texto sin formato de configuración que contiene los patrones de nombre de archivo que git debe considerar invisibles a los efectos del control de código fuente - por ejemplo ., \* o \*, clase, bin /.
* está fuera del directorio de git con el "normal" los archivos, con el fin de mantenerla bajo control de código fuente:. todos en un proyecto suele necesitar las mismas exclusiones.
* puede referirse a sí mismo, para excusarse de control de código fuente!

**Metadatos**

El "detrás de las escenas de" elementos de control de código fuente vivir dentro del git. / Directorio en la raíz del repositorio.

* Se compromete: lo que representa el check-in versiones de los archivos y directorios viven en .git / objetos
* cada confirmación representa una instantánea del árbol lleno de archivos actuales en el momento que se creó la confirmación.
* cada confirmación se identifican de forma única a Git comandos mediante un identificador de 40 caracteres, pero por lo general es lo suficientemente única para proporcionar sólo los siete primeros caracteres de la misma.
* git se quejará si un identificador abreviado es ambiguo.
* cada confirmación tiene una referencia a su compromiso padre (s).
* En conjunto, todas las confirmaciones formar una estructura de árbol sobre la base de la relación entre padres, donde el concepto de ramas deriva.
* Índice (etapa): una serie de cambios en los archivos están recogiendo (con "git add") para crear en el repositorio como una única confirmación (versión)
* cambios en el archivo agregado de forma explícita en el índice que se denomina "Los cambios en escena".
* cambios en los archivos pero no hicieron (o todavía no) que se añade al índice que se denomina "cambios" no escenificados.
* etapas y los cambios no escénicos conviven sin causar daño (y se puede ver con "git status").
* la creación de una confirmación (con "git commit") copia los cambios por etapas en un compromiso nuevo y luego se borra el índice de
* creando una confirmación no afecta el árbol de trabajo.
* el nuevo compromiso se convierte en el compromiso actual e incorpora los cambios por etapas, por lo que ya no se muestran como las diferencias entre el compromiso y el árbol de trabajo.
* los cambios no escenificados por lo tanto siguen mostrando como las diferencias, hasta que se añada al índice (o revertido).
* el índice aún se le puede decir a la etapa seleccionados "trozos" (parches o bloques de cambio) dentro de un archivo, dejando trozos de otros dentro del archivo mismo no escénica.
* Ramas y etiquetas: estas son formas de referirse a particular, se compromete, y viven en .git / árbitros y .git / envasados ​​de refs
* una etiqueta de marca un compromiso particular y sólo se ha movido nunca de forma manual; MEC otros que podríamos llamar una etiqueta.
* una rama es una etiqueta móvil especial que se actualiza cada vez que un compromiso se hace a esa rama.
* una rama es en realidad un sello que siempre está en el compromiso que la mayoría de SCM se llame a la sucursal / versión de HEAD.
* sucursales locales viven en .git / refs / cabezas
* las ramas de otros repositorios se mantienen en .git / refs / remotos

Las ramas y las etiquetas son muy baratos y casi idénticos - que se puede hacer cuando uno quiere recordar un punto particular en el desarrollo.

* .git / refs / jefes / Master contiene un identificador de 40 caracteres para la confirmación de que es la rama principal. Git es mucho metadatos en archivos de texto sin formato.
* Stash: el escondite ("git stash") es una lista especial de cometa se utiliza para mantener los cambios temporales.
* Configuración: información acerca de los repositorios remotos este repositorio se puede hablar, ¿cómo se relacionan con las ramas locales de las sucursales remotas, etc, vive en .git / config, en texto sin formato.
* Los ganchos: las secuencias de comandos especiales que se ejecutan para varios eventos de control de código fuente para hacer cumplir la política. Ganchos se encuentran dentro del directorio de git. y no por lo tanto bajo control de código fuente, es decir , no clonado a partir de repositorios remotos.
* La configuración global: Preferencias del usuario - nombre real, dirección de correo electrónico, las preferencias, los alias de comandos, etc - se llevan a cabo en el directorio home del usuario en un archivo llamado ~ / gitconfig.
* El ambiente de trabajo: El ambiente de trabajo se parece mucho a lo haría como si Git no estaba: el árbol de trabajo es un directorio normal con sólo la presencia sutil de un directorio de git para sugerir lo contrario.

Después de que el "git clone" en primer lugar o "git init" operación, git establece la "rama actual" para ser "maestro" y se asegurará de que el árbol de trabajo refleja el compromiso de que los "maestros" los puntos de ramificación. Cada uno tendrá su compromiso establecido en la matriz. Cometen actualmente marcado como "maestro" y "maestro" se trasladó a ese compromiso nuevo, listo para la siguiente.

1. **FIREBUG**

Firebug es una [extensión](http://es.wikipedia.org/wiki/Extensi%C3%B3n_(Mozilla)) de [Firefox](http://es.wikipedia.org/wiki/Firefox) creada y diseñada especialmente para desarrolladores y programadores web. Es un paquete de utilidades con el que se puede analizar (revisar velocidad de carga, estructura [DOM](http://es.wikipedia.org/wiki/DOM)), editar, monitorizar y [depurar](http://es.wikipedia.org/wiki/Depurar) el código fuente, [CSS](http://es.wikipedia.org/wiki/CSS), [HTML](http://es.wikipedia.org/wiki/HTML) y [JavaScript](http://es.wikipedia.org/wiki/JavaScript) de una página web de manera instantánea e inline.

Firebug no es un simple inspector como DOM Inspector, además edita y permite guardar los cambios, un paso por delante del conocido [Web Developer](http://es.wikipedia.org/wiki/Web_Developer). Su atractiva e intuitiva interfaz, con solapas específicas para el análisis de cada tipo de elemento (consola, HTML, CSS, Script, DOM y red), permite al usuario un manejo fácil y rápido. Firebug está encapsulado en forma de plug-in o complemento de [Mozilla](http://es.wikipedia.org/wiki/Mozilla_Application_Suite), es [Open Source](http://es.wikipedia.org/wiki/Open_Source), [libre](http://es.wikipedia.org/wiki/Software_libre) y de distribución gratuita.

1. **GOOGLE DEVELOPERS TOOLS**

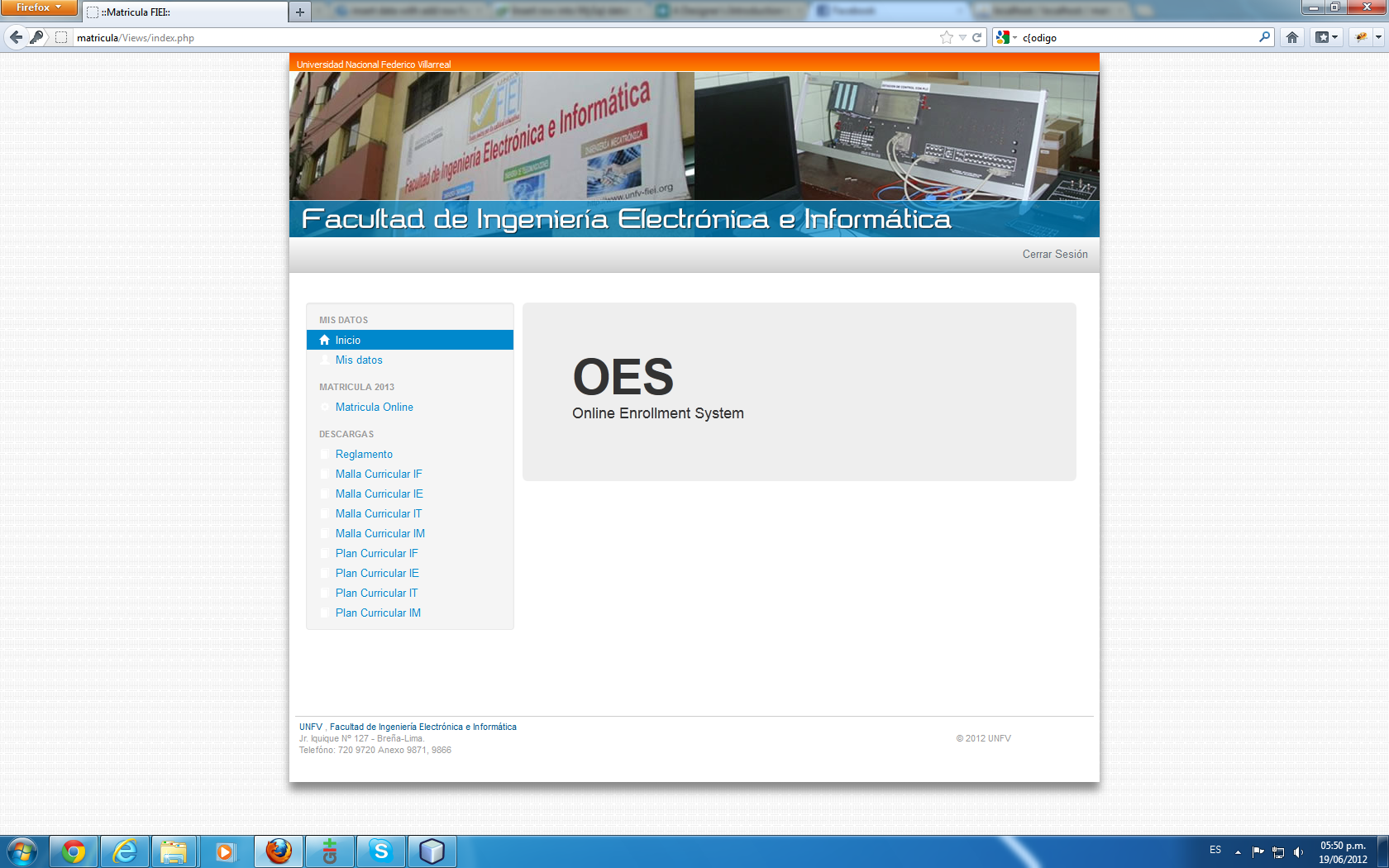
Es una herramienta de desarrollo que permite depurar nuestras páginas web, estableciendo un seguimiento y un control de los elementos de diseño que conforman la estructura de nuestros sitios, proporcionando un mecanismo para intentar solucionar los posibles problemas que se puedan presentar en un sitio web

**Descripción de funcionalidades**

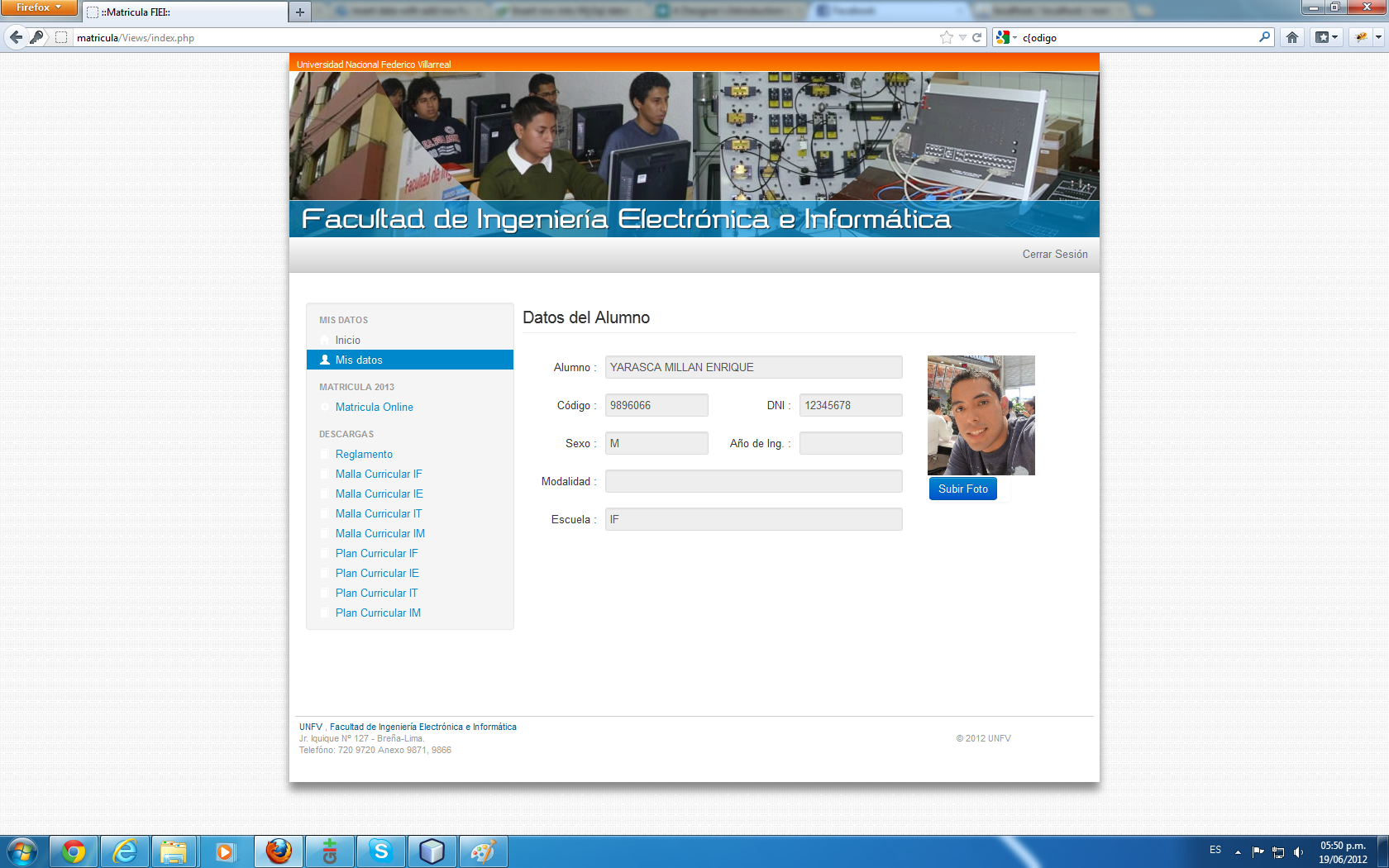
Antes de ver la parte más práctica de las Developer Tools, queremos indicar de forma teórica cuáles son sus principales características o puntos fuertes de uso:

* Integradas y fáciles de usar: Integradas, porque cada instalación viene con una instancia de las Developer Tools que nos va a permitir realizar depuración de los sitios web explorados. Se dispone además de la ventaja de poderlas utilizar cuando necesitemos limitar el impacto en el rendimiento que pueda producir un posible mal uso del código o de la programación existente que interprete el navegador. Fáciles de usar, gracias a una interfaz sencilla que además nos va a permitir realizar depuración de scripts para un determinado proceso del navegador en lugar de habilitar la depuración para toda la aplicación.
* Interfaz visual para la plataforma: Mediante la interfaz visual proporcionada podemos analizar y observar los códigos Scripts, HTML, CSS, etc. que se ejecutan dentro del navegador lo que nos va a permitir obtener de forma cómoda y rápida una visualización de la estructura de nuestros sitios. Esto nos repercute positivamente, reduciendo el tiempo que invertimos muchas veces en depurar determinados sitios dinámicos, donde la inspección de los mismos a través del código fuente no es fácil. Piénsese, también en aquellos casos donde el comportamiento de una determinada funcionalidad que se ejecuta, requiere de un análisis profundo que no puede proporcionarnos una herramienta de edición genérica como por ejemplo el simple pero todopoderoso Bloc de Notas.
* Habilitar experimentación rápida: Cuando se habla de experimentación rápida, siempre me viene a la mente algún diseñador realizando cambios sobre la estructura de una página, probando algún nuevo diseño o mejorando un determinado código. El proceso, antes de la existencia de las Developer Tools, consistía en editar el código, guardarlo y a continuación refrescar la página en el navegador, para más tarde volver a repetir todo el proceso. Ahora, mediante el uso de las Developer Tools, podemos acotar acortar este proceso editando nuestros sitios para ver los cambios de forma inmediata.
* Optimización del rendimiento de las aplicaciones: Mediante las Developer Tools vamos a poder identificar y solucionar posibles problemas potenciales de rendimiento. Gracias al analizador de scripts, es posible recopilar estadísticas y procesarlas en un informe que nos proporcione datos valiosos referidos al rendimiento. Ejemplos de ello pueden ser análisis del tiempo de ejecución invertido para la carga de determinada página web o el número de veces que se ha realizado una llamada a una determinada función JavaScript.

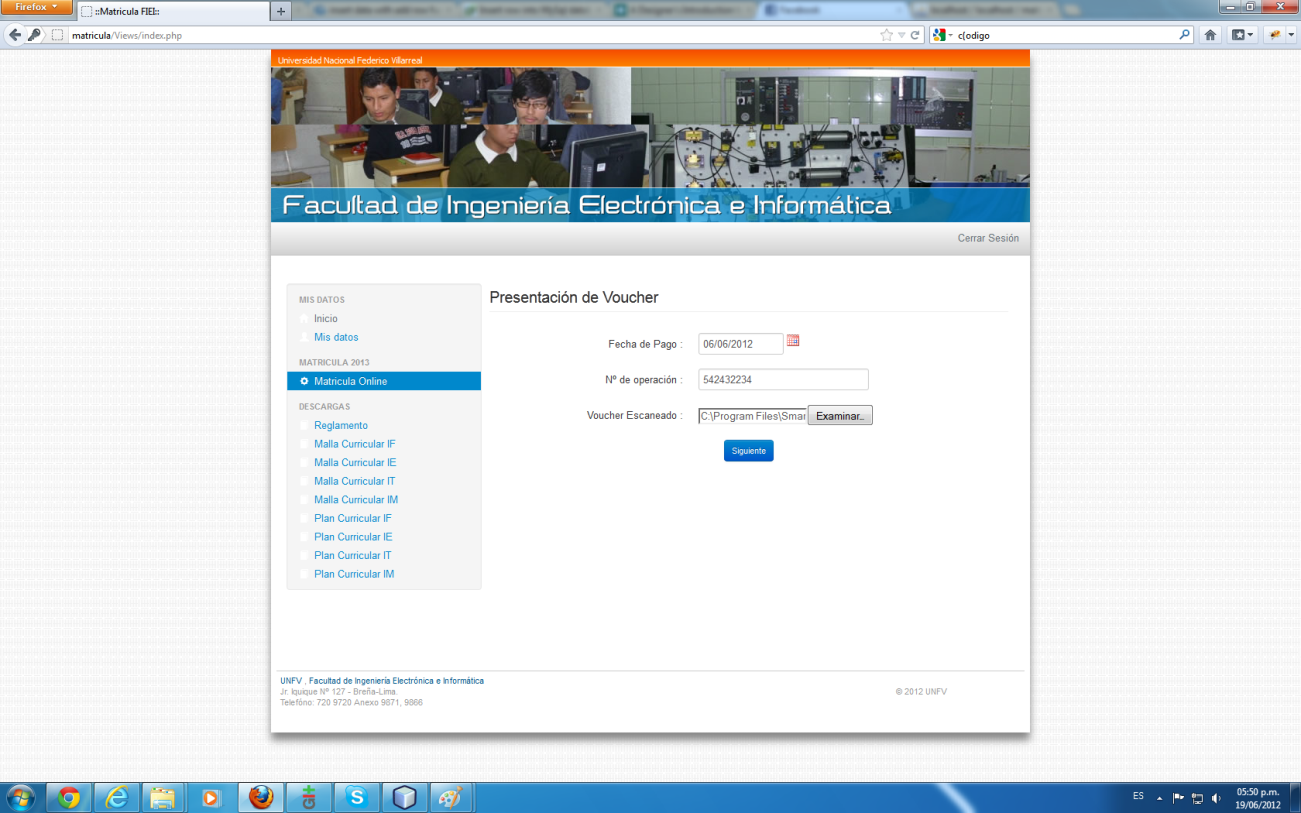
1. **IMAGENES DEL OES**
2. **BIENVENIDA AL OES**

****

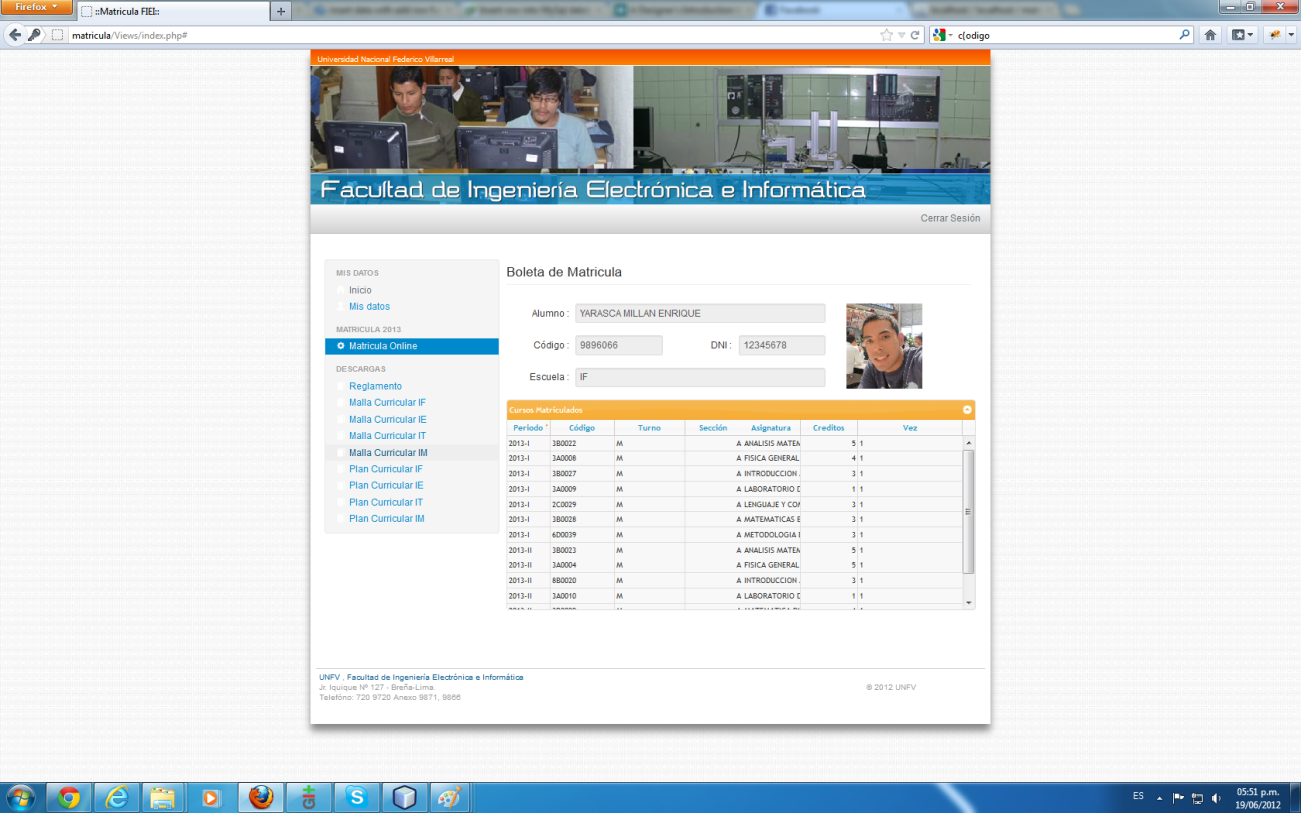
1. **DATOS DEL ALUMNO**

****

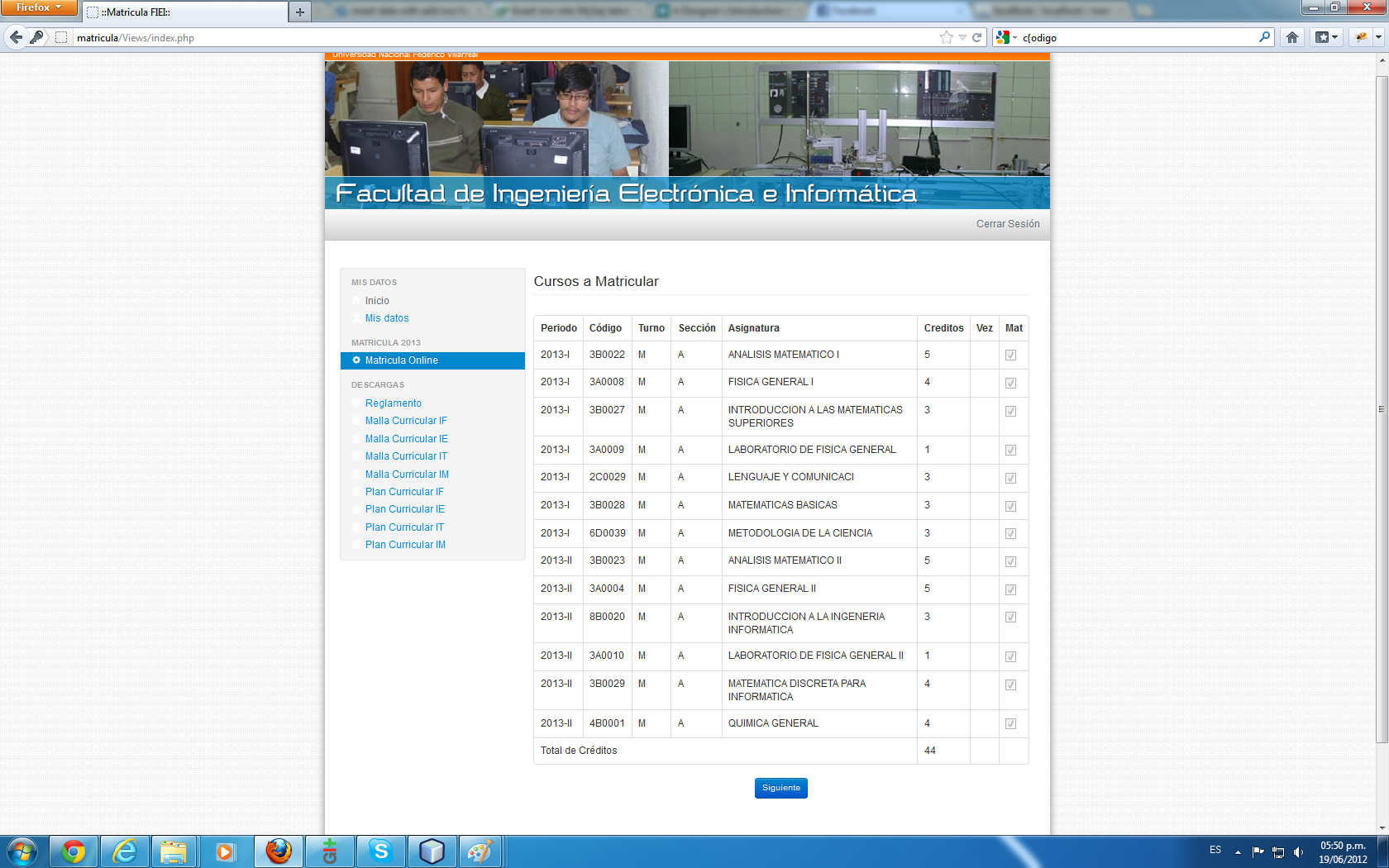
1. **PRESENTACION DEL VOUCHER DE PAGO**

****

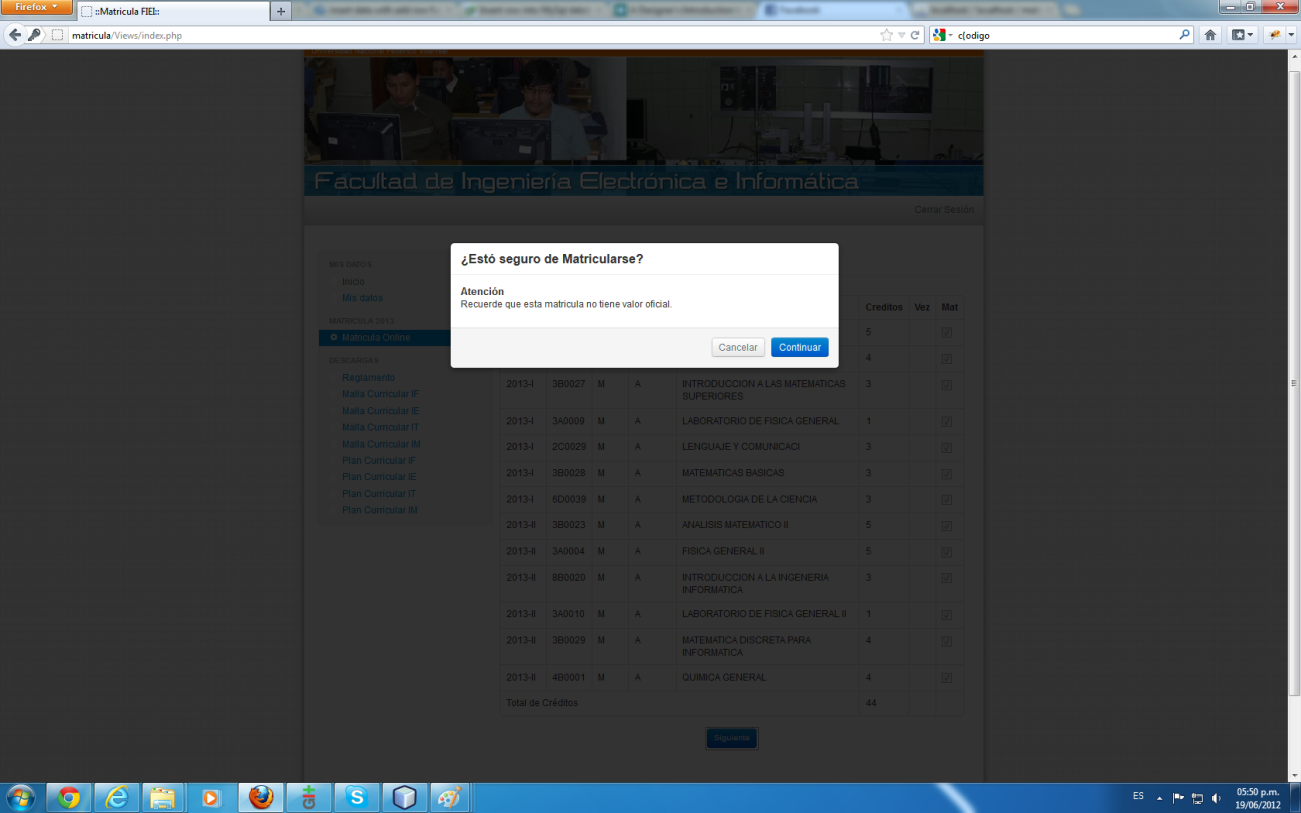
1. **MATRICULA ONLINE**

****

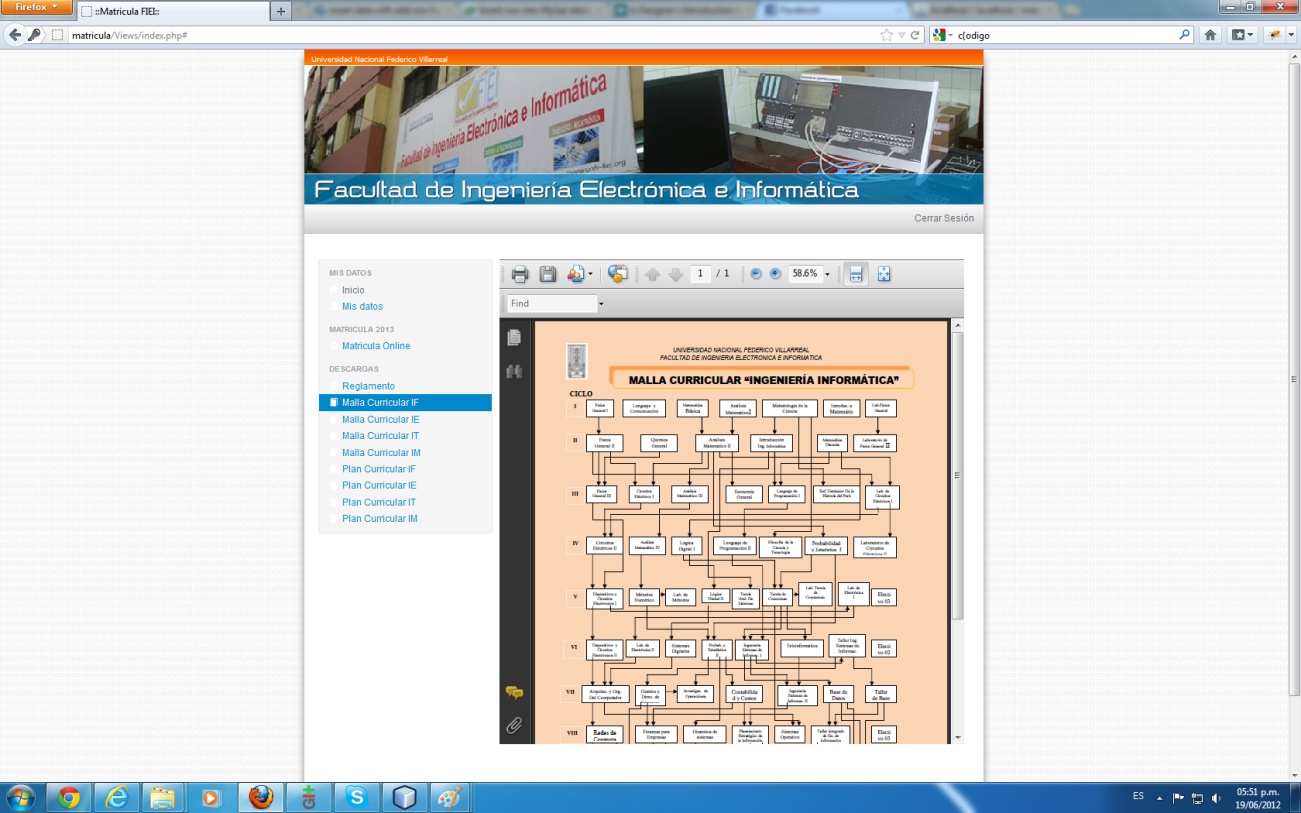
1. **CURSOS A MATRICULARSE**

****

1. **CONFIRMACION DE MATRICULA**

****

1. **DOCUMENTOS A DESCARGAR**

****

1. **ANEXOS**

**ENTREVISTA CON LA SRTA. IRMA (ENCARGADA DE LA OSA)**

1. ¿Cómo es el proceso de matrícula de los rezagados?

Es igual al proceso normal de matrícula regular y de ingresantes, solo que existe un recargo.

1. ¿Qué procesos son los más importantes en el proceso de matrícula actual?

No se pueden obviar ningún proceso, todos son importantes.

1. ¿De qué manera Ud. puede verificar la matrícula de un alumno? ¿Cuál es el proceso?

Si por alguna razón el alumno pierde la boleta de matrícula, se podrá verificar su matrícula a través del sistema.

1. ¿Qué dificultades ha tenido al momento de registro de la matrícula? (Alguna vez le ha aparecido un error del sistema)

No, nunca he tenido un error de sistema para este proceso.

1. ¿Qué problemas podrían ocurrir para que un alumno no se considere matriculado?

Que el alumno este suspendido, o se haya cancelado su matrícula.

1. ¿En qué casos se da la cancelación de la matricula?

Uno por no aprobar el mínimo de créditos por tres años consecutivos, el primer año es condicional, el segundo se suspende y el tercer año se cancela la matrícula. Otro es por haber desaprobado

1. ¿Cómo funciona la reserva de matrícula?

En el caso de los ingresantes tienen primero que matricularse para así acceder a la posibilidad de reservar su matrícula; mientras que los alumnos regulares pueden reservar sin necesidad de matricularse. Máximo un año de reserva.

1. ¿Puede el alumno escoger el turno de estudio? ¿Lo contempla el sistema?

El alumno no puede escoger su turno. Debe estar realizado antes del proceso de matrícula. Los encargados son los directores de escuela.

1. ¿Dónde queda guardada la documentación física que brinda el alumno (voucher, record, etc.)? ¿Ingresan esa documentación al sistema?

Todo la documentación que el alumno entrega al momento de sus matricula esta archivada en sus respectivas carpetas.

1. ¿Cuál es la función de los consejeros de matrícula? ¿Es fundamental la función de ellos?

Ayudar a corregir la matrícula de los alumnos. Su función ya no sería necesaria en el OES, pero esta normada su presencia.

1. ¿Cuántos empleados laboran para la organización en el área que se pretende desarrollar el sistema?

Para la matrícula, se necesita la colaboración de dirección de escuela y OSA

1. ¿Qué dificultades tienen con su sistema actual, y como cree se podría superar estas dificultades?

Hasta el momento no se han presentado problemas.

1. ¿El sistema actual permite la búsqueda de alumnos por apellidos? ¿Qué contempla el sistema?

Si, cuenta con todo lo necesario.

1. ¿Qué procesos del sistema actual, Ud. Encuentra innecesarios?

Ninguno, todos son necesarios.

1. ¿Cómo funciona la ampliación de créditos?

Se pueden ampliar hasta 4 créditos, la ampliación es bajo aprobación del decanato y tiene que estar invicto en el último año académico.

1. ¿Cómo es el procedimiento cuando un alumno desaprueba un curso?

Volverse matricular y pagar por el curso desaprobado, para llevar nivelación debe tener como nota mínima de 7, o dejar hasta la siguiente matricula regular.

1. ¿Qué pasa con un alumno que desaprueba 4 o más cursos?

Si en el ciclo desaprobó 4, se considera repitente.

Si durante el año desaprobó 2 en el primer semestre y otros dos en el segundo semestre, puede matricularse normalmente.

1. ¿El sistema actual se encarga de la matrícula del ciclo verano e interciclo?

Si, máximo de curso por nivelar 4, en toda la carrera el alumno puede adelantar 6 cursos.

1. ¿Qué es lo que usted desea con respecto al sistema OES?

Que haga más fáciles las cosas, que no haya mucho papeleo.

1. ¿Qué es lo que debe hacer el sistema?

Reducción de procesos.

1. Propiedades de las Pc’s de OSA.

Core 2 Duo.

**GLOSARIO DE TÉRMINOS**

**A**

**Ampliación de créditos,** solicitud que realiza un alumno que es invicto, es decir que no desaprobó ningún curso.

**B**

**Boleta de matrícula,** documento en el que se registra todos los cursos que el alumno va a llevar durante el año, con este el alumno acredita que se ha matriculado.

**C**

**Certificado de ingreso,** documento que acredita el ingreso del alumno a la universidad y en la facultad correspondiente.

**Consejeros de matrícula,** responsables de verificar que la conformidad de la matrícula.

**Crédito,** es la valoración que se le da a un curso dentro de un plan curricular.

**Cursos electivos,** son aquellos que no son considerados esenciales en la formación del alumno.

**Cursos regulares,** aquellos que se llevanobligatoriamente durante el año.

**D  
Dirección de escuela,** oficina encargada de entregar las carpetas de matricula a los alumnos, por medio de las secretarias de la escuela correspondiente.

**E**

**Escuela,** en la facultad existen 4 escuelas: Electrónica, Informática, Mecatrónica y Telecomunicaciones.

**Estado de no matriculado,** dentro del OES, es el estado inicial en que se encontraran todos los alumnos de la FIEI.

**Estado en proceso,** se dará cuando el alumno haya culminado su matrícula en el OES, y este a espera de la conformidad de su voucher de pago.

**Estado matriculado,** el alumno ha completa con todos los requisitos exitosamente y se puede considerar matriculado.

**Estado rechazado,** cuando ocurre algún error por parte del alumno al momento de registrar su matrícula.

**F**

**Ficha de solicitud de matrícula,** donde el alumno escribe todos los cursos que va a llevar tanto en el primer como en el segundo semestre.

**I**

**Instructivo de matrícula,** es una guía que el alumno puede considerar para realizar su matrícula.

**M**

**Matrícula cancelada,** se da cuando el alumno no ha cumplido con los créditos mínimos (10% del total) necesarios, durante tres años seguidos; o por desaprobar un curso tres veces.

**Matrícula condicional,** se da en los alumnos que no hayan aprobado con el mínimo de créditos, durante el ciclo anterior. También se da esta matrícula en los alumnos que hayan ingresado por traslado externo.

**Matrícula extemporánea,** se da cuando el alumno no se matriculada en los días oficiales, por lo que habrá un recargo económico.

**Matrícula regular,** proceso que realiza el alumno a partir del segundo año hasta la culminación de su carrera.

**O**

**Obligación de pago,** documento donde se registra el pago que debe realizar el alumno para su matrícula, o las deudas que este tiene con la universidad.

**OCA**, Oficina Central de Admisión, en donde los alumnos se apersonan a recoger su certificado de ingreso.

**OCRACC,** Oficina Central de Registros académicos y centro de cómputo, encargada de remitir las carpetas de matrícula.

**Operador de matrícula,** persona encargada de registrar la matrícula en el sistema.

**OSA,** Oficina de Servicios Académicos, encargada de realizar el proceso de matrícula y otros, como: rectificación de matrícula.

**P**

**Plan curricular,** diseño donde se planifica los cursos que va a llevar el alumno durante su periodo académico.

**Pre-requisito,** curso previo que debe haber aprobado un alumno, antes de seguir sus estudios en otro curso.

**R**

**Record académico,** documento en el cual están todas las notas del alumno desde su ingreso.

**Rezagado,** alumno que se retrasa en la matricula correspondiente en las fechas establecidas.

**T**

**Turno,** relacionado al horario de clases, el turno puede ser: mañana, tarde o noche.