

INTRODUCCION

El manual de Sistema pretende cubrir la explicación de todos los aspectos técnicos y arquitectónicos de EGA. Para ello se ha utilizado un conjunto de plantillas utilizadas en el Proceso Unificado Racional (RUP por sus siglas en inglés Rational Unified Process) para la documentación de software de una manera clara y estructurada, a fin de que en un futuro sirva como una pista para la solución de problemas y como guía para avances en el sistema.

El Desarrollo del sistema se divide en dos iteraciones, de manera tal de poder detallar todos los cambios que ha experimentado el sistema a lo largo de su desarrollo.

Al principio se brindan las instrucciones para asegurar la correcta instalación, funcionamiento y configuración del sistema, también se provee el código fuente y documentación básica para el mantenimiento propiamente dicho de este sistema.

En este manual, también presenta un glosario con toda la terminología relacionada al vocabulario técnico vinculado al sistema EGA, que podría presentar algún tipo de dificultad a la hora de poder entender este libro, ayudando a la fácil comprensión del lector.

MANUAL DE SISTEMA



ENTORNO DE GESTION ACADEMICA

PRIMERA ITERACION



ENTORNO DE GESTION ACADEMICA

NOCIONES PREVIAS

VERSION 0.1



ENTORNO DE GESTION ACADEMICA

FECHA:26/08/14

DOCUMENTO: NOCIONES PREVIAS

FICHA RESUMEN ENTORNO DE GESTION ACADEMICA

NOMBRE DE SISTEMA:

- EGA (ENTORNO DE GESTION ACADEMICA)

LOGOTIPO:



VERSION:

- v 1.0



PARADIGMA DE PROGRAMACION:

- MTV (MODEL- VIEW - TEMPLATE)

FRAMEWORK:

- **Nombre:** Django
- **Versión:** 1.6.5
- **Soporte:** <https://www.djangoproject.com/>

LENGUAJE DE PROGRAMACION

- **Nombre:** Python
- **Versión:** 2.7
- **Soporte:** <https://www.python.org/>

MOTOR DE BASE DE DATOS: POSTGRESQL

- **Nombre:** Postgresql
- **Versión:** 9.3
- **Soporte:** <http://www.postgresql.org/>

APLICACIONES DE TERCEROS:

- **UNIPATH**
 - **Version:** 1.0
 - **Soporte:** <https://pypi.python.org/pypi/Unipath/>



➤ PSYCOPG2

- **Versión:** 2.5.3
- **Soporte:** <http://initd.org/psycopg/>

➤ SOUTH

- **Versión:** 0.8.4
- **Soporte:** <http://south.aeracode.org/>

➤ PILLOW

- **Versión:** 2.4.0
- **Soporte:** <https://pillow.readthedocs.org/en/latest/>

➤ DJANGO-BRACES

- **Versión:** 1.4.0
- **Soporte:** <http://django-braces.readthedocs.org/en/v1.4.0/>

➤ SORL-THUMBNAIL

- **Versión:** 11.12
- **Soporte:** <http://sorl-thumbnail.readthedocs.org/en/latest/>



➤ WKHTMLTOPDF

- **Versión:** 0.2
- **Soporte:** <http://wkhtmltopdf.org/>

➤ REQUEST

- **Versión:** 2.3.0
- **Soporte:** <http://docs.python-requests.org/en/latest/>

➤ DJANGO - WKHTMLTOPDF

- **Versión:** 2.0.0
- **Soporte:** <https://readthedocs.org/projects/django-wkhtmltopdf/>



PREPARANDO EL ENTORNO

En esta sección prepararemos el entorno para establecer las condiciones óptimas para la ejecución de EGA, como así también su descarga y posteriormente su puesta en marcha.

A continuación se enumerarán una serie de pasos para lograr configurar nuestro entorno Linux, de manera tal, que podremos instalar EGA de forma correcta y sencilla, sin tener mayores complicaciones a la hora de su descarga e instalación.

Cabe destacar que para la preparación del entorno e instalación de EGA, se debe poseer ciertos conocimientos acerca del manejo en consola de Linux. También haremos hincapié en que las instrucciones deben ser llevadas a cabo como especifica este instructivo, como así también se debe seguir al pie de la letra los nombres de los usuarios, passwords, directorios y demás configuraciones.

PASO 1:

Como primer paso instalaremos **PIP**, esta es una herramienta de desarrollo para poder instalar y administrar paquetes de Python. Para ello antes deberemos instalar **PYTHON SETUP TOOLS**:

```
alejandro@alejandro-KJ-15-6: ~
alejandro@alejandro-KJ-15-6:~$ apt-get install python-setuptools
```

Una vez instalado **PYTHON SETUP TOOLS** ya podemos instalar **PIP** desde la terminal:



```
alejandro@alejandro-KJ-15-6: ~
alejandro@alejandro-KJ-15-6:~$ sudo easy_install pip
```

PASO 2:

En este paso instalaremos el entorno virtual en el cual correrá nuestro sistema, para ello instalaremos el **VIRTUALENV**

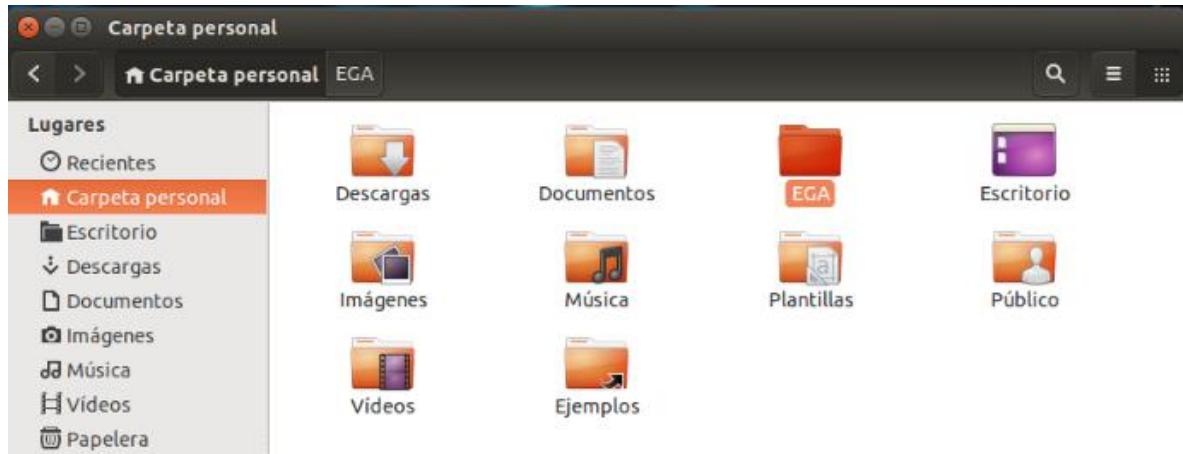
```
alejandro@alejandro-KJ-15-6: ~
alejandro@alejandro-KJ-15-6:~$ pip install virtualenv
```

PASO 3:

Una vez instalado **VIRTUALENV**, podremos crear nuestro entorno virtual de nombre “**EGA**” y posteriormente activarlo para correr la aplicación “**EGA**”. Entonces creamos primero lo que será nuestro entorno virtual:

```
alejandro@alejandro-KJ-15-6: ~
alejandro@alejandro-KJ-15-6:~$ virtualenv EGA
```

Se creara una carpeta en nuestra carpeta personal, que será nuestro entorno. Luego de creado solo deberemos activarlo de la siguiente forma



```
alexandro@alejandro-KJ-15-6: ~
alexandro@alejandro-KJ-15-6:~$ source EGA/bin/activate
(EGA)alexandro@alejandro-KJ-15-6:~$
```

PASO 4:

Ahora vamos a descargar nuestra aplicación desde un repositorio de **GIT HUB** para luego poder correr **EGA**. Cabe recordar que antes se deberá tener instalado **GIT**

```
alexandro@alejandro-KJ-15-6: ~
(EGA)alexandro@alejandro-KJ-15-6:~$ sudo apt-get install git
```

Una vez instalado GIT, descargamos nuestra aplicación a través de una clonación del repositorio



```
alexandro@alejandro-KJ-15-6: ~
(EGA)alexandro@alejandro-KJ-15-6:~$ git clone https://github.com/fabianfalon/PrototipoEGA.git
```



Una vez clonado nuestro proyecto debemos instalar las dependencias situándolos dentro de él y ejecutando el siguiente comando:

```
alexandro@alejandro-KJ-15-6: ~
alexandro@alejandro-KJ-15-6:~$ pip install -r base.txt
```

De esta manera instalaremos todo lo necesario para poder ejecutar **EGA**, esto sería instalar los **REQUIREMENTS**.



NOTA: Puede que el hecho de que no tengamos instalado Postgresql y sus paquetes dificulte la instalación de los REQUIREMENTS.

- Primero se instalaran los paquetes necesarios para instalar postgresql.

```
alejandro@alejandro-KJ-15-6: ~
alejandro@alejandro-KJ-15-6:~$ sudo apt-get install libpq-dev python-dev
```

- Luego se instalara **Postgresql**.

```
alejandro@alejandro-KJ-15-6: ~
alejandro@alejandro-KJ-15-6:~$ sudo apt-get install postgresql
```

PASO 5:

En este paso se creara y configurara la base de datos y su Usuario en **Postgresql**, En primer lugar iniciaremos el servidor de postgresql a través del siguiente comando:

```
alejandro@alejandro-KJ-15-6: ~
alejandro@alejandro-KJ-15-6:~$ sudo service postgresql start
[sudo] password for alejandro:
 * Starting PostgreSQL 9.3 database server [ OK ]
```



Luego de iniciado el servidor, accedemos a **Postgresql** por Terminal:

```
alejandro@alejandro-KJ-15-6: ~
alejandro@alejandro-KJ-15-6:~$ su postgres
Contraseña:
postgres@alejandro-KJ-15-6:/home/alejandro$
```

Una vez en acceditos pasamos a la creación de un Usuario

```
alejandro@alejandro-KJ-15-6: ~
postgres@alejandro-KJ-15-6:/home/alejandro$ createuser sistemaega
```

Ahora creamos la Base de Datos

```
alejandro@alejandro-KJ-15-6: ~
postgres@alejandro-KJ-15-6:/home/alejandro$ createdb sistemaega
```

PASO 6:

En este paso se configurara el acceso y la conexión a la Base de datos, en primer término se ingresara a la Base de datos:

```
alejandro@alejandro-KJ-15-6: ~
postgres@alejandro-KJ-15-6:/home/alejandro$ psql sistemaega
psql (9.3.5)
Type "help" for help.

sistemaega=#
```



Luego tendremos que asignarle un password al usuario que habíamos creado anteriormente, para poder trabajar en la Base de Datos.

```
alejandro@alejandro-KJ-15-6: ~
postgres@alejandro-KJ-15-6:/home/alejandro$ psql sistemaega
psql (9.3.5)
Type "help" for help.

sistemaega=# ALTER USER sistemaega WITH PASSWORD 'sistemaega';
ALTER ROLE
sistemaega=#
```

Con esto ya tendremos todos los datos necesarios para poder configurar el archivo **Local.py** de nuestra aplicación, con el fin de conectar con la Base de Datos

PASO 7:

En este paso sincronizaremos la base de datos para la creación de las tablas que indican los modelos de nuestra aplicación, esto una vez teniendo el entorno listo y la Base de datos creada, configurada y conectada a la aplicación, tal como hicimos en los pasos anteriores. Luego iniciaremos el servidor de Django para poder correr nuestra aplicación

Comenzaremos posicionándonos en el directorio de la aplicación donde se encuentra el archivo **manage.py**

```
alejandro@alejandro-KJ-15-6: ~/PrototipoEGA/ega/SistemaEGA
(EGA)alejandro@alejandro-KJ-15-6:~/PrototipoEGA/ega/SistemaEGA$
```



Luego sincronizaremos con la base de datos para actualizar los cambios de los modelos de nuestra aplicación

```
alejandro@alejandro-KJ-15-6: ~/PrototipoEGA/ega/SistemaEGA
(EGA)alejandro@alejandro-KJ-15-6:~/PrototipoEGA/ega/SistemaEGA$ python manage.py
syncdb --settings=PrototipoEGA.settings.local
```

Una vez con los cambios reflejados en nuestra base de datos, ejecutamos el servidor de Django.

```
alejandro@alejandro-KJ-15-6: ~/PrototipoEGA/ega/SistemaEGA
(EGA)alejandro@alejandro-KJ-15-6:~/PrototipoEGA/ega/SistemaEGA$ python manage.py
runserver --settings=PrototipoEGA.settings.local
```

ARCHIVO DE CONFIGURACIÓN

A continuación se detallara la configuración del módulo “settings”, que en sus principios era tan solo un archivo conocido como “settings.py” teniendo consigo todo los aspectos de la configuración del sistema, y que luego se dividió y modifco para que el mismo fuese un módulo en donde cada una de sus partes responde a una cierta configuración de un aspecto particular del sistema, cumpliendo la misma con las buenas prácticas de desarrollo de software.



ESTRUCTURA

```
FOLDERS
▼ settings
  __init__.py
  base.py
  email_info.py
  local.py
  production.py
  staging.py
  test.py
```

“__init__.py”: este archivo es para que el directorio sea tomado como un modulo.

“base.py”: aquí se declaran las configuraciones comunes para todas las instancias del proyecto.

“local.py”: esta es la configuración usada cuando se trabaja en un ambiente de desarrollo local.

“staging.py”: se utiliza antes de pasar el proyecto a producción.

“Production.py”: se usa para la configuración del servidor de producción.

“test.py”: configuración para realizar pruebas.



“local.py”

```
base.py *  
1  from .base import *  
2  
3  DEBUG = True  
4  
5  TEMPLATE_DEBUG = True  
6  
7  ALLOWED_HOSTS = []  
8  
9  
10 DATABASES = {  
11     'default': {  
12         'ENGINE': 'django.db.backends.postgresql_psycopg2',  
13         'NAME': 'sistemaega',  
14         'USER': 'sistemaega',  
15         'PASSWORD': 'sistemaega',  
16         'HOST': 'localhost',  
17         'PORT': '5432',  
18     }  
19 }  
20  
21  
22 STATIC_URL = '/static/'  
23 STATICFILES_DIRS = [BASE_DIR.child('static')]  
24 STATIC_ROOT = 'staticfiles'  
25 MEDIA_URL = 'http://192.168.0.105:8000/media/'  
26 MEDIA_ROOT = BASE_DIR.child('media')  
27  
28 #MANDRILL_API_KEY = "B7lPKBe39KmpXQTjtJF68g"
```



“base.py”

```
base.py  *
1 |from .email_info import EMAIL_USE_TLS, EMAIL_HOST, EMAIL_HOST_USER, EMAIL_HOST_PASSWORD, EMAIL_PORT
2 |from unipath import Path
3 |
4 |EMAIL_USE_TLS= EMAIL_USE_TLS
5 |EMAIL_HOST = EMAIL_HOST
6 |EMAIL_HOST_USER = EMAIL_HOST_USER
7 |EMAIL_HOST_PASSWORD = EMAIL_HOST_PASSWORD
8 |EMAIL_PORT = EMAIL_PORT
9 |
10|BASE_DIR = Path(__file__).ancestor(3)
11|
12|SECRET_KEY = '98_x--00^13)h%6td6%5o70&d6su-014$31xo*hr84pj18t3i3'
13|
14|DJANGO_APPS = (
15|    'django.contrib.admin',
16|    'django.contrib.auth',
17|    'django.contrib.contenttypes',
18|    'django.contrib.sessions',
19|    'django.contrib.messages',
20|    'django.contrib.staticfiles',
21|    )
22|
23|
24|THIRD_PARTY_APPS = (
25|    #'south',
26|    'djrill',
27|    'wkhtmltopdf',
28|
29|    )
30|
31|LOCAL_APPS = (
32|    'apps.alumno',
33|    'apps.home',
34|
35|
36|    )
37|
38|INSTALLED_APPS = DJANGO_APPS + THIRD_PARTY_APPS + LOCAL_APPS
39|MIDDLEWARE_CLASSES = (
40|    'django.contrib.sessions.middleware.SessionMiddleware',
41|    'django.middleware.common.CommonMiddleware',
42|    'django.middleware.csrf.CsrfViewMiddleware',
43|    'django.contrib.auth.middleware.AuthenticationMiddleware',
44|    'django.contrib.messages.middleware.MessageMiddleware',
45|    'django.middleware.clickjacking.XFrameOptionsMiddleware',
46|    )
47|
48|
49|ROOT_URLCONF = 'SistemaEGA.urls'
50|
51|WSGI_APPLICATION = 'SistemaEGA.wsgi.application'
52|
53|LANGUAGE_CODE = 'es-AR'
54|
55|
56|TIME_ZONE = 'America/Buenos_Aires'
57|
58|USE_I18N = True
59|
60|USE_L10N = True
61|
62|USE_TZ = True
63|
64|MEDIA_URL = 'http://192.168.0.105:8000/media/'
65|
66|TEMPLATE_DIRS = [BASE_DIR.child('templates')]
67|
68|AUTH_USER_MODEL = 'alumno.User'
69|
70|WKHTMLTOPDF_CMD_OPTIONS = {
71|    'encoding':'utf8',
72|    'quiet':True,
73|    }
74|
75|#EMAIL_BACKEND = "djrill.mail.backends.djrill.DjrillBackend"
```





ENTORNO DE GESTION ACADEMICA

DISEÑO DE DATOS

VERSION 1.0



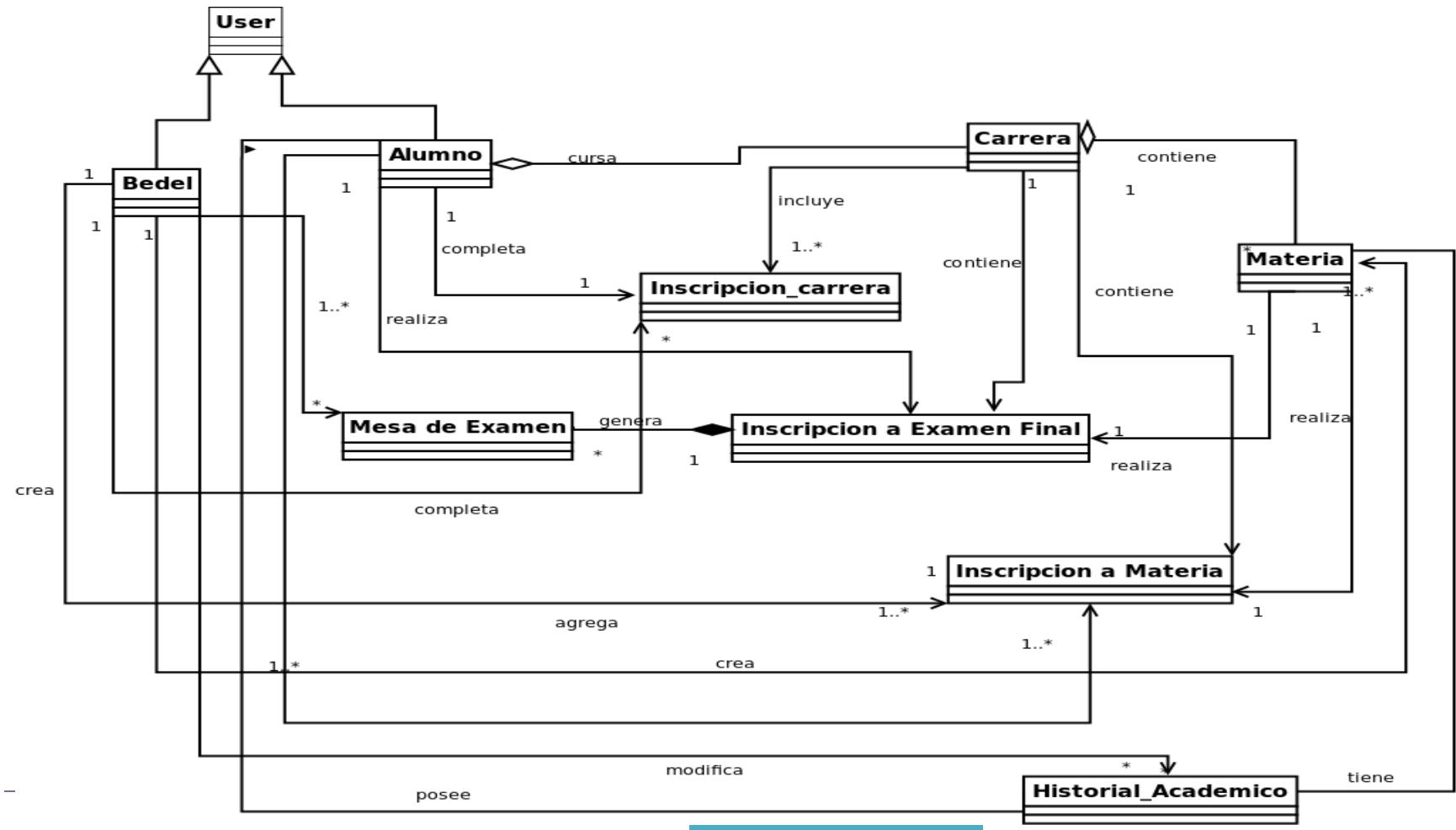
ENTORNO DE GESTION ACADEMICA

FECHA:26/08/14

DOCUMENTO: NOCIONES PREVIAS

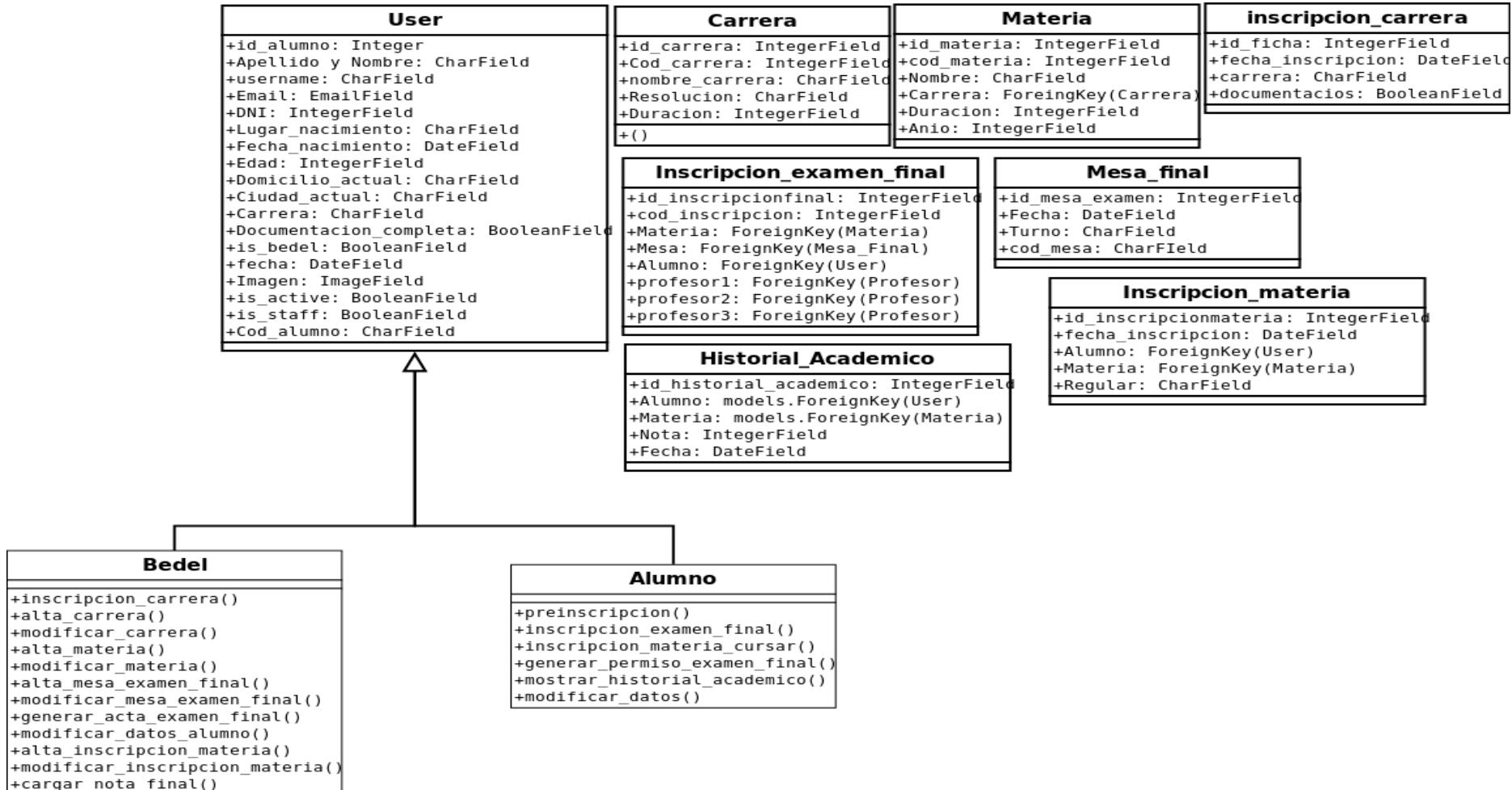


DIAGRAMA DE CLASES





CLASES





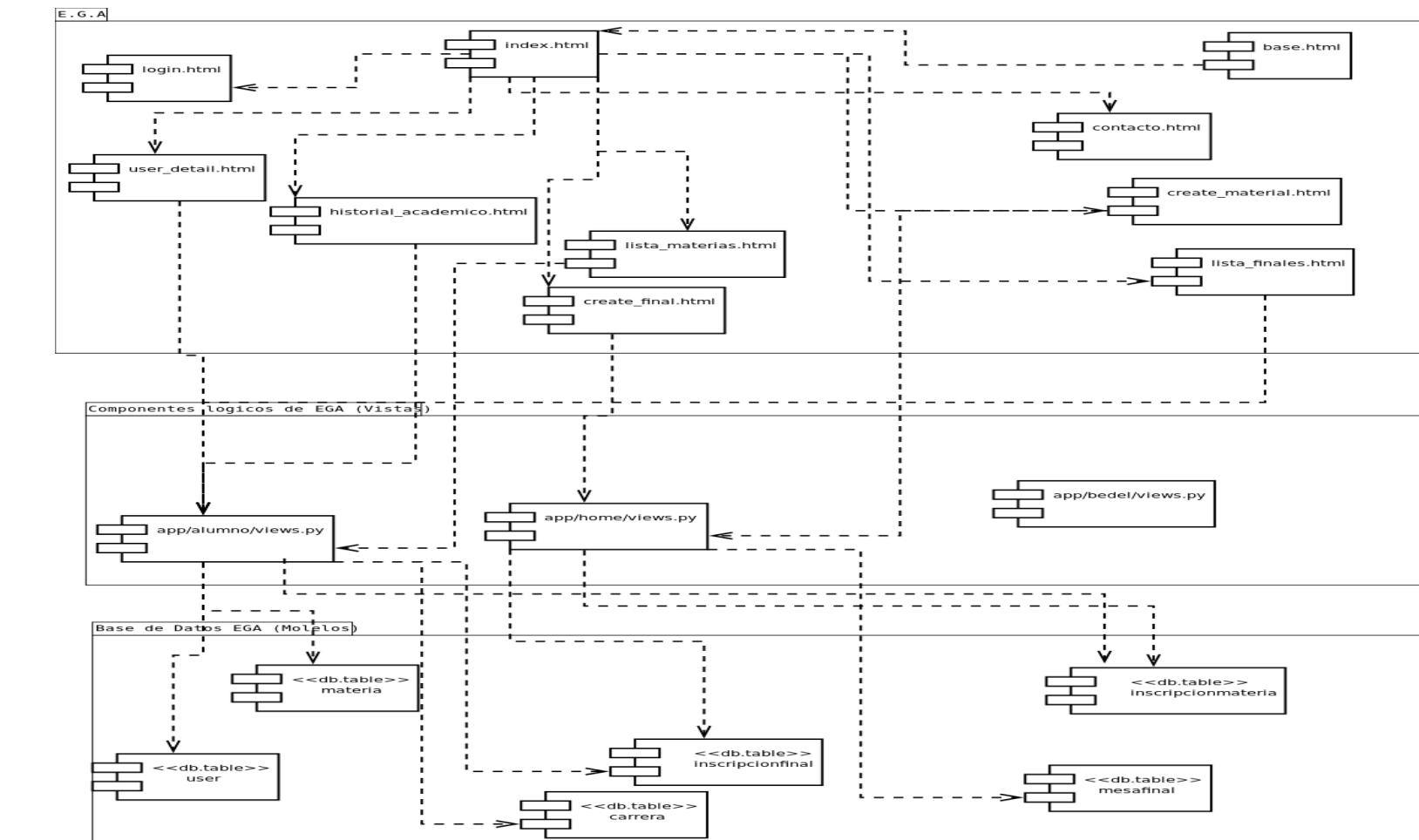
ENTORNO DE GESTION ACADEMICA

DISEÑO ARQUITECTONICO

VERSION 1.0



DIAGRAMA DE COMPONENTES







ENTORNO DE GESTION ACADEMICA

DIAGRAMAS DE CASOS DE USO

VERSION 1.0



FUNCIONES POR ACTOR

Alumnos:

- Preinscripción a carrera.
- Inscripción a materia por cursar.
- Inscripción a examen final.
 - Libre.
 - Regular
 - Promoción directa o indirecta.
- Visualizar historial académico.

Bedeles:

- Imprimir acta volante.
- Imprimir permiso de Examen
- ABM de exámenes finales.
- Verificar que los datos proporcionados por el alumno estén completos para luego inscribirlo.

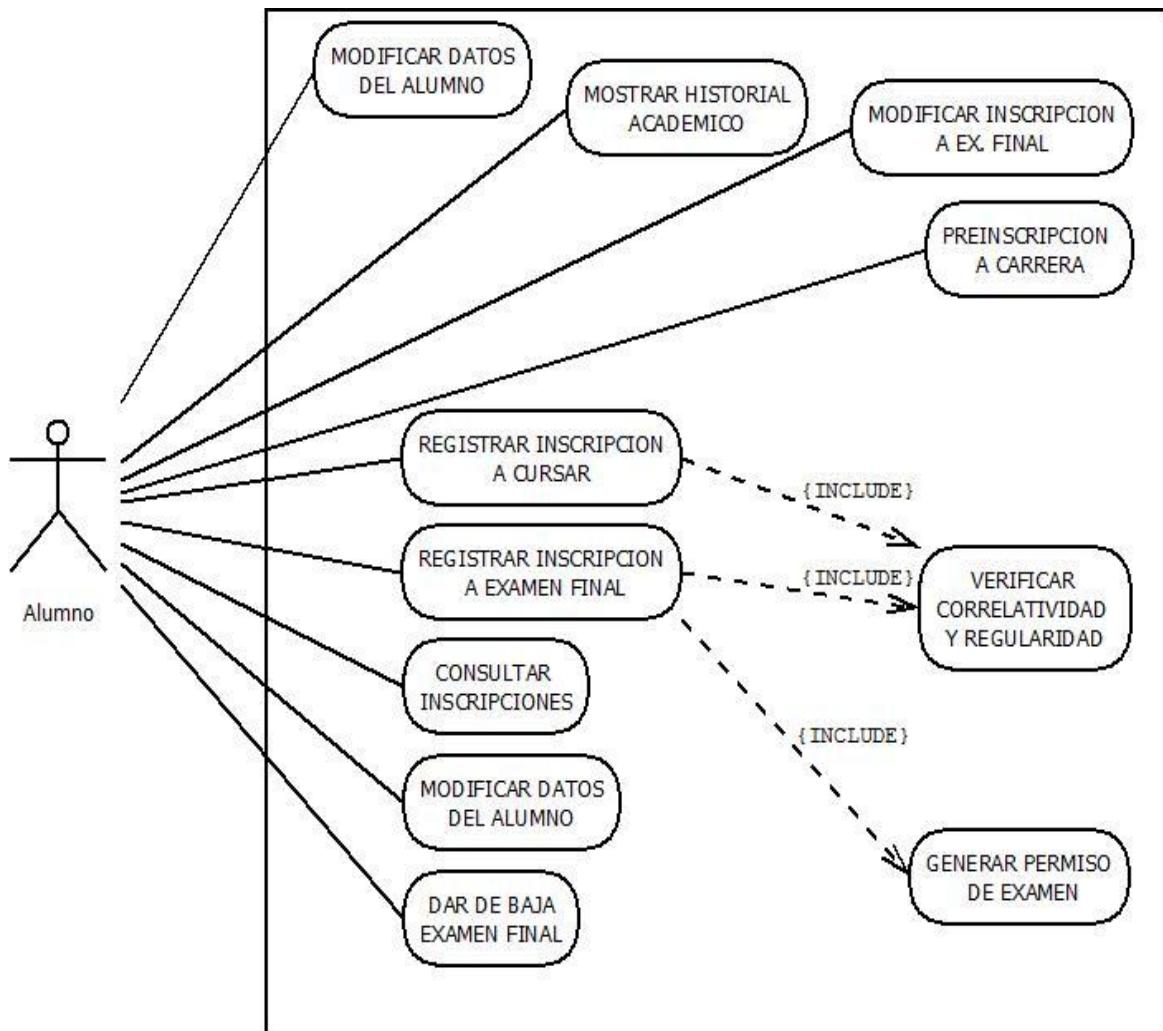
Administrador:

- ABM alumnos.
- ABM Profesores.
- ABM Carreras.
- ABM Materias.
- ABM Correlatividad.
- ABM Examenes.



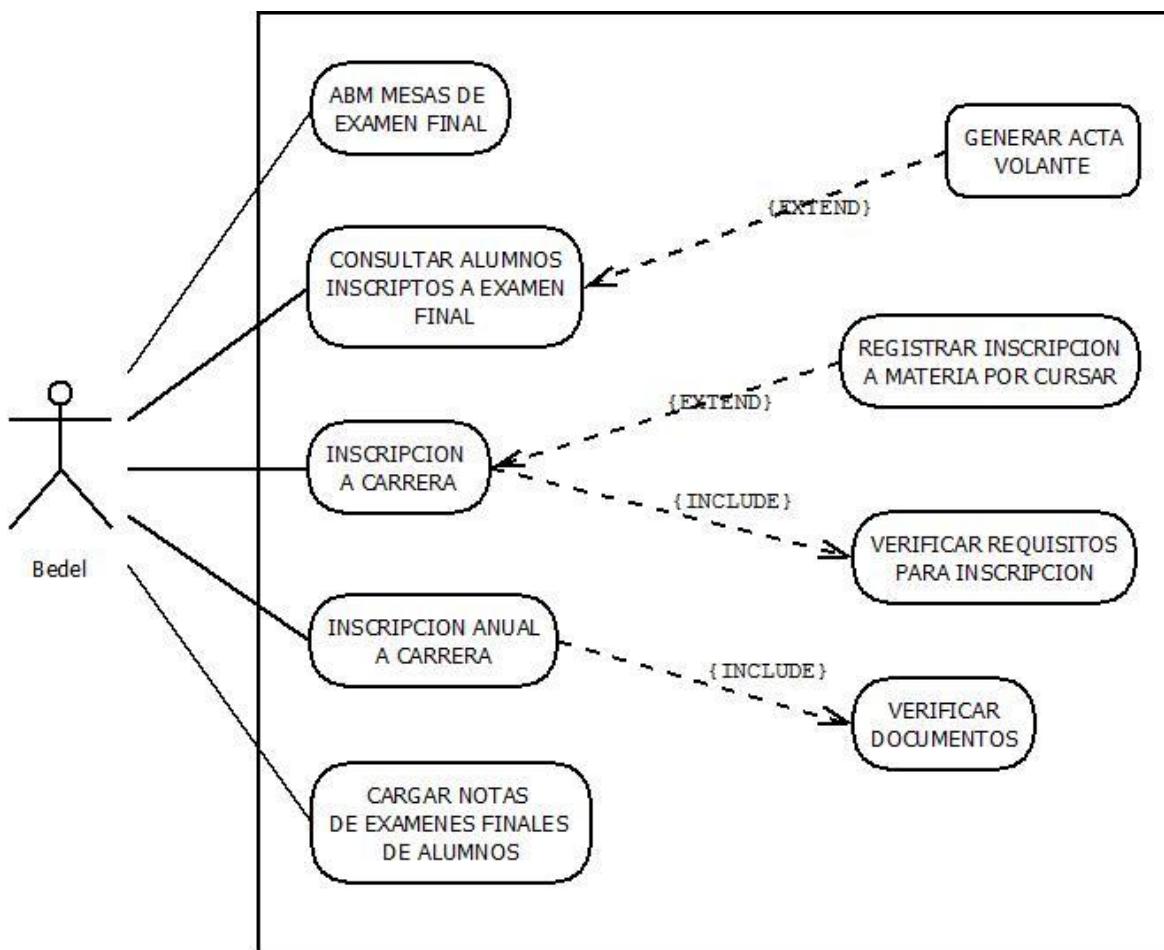
CASOS DE USO POR ACTORES

CASOS DE USO GENERAL DEL ALUMNO



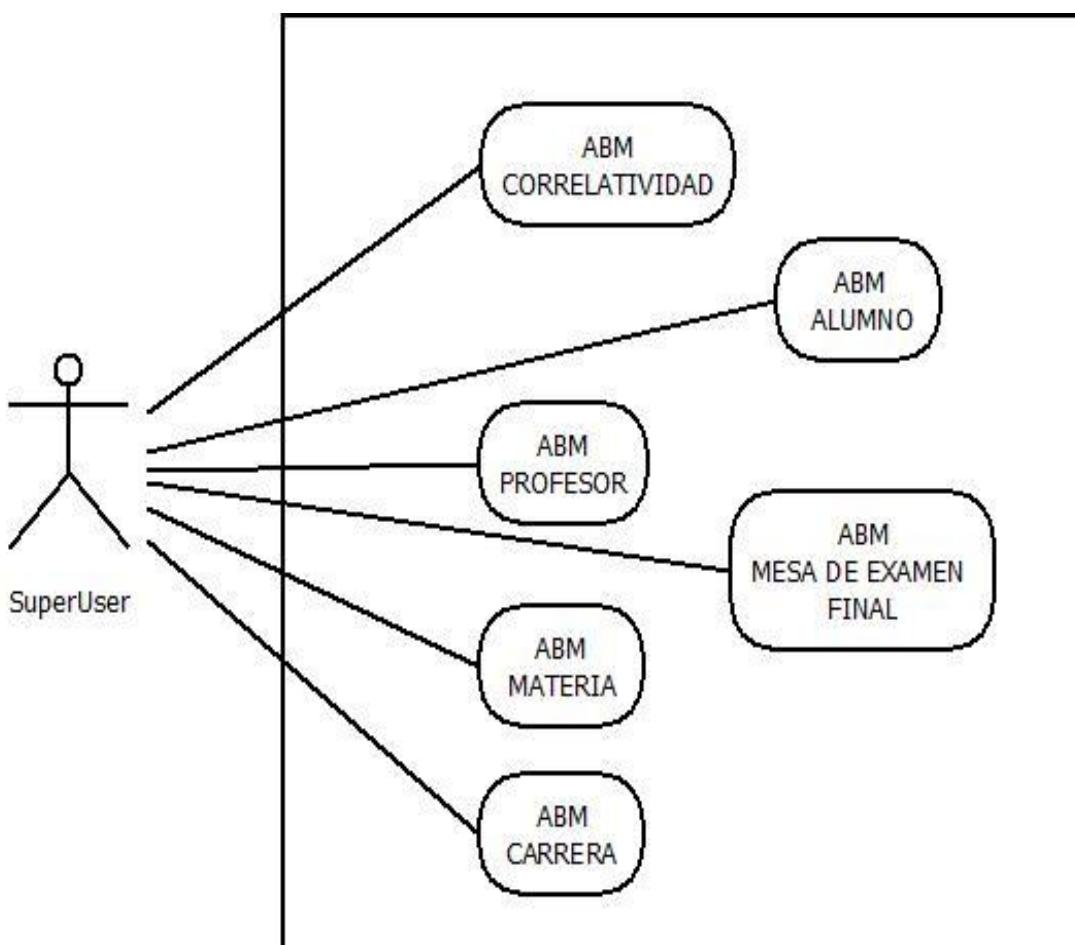


CASOS DE USO GENERAL DEL BEDEL





CASOS DE USO GENERAL DEL ADMINISTRADOR





ENTORNO DE GESTION ACADEMICA

FECHA: 26/08/14

DOCUMENTO: DIAGRAMAS DE CASOS DE USO



ENTORNO DE GESTION ACADEMICA

**ESPECIFICACION DE
REQUERIMIENTOS DE
SOFTWARE**

VERSION 1.0



Especificación de Requerimientos de Software

1. Introducción

1.1. Propósito

En el presente documento se describe de manera detallada y completa el comportamiento de E.G.A., observando el sistema desde el exterior, y no como desarrolladores. Incluye el conjunto de casos de uso, su descripción por trazo fino y trazo grueso conjuntamente con sus diagramas correspondientes que ayudarán a los usuarios a la mejor comprensión de todas las interacciones.

1.2. Definiciones, Acrónimos y Abreviaturas

- BD: Base de datos.
- EGA: Entorno de Gestión Académica.
- RC: Responsable de Confección
- RA: Responsable de Aprobación
- UD: Usuario Directo
- NT: Notificado
- ISFD: Instituto superior de formación docente.
- Regular: Alumno que aprobó todos los parciales, presentó y aprobó todos los trabajos prácticos y tiene 80% o más de asistencia a clases

1.3. Audiencia

UD: Cliente



RC: Falon Fabián, Peralta Alejandro, Rodríguez Brenda.

RA: Tribunal Evaluador.

1.4. Alcance

El alcance de este documento comprende la Especificación de Requerimientos de Software que debe satisfacer el sistema a desarrollar denominado “Entorno de Gestión Académica EGA” tanto aquellos requerimientos funcionales como no funcionales así como también las reglas de negocio aplicables. Las funcionalidades están categorizadas dependiendo de su grado de prioridad dentro del sistema.

Este documento está estrechamente relacionado con la documentación del ante proyecto de software y las entrevistas realizadas por el equipo de trabajo.

1.5. Referencias

<u>ID Archivo de Documento</u>	<u>Título del Documento</u>	<u>Número de Reporte</u>	<u>Fecha de Publicación</u>	<u>Organización que lo Publica</u>
01	Entrevista	-	-	-
02	Encuesta	-	-	-

1.6. Descripción General del Resto del Documento

El presente documento contiene las especificaciones de casos de uso, los diagramas de casos de uso e información detallada de cada caso de uso. También se describen los requerimientos funcionales y no funcionales del sistema a desarrollar, el objetivo, propósito y alcances del mismo.



2. Presentación del Producto

2.1. Propósito del Sistema

1. **Objetivo:** El objetivo principal de desarrollo del “Entorno de Gestión Académica EGA” es el de proporcionar a los alumnos del ISFD “Dr. José S. Salinas” un medio rápido, eficaz y seguro para realizar los trámites administrativos que exige la Institución, tales como la inscripción a las carreras que la misma ofrece, así como también la registración a un examen final, a una materia por cursar, entre otras.

2. Alcance:

El alcance del sistema dentro de la cobertura será:

- Agilizar trámites de inscripción a las diferentes carreras.
- Registrar datos de inscripciones de los estudiantes a las distintas carreras.
- Generar reporte sobre la inscripción de los estudiantes a los distintos espacios curriculares.
- Emitir informes a los docentes sobre la cantidad de alumnos a rendir en cada mesa de examen final, como así también de inscriptos para el cursado de las materias.
- Presentar listado de materias rendidas y regularizadas de cada estudiante.
- Proporcionar información rápida cuando los diferentes actores institucionales lo requieran.

No Contempla:

- Capacitaciones a alumnos del ISFD.
- Mantenimiento futuro del sistema.



2.2. Restricciones y Supuestos

Todas las herramientas utilizadas para el desarrollo del sistema EGA serán mediante licencias del software libre, por lo tanto la utilización de estos programas se hará bajo las políticas del licenciamiento correspondiente.

Las restricciones con las que cuenta la institución a la cual va destinada el software “E.G.A” están referidas a: limitaciones de hardware, a la poca experiencia técnica y nivel de conocimiento de los usuarios administrativos a los que va destinado la aplicación, y por ultimo económicas, las cuales influyen en el desarrollo del sistema.

Además, se cuenta con que los usuarios del sistema, tanto administrativos como alumnos contarán con acceso a Internet no menor a 1 MB.

3. Descripción General

3.1. Listado de la Funcionalidad del Sistema



Lista de casos de usos

Nº	Nombre	Categoría	Complejidad
1	Dar de Baja a examen Final	Esencial	Media
2	Buscar Materias	Esencial	Media
3	Generar Acta	Esencial	Media
4	Generar Comprobante	Esencial	Media
5	Iniciar Sesion Login	Esencial	Media
6	Inscribir a examen final	Esencial	Media
7	Modificar Alumno	Esencial	Baja
8	Modificar Contraseña	Esencial	Baja
9	Modificar Mesa de Examen finales	Esencial	Media
10	Modificar datos de Cuenta de usuario	Esencial	Media
11	Mostrar Historial Académico	Esencial	Media
12	Registrar alumno	Esencial	Media
13	Registrar carrera	Esencial	Alta
14	Registrar Materia	Esencial	Media

**ENTORNO DE GESTION ACADEMICA**

FECHA:26/08/14

DOCUMENTO: ESPECIFICACION DE REQUERIMIENTOS DE SOFTWARE

15	Registrar Mesa de Examen Final	Esencial	Media
16	Emitir acta	Esencial	Baja
17	Seleccionar alumno	Esencial	Media
18	Seleccionar Carrera	Esencial	Media
19	Seleccionar mesa de Examen Final	Esencial	Media
20	Registrar Correlatividad	Esencial	Baja
21	Registrar Inscripción a Cursada	Esencial	Baja
22	Imprimir Comprobante	Esencial	Baja
23	Buscar Alumno	Esencial	Baja
24	Buscar Fecha de Examen final	Esencial	Baja



3.2. Listado de Actores

<u>Nombre del Actor</u>	<u>Descripción</u>	<u>Tipo</u>	<u>Categoría</u>
Bedel	Encargado de administrar la gestión académica, de manera tal que facilite el correcto funcionamiento del ISDF.	Concreto	Visual
Alumno	Encargado de interactuar con el sistema de tal manera que pueda realizar sus trámites administrativos de manera eficiente durante el transcurso de su carrera en el ISFD.	Concreto	Visual
Super Usuario	Encargado de llevar a cabo la administración del Sistema para que su desempeño sea el óptimo cuando alguno de los actores lo requiera.	Concreto	Visual

3.3. Perspectiva del Producto

Se proyecta implementar un sistema de información que permita la realización de una manera más rápida, eficaz y segura la gestión administrativa de ISFD, será un producto diseñado para interactuar en un entorno web, lo que permite su manejo en forma descentralizada. El sistema de información a implementarse es un software independiente, ya que no tendrá relación con otros sistemas.

4. Descripción Detallada de Requerimientos

4.1. Requerimientos Funcionales

- Iniciar Sesión
- Preinscripción e Inscripción a la Carrera
- Consultar Datos Personales.



- Consultar Historial Académico.
- Modificar Datos personales.
- Registrar Usuarios (Alumnos, Administradores, Bedel) en el Sistema.
- Registrar Carreras.
- Registrar Materias.
- Registrar inscripción a examen Final.
- Registrar mesas de examen final.
- Eliminar inscripción a examen Final.
- Registrar inscripción a materia por cursar.
- Modificar Inscripción a Examen Final.
- Registrar Correlatividad.
- Registrar Notas de Alumnos.
- Administrar notas de los exámenes de los alumnos durante el cursado de las materias.
- Administrar historial académico de cada alumno visualizando materias aprobadas y desaprobadas.
- Agregar mesa de exámenes finales para que puedan ser visualizada por los alumnos.
- Emitir Acta volante y Permiso de examen.

4.2. Reglas y Funciones de Negocio

- La inscripción de un alumno a un examen final solo podrá realizarse si el alumno esta en condición regular (Con nota mayor a 6, 80% de asistencia y los trabajos prácticos presentados y aprobados) en la materia que desea inscribirse.
- La inscripción para realizar las observaciones solamente podrá realizarse si el alumno posee todas las materias de primer año aprobado.
- La inscripción para realizar las practicas solamente podrá realizarse si el alumno posee todas las materias de segundo año aprobado.



- La inscripción a una materia por cursar solo podrá realizarse si el alumno tiene regular (Con nota mayor a 6, 80% de asistencia y los trabajos prácticos presentados y aprobados) la materia correlativa anterior a la que se quiere inscribir.

4.3. Requerimientos No Funcionales

- Capacidad
- Disponibilidad
- Seguridad

4.3.1. Del Producto

Usabilidad: se incluyen todos aquellos requerimientos que afectan la usabilidad del sistema (facilidad de aprendizaje, facilidad de uso, flexibilidad y robustez):

- **Tiempo requerido para capacitación de usuarios normales y usuarios expertos:** Dependiendo del perfil del usuario y nivel de experiencia el tiempo promedio mínimo requerido se estima de no menos de 14 días, con una carga horaria de 3hs diarias (días hábiles).
- **Efectividad.**
- **Eficiencia.**
- **Atractivo.**

Confiabilidad: la confiabilidad podría expresarse en término de alguno de estos aspectos:



- Disponibilidad: Para el uso de alumnos, bedeles y administradores el sistema estará disponible las 24 hs de días y durante todo el año lectivo.

Performance:

- Tiempo de respuesta para una transacción: Dependiendo del ancho de banda disponible en los momentos de transacción y la cantidad de usuarios conectados.
- Capacidad: Máximo de inscriptos en el ISFD.
- Utilización de Recursos mínimos aceptables: 512 GB. memoria RAM, Procesador Intel Pentium/ AMD Athlon, navegador web que soporte HTML5, ancho de banda no menor a 1 MB.

Documentación: Manual de Usuario impreso.

4.3.2. Del Ambiente

Ético:

El administrador del sistema tiene prohibido:

1. Nombrar, elegir o tener a su servicio en forma permanente o transitoria a cualquier persona ajena a la institución.
2. Permitir, tolerar o facilitar ejercicios incompatibles con la confiabilidad del sistema.



3. Solicitar o aceptar comisiones de dinero para que preste sus servicios institucionales.
4. Ejecutar actos de violencia, malos tratos, calumnias o injurias contra compañeros de trabajo (bedeles, directores y/o profesores) o alumnos de la institución educativa.
5. Causar intencional o culposamente pérdida de información.
6. Expedir, permitir o contribuir para que se expidan títulos, diplomas, matrículas, certificados de inscripción, de regularidad o cualquier otro documento a personas que no reúnan los requisitos para ello.
7. Ejecutar actos que atenten a la moral y a las buenas costumbres.

Lo antes mencionado también debe ser considerado para los bedeles de la institución educativa.

Legales:

Adjunto Anexo Nro. 24.521: Ley Nacional de Educación Superior

4.4 Requerimientos de Hardware

- Los requerimientos necesarios para un óptimo funcionamiento del servidor son:
 - Procesador Dual Core en adelante
 - Memoria 2GB en adelante
 -



4.5 Requerimientos de Software

- **Los requerimientos del servidor son:**
 - Sistema operativo Debian o Ubuntu Server.
 - PostgreSQL
 - Python
 - Django
 - Virtualenv

4.6 Requerimientos de interfaz

EGA estará dividido en 3 entornos dependiendo el tipo de usuario logeado, los cuales pueden ser alumno, bedel o administrador, cada uno de ellos con permisos y disponibilidad de funcionalidades diferentes teniendo en cuenta las tareas que desempeñan dentro de la institución. Los entornos de bedel y alumno estarán orientados al manejo sencillo e intuitivo del sistema, mientras que el administrador, considerando que el mismo será un administrador de red experimentado (el cual ya se encuentra en la institución) tendrá una interfaz menos estética que la de los otros usuarios.

Cada una de estas interfaces no tendrá relación alguna entre ellas.

En este nivel, no se define de manera detallada la interfaz, solo se pretende tener una primera aproximación a los elementos que deben ser tenidos en cuenta en el desarrollo de estas.



4.6.1 Interfaz de usuario

La interfaz de usuario bedel y alumno debe ser orientada a ventanas, intuitiva y de fácil uso. Deberá tener alto grado de usabilidad y ser sencilla, cumpliendo todos los requisitos especificados. Incluirán menús en donde ellos podrán seleccionar la opción correspondiente a la transacción académica que desean realizar.

4.6.2 Interfaz de hardware

Permiten la interacción hombre-máquina. Mouse y teclado. Pantalla táctil, impresora mediante la cual se podrá obtener actas volantes, permisos, historias académicas y demás informes que emitirá el sistema

4.6.3 Interfaz de software

No aplica.

**PLANTILLAS DE CASO DE USO: ACTOR ALUMNO****PLANTILLAS TRAZO FINO**

DAR DE BAJA A EXAMEN FINAL		NRO. DE ORDEN: 001
		PAQUETE: ALUMNO
Actor Principal: Alumno		Actor Secundario: No Aplica
Nivel de Caso de Uso:		<input type="checkbox"/> Negocio <input checked="" type="checkbox"/> Sistema de Información
Tipo de Caso de Uso:		<input checked="" type="checkbox"/> Concreto <input type="checkbox"/> Abstracto
Prioridad:		<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Útil <input type="checkbox"/> Deseable
Categoría: <input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Soporte		Significativo para la Arquitectura: <input checked="" type="checkbox"/> Si <input type="checkbox"/> No
Complejidad: <input type="checkbox"/> Simple <input checked="" type="checkbox"/> Mediano <input type="checkbox"/> Complejo <input type="checkbox"/> Muy Complejo <input type="checkbox"/> Extremadamente Complejo		
Objetivo: El alumno debe poder anularse de un examen final en el cual se inscribió anteriormente.		
Precondiciones: El alumno debe estar inscripto en el examen final del cual desea anularse.		
Post- Condiciones:	Éxito 1: El alumno se dio de baja al examen final correctamente. Fracaso 1: El alumno selecciona la opción “No, cancelar”. Fracaso 2: El alumno no pudo realizar la operación debido a que ya caducó el tiempo para ello. (5 días hábiles antes de la mesa de examen final). Fracaso 3: El alumno no puede realizar la operación debido a que no está inscripto en la mesa de examen final.	
Curso Normal		Alternativas
1. Actor: El caso de uso comienza cuando el alumno inscripto a un examen final, decide eliminar la inscripción a dicha materia. El alumno hace click en la pestaña “Consultar Inscripciones”		
2. Sistema: Muestra una lista de opciones: consultar”Consultar Inscripción a Examen final”, ”Consultar Inscripcion a Materia por Cursar”, ”Certificado de Alumno Regular”		
3. Actor: Selecciona la opción ”Consultar Inscripción a Examen final”		
4. Sistema: Muestra la grilla de materias a las cuales el alumno está inscripto para rendir el examen final		



5. Actor: hace click en el botón eliminar de la materia a la cual desea darse de baja, ubicado a la derecha de cada registro	
6. Sistema: muestra un mensaje: “¿Estas seguro de eliminar inscripción a la MATERIA?”,en conjunto con los botones “Confirmar” y “No, Cancelar”	
7. Actor: El actor selecciona la opción “Confirmar”	El actor selecciona la opción “No, cancelar”
8. Sistema: Lleva a cabo la operación y re direcciona a la lista de las materias para inscribirse, donde se puede ver que es posible volver a inscribirse a la materia dada de baja	Sistema: Lleva a cabo la operación y re direcciona a la lista de las materias en las cuales el alumno está inscripto, y que pueden darse de baja
9. Fin del caso de uso.	

Observaciones:

1. **El Alumno** puede cancelar la operación en cualquier momento.

Requerimientos no Funcionales Asociados: No Aplica**Reglas de Negocio Asociadas:** No Aplica**Fuente:** Reunión, Entrevista.**Referencia Fuente:** Entrevista.doc**Asociaciones de Inclusión:** No Aplica**Asociaciones de Extensión:** No Aplica**Use case donde se incluye:** No Aplica**Use Case al que extiende:** No Aplica**Use Case de Generalización:** No Aplica



PREINSCRIPCION A CARRERA

NRO. DE ORDEN: 002

PAQUETE:ALUMNO

Actor Principal: Alumno

Actor Secundario: No Aplica

Nivel de Caso de Uso: Negocio Sistema de InformaciónTipo de Caso de Uso: Concreto AbstractoPrioridad: Esencial Útil DeseableCategoría: Esencial Soporte Significativo para la Arquitectura: Si NoComplejidad: Simple Mediano Complejo Muy Complejo Extremadamente Complejo

Objetivo: El alumno debe poder preinscribirse en la carrera que escogió para cursar.

Precondiciones: No Aplica

Post- Condiciones: Éxito 1: El alumno se preinscribió satisfactoriamente y debe aguardar La fecha para presentar la documentación en bedelia para que su inscripción quede registrada.

Fracaso 1: El alumno selecciona la opción cancelar.

Curso Normal	Alternativas
10. El caso de uso comienza cuando el alumno ingresa al sistema para inscribirse a una carrera en el ISFD.	
11. El sistema muestra un Formulario de Preinscripción.	
12. El alumno completa los datos (Username, Email, Nombre y Apellido, DNI, Lugar de Nacimiento, Fecha de Nacimiento, Ciudad Actual, Domicilio Actual, Imagen, Password, Carrera) que se solicitan en el formulario.	El alumno no completa algún campo del formulario, el sistema muestra un mensaje indicando “campo obligatorio”.
13. El alumno selecciona la opción “Guardar”	El alumno selecciona la opción cancelar.
14. El sistema almacena los datos registrados por el alumno.	
15. Fin del caso de uso.	

Observaciones:

2. El alumno puede cancelar la operación de preinscripción en cualquier momento.



Requerimientos no Funcionales Asociados: No Aplica

Reglas de Negocio Asociadas: No Aplica

Fuente: <Minuta, Reunión, Entrevista, Documento, etc>

Referencia Fuente: <Nombre de Minuta, Documento, etc. >

Asociaciones de Inclusión: No Aplica

Asociaciones de Extensión: No Aplica

Use case donde se incluye: No Aplica

Use Case al que extiende: No Aplica

Use Case de Generalización: No Aplica



REGISTRAR INSCRIPCION A MATERIA POR CURSAR		NRO. DE ORDEN: 003
		PAQUETE:ALUMNO
Actor Principal: Alumno		Actor Secundario: No Aplica
Nivel de Caso de Uso:		<input type="checkbox"/> Negocio <input checked="" type="checkbox"/> Sistema de Información
Tipo de Caso de Uso:		<input checked="" type="checkbox"/> Concreto <input type="checkbox"/> Abstracto
Prioridad:		<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Útil <input type="checkbox"/> Deseable
Categoría: <input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Soporte		Significativo para la Arquitectura: <input checked="" type="checkbox"/> Si <input type="checkbox"/> No
Complejidad: <input type="checkbox"/> Simple <input checked="" type="checkbox"/> Mediano <input type="checkbox"/> Complejo <input type="checkbox"/> Muy Complejo <input type="checkbox"/> Extremadamente Complejo		
Objetivo: El alumno debe poder inscribirse a un espacio curricular para cursar la materia que desee.		
Precondiciones: El alumno debe tener aprobada o regularizada la correlativa anterior.		
Post- Condiciones:	Éxito 1: El alumno se inscribió al espacio curricular de manera satisfactoria. Fracaso 1: el alumno aborta la operación, haciendo click en cancelar o volviendo a la pantalla principal. Fracaso 2: no se pudo concretar la operación debido a que no se cuenta con las condiciones de correlatividad necesarias.	
Curso Normal	Alternativas	
16. Actor: El caso de uso comienza cuando hace click en la opción “inscripción a cursar”, en el menú “inscripciones”		
17. Sistema: muestra un vista con la grilla de materias, en las cuales se puede inscribir para poder realizar el cursado de las mismas		
18. Actor: selecciona la materia a cursar de la grilla de materias, luego aprieta el botón “inscribirse”.	Actor: 1. Selecciona una materia la cual no puede cursar, debido a que no tiene regular su correlativa anterior.	
19. Sistema: muestra un mensaje con la leyenda “la operación se llevo a cabo satisfactoriamente”, y luego se redirecciona a la lista de materias.	Sistema: 1 Muestra un mensaje informando al alumno que no puede llevarse a cabo la inscripción a dicha materia debido a que no está inscripto en la materia correlativa anterior.	



20. Fin del caso de uso.

Observaciones:

3. El alumno puede cancelar la operación en cualquier momento.

Requerimientos no Funcionales Asociados: No Aplica

Reglas de Negocio Asociadas: El alumno debe tener aprobada o regularizada la correlativa anterior.

Fuente: <Minuta, Reunión, Entrevista, Documento, etc>

Referencia Fuente: <Nombre de Minuta, Documento, etc. >

Asociaciones de Inclusión: “Verificar Correlatividad” y “Verificar Regularidad”

Asociaciones de Extensión: No Aplica

Use case donde se incluye: No Aplica

Use Case al que extiende: No Aplica

Use Case de Generalización: No Aplica



REGISTRAR INSCRIPCION A EXAMEN FINAL

NRO. DE ORDEN: 004

PAQUETE: ALUMNO

Actor Principal: Alumno

Actor Secundario: No Aplica

Nivel de Caso de Uso: Negocio Sistema de InformaciónTipo de Caso de Uso: Concreto AbstractoPrioridad: Esencial Útil DeseableCategoría: Esencial Soporte Significativo para la Arquitectura: Si NoComplejidad: Simple Mediano Complejo Muy Complejo Extremadamente Complejo

Objetivo: El alumno debe poder inscribirse para rendir un examen final en la mesa que corresponde

Precondiciones: El alumno debe estar en condición de regular en la materia a la cual desea inscribirse para rendir el final y debe tener aprobada la correlativa anterior.

Post- Condiciones: Éxito 1: El alumno pudo realizar la inscripción al examen final de manera exitosa.

Fracaso 1: el alumno no está en condición de regular en la materia que desea inscribirse.
Fracaso 2: Los casos de uso “Verificar regularidad” y “Verificar Correlatividad” no se ejecutaron de manera exitosa.

Curso Normal	Alternativas
21. El caso de uso comienza cuando el alumno selecciona en el menú la opción Inscripciones, elige la subopción “Examen Final” en la solapa de inscripciones.	
22. El Sistema muestra una pantalla con las materias disponibles para que el alumno pueda realizar la inscripción.	
23. El alumno selecciona la materia a la cual desea inscribirse.	
24. El sistema muestra las fechas disponibles en ese turno.	
25. El alumno escoge la fecha en la cual desea rendir y presiona el botón “Inscribirse”.	5. el alumno presiona el botón “Cancelar”
26. El sistema muestra La inscripción se realizó correctamente y re direcciona a la lista de materias para poder inscribirse en otra si así lo desea.	
27. Fin del caso de uso.	



Observaciones: 4. El alumno puede cancelar la operación en cualquier momento.	
Requerimientos no Funcionales Asociados: No Aplica	
Reglas de Negocio Asociadas: El sistema permite la inscripción con la condición de libre en el caso de los espacios curriculares denominados "Taller"	
Fuente: <Minuta, Reunión, Entrevista, Documento, etc>	Referencia Fuente: <Nombre de Minuta, Documento, etc. >
Asociaciones de Inclusión: No Aplica	
Asociaciones de Extensión: No Aplica	
Use case donde se incluye: Verificar Correlatividad. Verificar Regularidad	
Use Case al que extiende: No Aplica	
Use Case de Generalización: No Aplica	



VERIFICAR CORRELATIVIDAD Y REGULARIDAD

NRO. DE ORDEN: 005

PAQUETE: ALUMNO

Actor Principal: Sistema

Actor Secundario: No Aplica

Nivel de Caso de Uso: Negocio Sistema de InformaciónTipo de Caso de Uso: Concreto AbstractoPrioridad: Esencial Útil DeseableCategoría: Esencial Soporte Significativo para la Arquitectura: Si NoComplejidad: Simple Mediano Complejo Muy Complejo Extremadamente Complejo**Objetivo:** El sistema debe verificar que el alumno antes de inscribirse a una materia por cursar o a un examen final esté en condiciones, esto es materia aprobada o materia regular dependiendo el caso**Precondiciones:** El alumno debe estar inscripto en una carrera.**Post- Condiciones:** **Éxito 1:** Se verifica en que condicion se encuentra el alumno.**Fracaso 1:****Fracaso 2:****Curso Normal****Alternativas**

28. El caso de uso comienza cuando el alumno selecciona la opción “Inscripción a materia por cursar” o “Inscripción a Examen Final”

29. El sistema verifica que el alumno cumpla los requisitos de regularidad para una materia (Asistencia igual o mayor al 80%, parciales aprobados o uno recuperado y aprobado, 80% de trabajos prácticos presentados y aprobados)

30. El sistema informa de su situación al alumno y permite la inscripción a una materia por cursar o inscripción a examen final

31. **Fin del caso de uso.**

3.A El sistema informa al alumno que no se encuentra en condiciones de inscribirse a una materia por cursar o de inscribirse en un examen final.

Observaciones: No Aplica**Requerimientos no Funcionales Asociados:** No Aplica



Reglas de Negocio Asociadas: No Aplica

Fuente: <Minuta, Reunión, Entrevista, Documento, etc>

Referencia Fuente: <Nombre de Minuta, Documento, etc. >

Asociaciones de Inclusión: No Aplica

Asociaciones de Extensión: No Aplica

Use case donde se incluye: Inscripcion a Examen Final y Registrar Inscripcion a Materia por Cursar

Use Case al que extiende: No Aplica

Use Case de Generalización: No Aplica



Plantillas de Trazo Grueso

GENERAR PERMISO DE EXAMEN FINAL		NRO. DE ORDEN: 006	
PAQUETE:ALUMNO			
Actor Principal: Alumno	Actor Secundario: No Aplica		
Nivel de Caso de Uso:	<input type="checkbox"/> Negocio	<input checked="" type="checkbox"/> Sistema de Información	
Tipo de Caso de Uso:	<input checked="" type="checkbox"/> Concreto	<input type="checkbox"/> Abstracto	
Prioridad:	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Útil	<input type="checkbox"/> Deseable
Categoría: <input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Soporte	Significativo para la Arquitectura: <input checked="" type="checkbox"/> Si <input type="checkbox"/> No		
Complejidad: <input type="checkbox"/> Simple <input checked="" type="checkbox"/> Mediano <input type="checkbox"/> Complejo <input type="checkbox"/> Muy Complejo <input type="checkbox"/> Extremadamente Complejo			
Objetivo: El alumno debe generar su permiso de examen final en PDF con las materias a las cuales está inscripto para rendir examen final.			
Precondiciones: Alumno inscripto en al menos 1 materia para examen final.			
Post- Condiciones:	Éxito 1: El alumno genera su permiso de examen final de manera satisfactoria.		
	Fracaso 1: Alumno no inscripto en materias por rendir.		
El caso de uso comienza cuando el alumno selecciona la opción “Consultar inscripción a Examen Final” en la pestaña “Inscripciones”, pudiendo generar el permiso de examen final con las materias en las que se inscribió a rendir.			
Fin del caso de uso.			
Observaciones:			
5. El alumno puede cancelar la operación en cualquier momento.			
Requerimientos no Funcionales Asociados: No Aplica			
Reglas de Negocio Asociadas: No Aplica			
Fuente: Reunión, Entrevista, Documento	Referencia Fuente:		
Asociaciones de Inclusión: No Aplica			



Asociaciones de Extensión: No Aplica

Use case donde se incluye: No Aplica

Use Case al que extiende: No Aplica

Use Case de Generalización: No Aplica



MOSTRAR HISTORIAL ACADÉMICO		NRO. DE ORDEN:007			
		PAQUETE:ALUMNO			
Actor Principal: Alumno	Actor Secundario: No Aplica				
Nivel de Caso de Uso:	<input type="checkbox"/> Negocio	<input checked="" type="checkbox"/> Sistema de Información			
Tipo de Caso de Uso:	<input checked="" type="checkbox"/> Concreto	<input type="checkbox"/> Abtracto			
Prioridad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Útil	<input checked="" type="checkbox"/> Deseable		
Categoría:	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Soporte	Significativo para la Arquitectura: <input checked="" type="checkbox"/> Si <input type="checkbox"/> No		
Complejidad:	<input checked="" type="checkbox"/> Simple	<input type="checkbox"/> Mediano	<input type="checkbox"/> Complejo	<input type="checkbox"/> Muy Complejo	<input type="checkbox"/> Extremadamente Complejo
Objetivo: El alumno debe poder visualizar la totalidad de materias rendidas, aprobadas o no, con sus respectivas calificaciones y el promedio total de materias aprobadas y materias rendidas.					
Precondiciones: Alumno con uno o más finales rendidos.					
Post- Condiciones:	Éxito 1: El alumno puede visualizar su historial académico exitosamente. Fracaso 1: El alumno no ha rendido ningún final.				
El caso de uso comienza cuando el alumno selecciona la opción “Mostrar Historial Académico” en el Menú principal del sistema.					
Fin del caso de uso.					
Observaciones: No aplica					
Requerimientos no Funcionales Asociados: No Aplica					
Reglas de Negocio Asociadas: No Aplica					
Fuente: <Minuta, Reunión, Entrevista, Documento, etc>	Referencia Fuente: <Nombre de Minuta, Documento, etc. >				



PLANTILLAS DE CASO DE USO: ACTOR BEDEL

PLANTILLAS TRAZO FINO

ALTA A MESA DE EXAMEN FINAL		NRO. DE ORDEN: 001			
		PAQUETE: BEDEL			
Actor Principal: Bedel	Actor Secundario: No Aplica				
Nivel de Caso de Uso:	<input type="checkbox"/> Negocio	<input checked="" type="checkbox"/> Sistema de Información			
Tipo de Caso de Uso:	<input checked="" type="checkbox"/> Concreto	<input type="checkbox"/> Abstracto			
Prioridad:	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Útil	<input type="checkbox"/> Deseable		
Categoría:	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Soporte	Significativo para la Arquitectura: <input checked="" type="checkbox"/> Si <input type="checkbox"/> No		
Complejidad:	<input type="checkbox"/> Simple	<input checked="" type="checkbox"/> Mediano	<input type="checkbox"/> Complejo	<input type="checkbox"/> Muy Complejo	<input type="checkbox"/> Extremadamente Complejo
Objetivo: El bedel debe poder registrar las fechas de los exámenes finales de cada materia en los turnos correspondientes.					
Precondiciones: No Aplica					
Post- Condiciones:	Éxito 1: La Mesa de examen final se registro correctamente. Fracaso 1: El Bedel selecciona la opción cancelar.				
Curso Normal		Alternativas			
32. Actor: El caso de uso comienza cuando el bedel selecciona la opción “Aregar mesa de examen final” de la pestaña “Finales”.					
33. Sistema: El sistema muestra un formulario con los campos (Fecha, Hora, Materia, Turno, Cod mesa).					
34. Sistema/Actor: El actor llena los campos correspondientes y selecciona la opción “Guardar”.	3.A El Bedel no completa algún campo del formulario, el sistema muestra un mensaje indicando “campo obligatorio”.				
35. Sistema/Actor: El sistema registra la mesa correctamente y re direcciona al bedel a una página en donde muestra la lista de Mesa de exámenes finales.					



36. Fin del caso de uso.

Observaciones:

6. El Bedel puede cancelar la operación en cualquier momento.

Requerimientos no Funcionales Asociados: No Aplica

Reglas de Negocio Asociadas: No Aplica

Fuente Reunión, Entrevista, Documento

Referencia Fuente: Entrevista.Docx

Asociaciones de Inclusión: No Aplica

Asociaciones de Extensión: No Aplica

Use case donde se incluye: No Aplica

Use Case al que extiende: No Aplica

Use Case de Generalización: No Aplica



BAJA MESA DE EXAMEN FINAL		NRO. DE ORDEN: 002
		PAQUETE: BEDEL
Actor Principal: Bedel		Actor Secundario: No Aplica
Nivel de Caso de Uso:		<input type="checkbox"/> Negocio <input checked="" type="checkbox"/> Sistema de Información
Tipo de Caso de Uso:		<input checked="" type="checkbox"/> Concreto <input type="checkbox"/> Abstracto
Prioridad:		<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Útil <input type="checkbox"/> Deseable
Categoría: <input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Soporte		Significativo para la Arquitectura: <input checked="" type="checkbox"/> Si <input type="checkbox"/> No
Complejidad: <input type="checkbox"/> Simple <input checked="" type="checkbox"/> Mediano <input type="checkbox"/> Complejo <input type="checkbox"/> Muy Complejo <input type="checkbox"/> Extremadamente Complejo		
Objetivo: El bedel debe poder eliminar una mesa de examen final		
Precondiciones: La mesa de Examen final debe estar previamente registrada		
Post- Condiciones:	Éxito 1: La mesa de examen final se elimino correctamente. Fracaso 1: La mesa de examen final que se quiere eliminar no se encuentra registrada en el sistema.	
Curso Normal	Alternativas	
37. Actor: El caso de uso comienza cuando cuando el bedel desea eliminar una mesa de examen final, hace click en la pestaña “Finales”		
38. Sistema: Muestra una lista de opciones: “Agregar Mesa de examen final”, “Lista Mesas de exámenes finales” y “Generar Acta de Examen Final”		
39. Actor: El bedel selecciona la opción “Lista Mesas de exámenes finales”		
40. Sistema: Muestra una lista con las mesas de examene finales		
41. Actor: el bedel hace click en el botón eliminar de la mesa de examen final a la cual desea darse de baja, ubicado a la derecha de cada registro		
42. Sistema: muestra un mensaje: “¿Estas seguro de eliminar inscripción a la MESA DE EXAMEN FINAL?”,en conjunto con los botones “Confirmar” y “No, Cancelar”		



43. Actor: El actor selecciona la opción “Confirmar”	El actor selecciona la opción “No, cancelar”
44. Sistema: Lleva a cabo la operación y re direcciona a la lista de las mesas de exámenes finales, donde se puede ver que es posible volver a eliminar otra mesa de examen final	Sistema: Lleva a cabo la operación y re direcciona al bedel a la pagina principal
45. Fin del caso de uso.	

Observaciones:

7. El Bedel puede cancelar la operación en cualquier momento.

Requerimientos no Funcionales Asociados: No Aplica

Reglas de Negocio Asociadas: No Aplica

Fuente: <Minuta, Reunión, Entrevista, Documento, etc>

Referencia Fuente: <Nombre de Minuta, Documento, etc. >

Asociaciones de Inclusión: No Aplica

Asociaciones de Extensión: No Aplica

Use case donde se incluye: No Aplica

Use Case al que extiende: No Aplica

Use Case de Generalización: No Aplica



CARGAR NOTA DE EXAMEN FINAL ALUMNO		NRO. DE ORDEN: 003
		PAQUETE: BEDEL
Actor Principal: Bedel		Actor Secundario: No Aplica
Nivel de Caso de Uso:		<input type="checkbox"/> Negocio <input checked="" type="checkbox"/> Sistema de Información
Tipo de Caso de Uso:		<input checked="" type="checkbox"/> Concreto <input type="checkbox"/> Abstracto
Prioridad:		<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Útil <input type="checkbox"/> Deseable
Categoría: <input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Soporte		Significativo para la Arquitectura: <input checked="" type="checkbox"/> Si <input type="checkbox"/> No
Complejidad: <input type="checkbox"/> Simple <input checked="" type="checkbox"/> Mediano <input type="checkbox"/> Complejo <input type="checkbox"/> Muy Complejo <input type="checkbox"/> Extremadamente Complejo		
Objetivo: El bedel carga las notas de los exámenes finales de los alumnos de manera satisfactoria.		
Precondiciones: El alumno debe haber rendido el examen final.		
Post- Condiciones:		Éxito 1: El Bedel cargo las notas de los alumnos de manera satisfactoria Fracaso 1: El Bedel selecciona la opción cancelar. Fracaso 2:
Curso Normal		Alternativas
46. Actor: El caso de uso comienza cuando el bedel selecciona la opción “cargar notas de final” de la pestaña “Consultas”		
47. Sistema: El sistema le muestra un formulario con los datos de (Alumno, Nota, Materia, Fecha y Cod_Acta) para que complete adecuadamente		2.A. El Bedel no completa alguno de los campos obligatorios, el sistema le informara con un mensaje que complete ese campo.
48. Actor: El actor selecciona la opción “Guardar” y vuelve a realizar la operación hasta que finaliza de cargar las notas.		
49. Fin del caso de uso.		
Observaciones:		
8. El Bedel puede cancelar la operación en cualquier momento.		
Requerimientos no Funcionales Asociados: No Aplica		



Reglas de Negocio Asociadas: No Aplica

Fuente: Reunión, Entrevista, Documento

Referencia Fuente:

Asociaciones de Inclusión: No Aplica

Asociaciones de Extensión: No Aplica

Use case donde se incluye: No Aplica

Use Case al que extiende: No Aplica

Use Case de Generalización: No Aplica



INSCRIPCIÓN A CARRERA

NRO. DE ORDEN: 004

PAQUETE: BEDEL

Actor Principal: Bedel

Actor Secundario: No Aplica

Nivel de Caso de Uso: Negocio Sistema de InformaciónTipo de Caso de Uso: Concreto AbstractoPrioridad: Esencial Útil DeseableCategoría: Esencial Soporte Significativo para la Arquitectura: Si NoComplejidad: Simple Mediano Complejo Muy Complejo Extremadamente Complejo

Objetivo: Inscribir a un alumno en la carrera que escogió realizar.

Precondiciones: Formulario de Preinscripción realizado. Documentación presentada.

Post- Condiciones:	Éxito 1: El bedel pudo realizar la inscripción del alumno a una Carrera de manera satisfactoria. Fracaso 1: EL alumno no realizó la preinscripción correspondiente. Fracaso 2: El alumno no presentó la documentación obligatoria. Fracaso 3: El bedel opriime el botón cancelar.
--------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Curso Normal	Alternativas
50. Actor: El caso de uso comienza cuando el Bedel selecciona la opción “Lista de Alumnos” y busca en el sistema alumno que desea inscribir.	
51. Sistema: El sistema muestra al Bedel el estado del alumno.	
52. Actor: El bedel comprueba que la información solicitada este completa y completa la inscripción. (a partir de este momento la persona que solicita la inscripción pasa a ser un alumno del ISFD).	3.A La información presentada por el alumno no está completa, el bedel no podrá inscribir al alumno.
53. Fin del caso de uso.	

Observaciones:

9. El Bedel puede cancelar la operación en cualquier momento.



Requerimientos no Funcionales Asociados: No Aplica

Reglas de Negocio Asociadas: No Aplica

Fuente: <Minuta, Reunión, Entrevista, Documento, etc>

Referencia Fuente: <Nombre de Minuta, Documento, etc. >

Asociaciones de Inclusión: No Aplica

Asociaciones de Extensión: No Aplica

Use case donde se incluye: No Aplica

Use Case al que extiende: No Aplica

Use Case de Generalización: No Aplica



MODIFICAR MESA DE EXAMEN FINAL

NRO. DE ORDEN: 005

PAQUETE: BEDEL

Actor Principal: Bedel

Actor Secundario: No Aplica

Nivel de Caso de Uso: Negocio Sistema de InformaciónTipo de Caso de Uso: Concreto AbstractoPrioridad: Esencial Útil DeseableCategoría: Esencial Soporte Significativo para la Arquitectura: Si NoComplejidad: Simple Mediano Complejo Muy Complejo Extremadamente Complejo**Objetivo:** Modificar Datos (Materia, Profesor 1, Profesor 2, Profesor 3, Fecha, Hora, Aula) que intervienen en la registración de la mesa de examen final.**Precondiciones:** La mesa de examen final debe estar registrada en el sistema**Post- Condicioness:** **Éxito 1:** La modificación de La mesa de examen final se realizó correctamente**Fracaso 1:** La mesa de examen final que se quiere modificar no se encuentra registrada en el sistema.**Fracaso 2:** El bedel deja vacío algún campo obligatorio que solicita el sistema.

Curso Normal	Alternativas
1. Actor: El caso de uso comienza cuando el bedel requiere modificar los datos de una mesa de examen final y hace click en la subopción “Lista de Mesas de Examen Final” en la pestaña “Finales”.	
2. Sistema: El sistema muestra al bedel una lista con las mesas de exámenes finales, para que el bedel seleccione cual es la que desea modificar	
3. Actor: El bedel presiona el botón modificar ubicado a la derecha de cada registro	
4. Sistema: Muestra al bedel un formulario con los datos de la mesa de examen final en la que hizo click (Fecha, Hora, Materia, Turno, Cod_Mesa)	
5. Actor: El bedel presiona “Guardar” y el caso de uso termina	3.A: El bedel deja algún dato sin completar. 3.A.1: el sistema informa al bedel y muestra de nuevo el formulario para que lo llene



adecuadamente.

Observaciones:

1. El Bedel puede cancelar la operación en cualquier momento.

Requerimientos no Funcionales Asociados: No Aplica**Reglas de Negocio Asociadas:** No Aplica**Fuente:** Reunión, Entrevista, Documento.**Referencia Fuente:** -----**Asociaciones de Inclusión:** No aplica.**Asociaciones de Extensión:** No Aplica**Use case donde se incluye:** No Aplica**Use Case al que extiende:** No Aplica**Use Case de Generalización:** No Aplica



REGISTRAR INSCRIPCION A MATERIA POR CURSAR

NRO. DE ORDEN: 006

PAQUETE: BEDEL

Actor Principal: Bedel

Actor Secundario: No Aplica

Nivel de Caso de Uso: Negocio Sistema de InformaciónTipo de Caso de Uso: Concreto AbstractoPrioridad: Esencial Útil DeseableCategoría: Esencial Soporte Significativo para la Arquitectura: Si NoComplejidad: Simple Mediano Complejo Muy Complejo Extremadamente Complejo

Objetivo: El Bedel debe poder inscribir a un alumno en una materia a cursar en caso de ser necesario

Precondiciones: El alumno debe estar inscripto en una carrera

Post- Condiciones: Éxito 1: El bedel inscribió al alumno en La materia de manera satisfactoria

Fracaso 1: El Bedel selecciona la opción cancelar.

Fracaso 2: El bedel no completa los campos obligatorios de manera correcta

Curso Normal

Alternativas

6. Actor: El caso de uso comienza cuando el bedel selecciona la opción “Inscripcion a Materia” en la pestaña “Consultas”

2. El bedel completa los campos con datos incorrectos.

2.A. El bedel no completo los datos obligatorios.

8. Actor: El bedel presiona el botón “Guardar”

9. Sistema: Re direcciona al bedel a la pagina principal

10. Fin del caso de uso.

Observaciones:

2. El <Actor> puede cancelar la operación en cualquier momento.



Requerimientos no Funcionales Asociados: No Aplica

Reglas de Negocio Asociadas: No Aplica

Fuente: Reunión, Entrevista, Documento

Referencia Fuente:

Asociaciones de Inclusión: No Aplica

Asociaciones de Extensión: No Aplica

Use case donde se incluye: No Aplica

Use Case al que extiende: No Aplica

Use Case de Generalización: No Aplica



VERIFICAR DOCUMENTACION

NRO. DE ORDEN: 007

PAQUETE: BEDEL

Actor Principal: Bedel

Actor Secundario: No Aplica

Nivel de Caso de Uso: Negocio Sistema de InformaciónTipo de Caso de Uso: Concreto AbstractoPrioridad: Esencial Útil DeseableCategoría: Esencial Soporte Significativo para la Arquitectura: Si NoComplejidad: Simple Mediano Complejo Muy Complejo Extremadamente Complejo**Objetivo:** El bedel verifica que la documentación presentada por el alumno para inscribirse anualmente este completa.**Precondiciones:** El alumno debe estar inscripto en una carrera**Post- Condiciones:****Éxito 1:** El bedel verifico la documentacion presentada por el alumno y llevo a cabo la inscripcion anual**Fracaso 1:** El Bedel selecciona la opción cancelar.**Fracaso 2:** El alumno no presento la documentación completa.**Curso Normal****Alternativas**11. **Actor:** El caso de uso comienza cuando el bedel verifica la documentación presentada por el alumno para su posterior inscripción anual**1.A.** El alumno no presento la documentación completa al Bedel.12. **Sistema:** El sistema muestra al Bedel una casilla a tildar para registrar que el alumno presento toda la documentación solicitada.13. **Actor:** El bedel pulsa el botón aceptar y la operación se lleva a cabo14. **Fin del caso de uso.****Observaciones:**

3. El Bedel puede cancelar la operación en cualquier momento.

Requerimientos no Funcionales Asociados: No Aplica



Reglas de Negocio Asociadas: No Aplica

Fuente: <Minuta, Reunión, Entrevista, Documento, etc>

Referencia Fuente: <Nombre de Minuta, Documento, etc. >

Asociaciones de Inclusión: No Aplica

Asociaciones de Extensión: No Aplica

Use case donde se incluye: Inscripcion a Anual

Use Case al que extiende: No Aplica

Use Case de Generalización: No Aplica



PLANTILLAS TRAZO GRUESO

CONSULTAR ALUMNOS INSCRIPTOS A MESA DE EXAMEN FINAL		NRO. DE ORDEN: 008
		PAQUETE: BEDEL
Actor Principal: Bedel		Actor Secundario: No Aplica
Nivel de Caso de Uso:		<input type="checkbox"/> Negocio <input checked="" type="checkbox"/> Sistema de Información
Tipo de Caso de Uso:		<input checked="" type="checkbox"/> Concreto <input type="checkbox"/> Abstracto
Prioridad:		<input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Útil <input type="checkbox"/> Deseable
Categoría: <input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Soporte		Significativo para la Arquitectura: <input checked="" type="checkbox"/> Si <input type="checkbox"/> No
Complejidad: <input checked="" type="checkbox"/> Simple <input type="checkbox"/> Mediano <input type="checkbox"/> Complejo <input type="checkbox"/> Muy Complejo <input type="checkbox"/> Extremadamente Complejo		
Objetivo: El bedel consulta la cantidad de alumnos inscriptos a una materia en una mesa de examen final		
Precondiciones: No aplica		
Post- Condiciones:	Éxito 1: La consulta sobre La cantidad de alumnos inscriptos a una mesa de examen final fue realizada de manera correcta	
Fracaso 1: El caso de uso comienza cuando el Bedel selecciona la opción “Generar Acta” en la pestaña “Finales” y obtiene la cantidad de Inscriptos a una materia en una mesa de examen final		
Fin del caso de uso.		
Observaciones:		
4. El Bedel puede cancelar la operación en cualquier momento.		
Requerimientos no Funcionales Asociados: No Aplica		
Reglas de Negocio Asociadas: No Aplica		
Fuente: <Minuta, Reunión, Entrevista, Documento, etc>	Referencia Fuente: <Nombre de Minuta, Documento, etc. >	



Asociaciones de Inclusión: No Aplica

Asociaciones de Extensión: No Aplica

Use case donde se incluye: No Aplica

Use Case al que extiende: No Aplica

Use Case de Generalización: No Aplica



GENERAR ACTA VOLANTE		NRO. DE ORDEN: 009
		PAQUETE:BEDEL
Actor Principal: Bedel		Actor Secundario: No Aplica
Nivel de Caso de Uso:		<input type="checkbox"/> Negocio <input checked="" type="checkbox"/> Sistema de Información
Tipo de Caso de Uso:		<input checked="" type="checkbox"/> Concreto <input type="checkbox"/> Abstracto
Prioridad:		<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Útil <input type="checkbox"/> Deseable
Categoría: <input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Soporte		Significativo para la Arquitectura: <input checked="" type="checkbox"/> Si <input type="checkbox"/> No
Complejidad: <input type="checkbox"/> Simple <input checked="" type="checkbox"/> Mediano <input type="checkbox"/> Complejo <input type="checkbox"/> Muy Complejo <input type="checkbox"/> Extremadamente Complejo		
Objetivo: El Bedel debe generar una Acta Volante de examen final en PDF con los alumnos inscriptos a una materias para rendir examen final.		
Precondiciones: No aplica.		
Post- Condiciones:	Éxito 1: El Bedel genera La Acta Volante de examen final de manera satisfactoria.	
Fracaso 1: El caso de uso comienza cuando el Bedel selecciona la opción “Generar Acta de Examen Final” en la pestaña “Finales”, pudiendo ver una lista de materias para generar el acta de examen final de cada una de ellas.		
Fin del caso de uso.		
Observaciones:		
5. El Bedel puede cancelar la operación en cualquier momento.		
Requerimientos no Funcionales Asociados: No Aplica		
Reglas de Negocio Asociadas: No Aplica		
Fuente: <Minuta, Reunión, Entrevista, Documento, etc>	Referencia Fuente: <Nombre de Minuta, Documento, etc. >	
Asociaciones de Inclusión: No Aplica		



Asociaciones de Extensión: No Aplica

Use case donde se incluye: No Aplica

Use Case al que extiende: No Aplica

Use Case de Generalización: No Aplica



PLANTILLAS DE CASO DE USO: ACTOR SUPERUSER

PLANTILLAS TRAZO FINO

ALTA ALUMNO		NRO. DE ORDEN: 001
		PAQUETE:ADMINISTRADOR
Actor Principal: Administrador		Actor Secundario: No Aplica
Nivel de Caso de Uso: <input type="checkbox"/> Negocio <input checked="" type="checkbox"/> Sistema de Información		
Tipo de Caso de Uso: <input checked="" type="checkbox"/> Concreto <input type="checkbox"/> Abstracto		
Prioridad: <input type="checkbox"/> Esencial <input type="checkbox"/> Útil <input type="checkbox"/> Deseable		
Categoría: <input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Soporte		Significativo para la Arquitectura: <input checked="" type="checkbox"/> Si <input type="checkbox"/> No
Complejidad: <input type="checkbox"/> Simple <input checked="" type="checkbox"/> Mediano <input type="checkbox"/> Complejo <input type="checkbox"/> Muy Complejo <input type="checkbox"/> Extremadamente Complejo		
Objetivo: Registrar un nuevo alumno en el sistema		
Precondiciones: No Aplica		
Post- Condiciones:	Éxito 1: El Administrador registra un nuevo usuario en el sistema exitosamente.	
	Fracaso 1: No se ingreso algún dato obligatorio	
	Fracaso 2: El administrador no confirma los datos del usuario a registrar	
Curso Normal		Alternativas
15. Actor: El caso de uso comienza cuando el Administrador selecciona la opción “Users”.		
16. Sistema: Muestra un formulario para que el administrador llene con los datos del alumno a registrar (Username, Email, Nombre y Apellido, DNI, Lugar de Nacimiento, Fecha de Nacimiento, Ciudad Actual, Domicilio Actual, Imagen, Password, Carrera)		
17. Actor: El Administrador llena el formulario y presiona “Guardar”.		3.A: El administrador no ingresa algún campo obligatorio. 3.A.1: El sistema informa de tal situación. 3.A.2: Se muestra el formulario con los datos



	cargados.
18. Sistema: Muestra el detalle de todos los datos ingresados.	
19. Fin del caso de uso.	
Observaciones:	
6. El Administrador puede cancelar la operación en cualquier momento.	
Requerimientos no Funcionales Asociados: No Aplica	
Reglas de Negocio Asociadas: No Aplica	
Fuente: <Minuta, Reunión, Entrevista, Documento, etc>	Referencia Fuente: <Nombre de Minuta, Documento, etc. >
Asociaciones de Inclusión: No Aplica	
Asociaciones de Extensión: No Aplica	
Use case donde se incluye: No Aplica	
Use Case al que extiende: No Aplica	
Use Case de Generalización: No Aplica	



BAJA ALUMNO

NRO. DE ORDEN: 002

PAQUETE:
ADMINISTRADOR

Actor Principal: Administrador

Actor Secundario: No Aplica

Nivel de Caso de Uso: Negocio Sistema de InformaciónTipo de Caso de Uso: Concreto AbstractoPrioridad: Esencial Útil DeseableCategoría: Esencial Soporte Significativo para la Arquitectura: Si NoComplejidad: Simple Mediano Complejo Muy Complejo Extremadamente Complejo

Objetivo: Eliminar un alumno del sistema

Precondiciones: Alumno existente en la base de datos del sistema.

Post- Condiciones: Éxito 1: El Administrador elimino un usuário del sistema correctamente.

Fracaso 1: El alumno que se desea eliminar no se encuentra registrado

Curso Normal	Alternativas
20. Actor: El caso de uso comienza cuando el Administrador presiona en la pestaña “Users”	
21. Sistema: Muestra una lista de alumnos registrados para que el administrador busque cual es el alumno que desea eliminar	
22. Actor: El Administrador selecciona el alumno a eliminar, en la parte superior elije la opción “Eliminar users seleccionado” y luego presiona el botón “Ejecutar”.	
23. Sistema: Muestra un mensaje preguntando si esta seguro de eliminar un alumno.	
24. Actor: El actor presiona la opción “Si, estoy seguro” y el alumno se elimina.	
25. Fin del caso de uso.	

Observaciones:

7. El Administrador puede cancelar la operación en cualquier momento.



Requerimientos no Funcionales Asociados: No Aplica

Reglas de Negocio Asociadas: No Aplica

Fuente: <Minuta, Reunión, Entrevista, Documento, etc>

Referencia Fuente: <Nombre de Minuta, Documento, etc. >

Asociaciones de Inclusión: No Aplica

Asociaciones de Extensión: No Aplica

Use case donde se incluye: No Aplica

Use Case al que extiende: No Aplica

Use Case de Generalización: No Aplica



MODIFICAR ALUMNO

NRO. DE ORDEN: 003

PAQUETE:
ADMINISTRADOR

Actor Principal: Administrador

Actor Secundario: No Aplica

Nivel de Caso de Uso: Negocio Sistema de InformaciónTipo de Caso de Uso: Concreto AbstractoPrioridad: Esencial Útil DeseableCategoría: Esencial Soporte Significativo para la Arquitectura: Si NoComplejidad: Simple Mediano Complejo Muy Complejo Extremadamente Complejo

Objetivo: Modificar los datos del alumno registrado en el sistema, en el caso de error o cambios necesarios.

Precondiciones: Alumno existente en la base de datos del sistema.

Post- Condiciones: **Éxito 1:** El alumno es modificado correctamente del sistema**Fracaso 1:** El alumno no está previamente registrado en el sistema.**Fracaso 2:** El administrador selecciona la opción cancelar.

Curso Normal	Alternativas
26. Actor: El caso de uso comienza cuando el administrador selecciona la pestaña “Users”	
27. Sistema: Muestra una lista de usuarios registrados para que el administrador busque cual es la que desea modificar	
28. Actor: El Administrador selecciona el usuario a modificar y hace click en su nombre	
29. Sistema: Muestra un formulario con los datos del usuario que se hizo click para que el Super Usuario los modifique ((Username, Email, Nombre y Apellido, DNI, Lugar de Nacimiento, Fecha de Nacimiento, Ciudad Actual, Domicilio Actual, Imagen, Password, Carrera))	
30. Actor: El actor presiona la “Guardar”	5.1 El administrador selecciona la opción Cancelar.
31. Fin del caso de uso.	

Observaciones:

8. El Administrador puede cancelar la operación en cualquier momento.



Requerimientos no Funcionales Asociados: No Aplica

Reglas de Negocio Asociadas: No Aplica

Fuente: <Minuta, Reunión, Entrevista, Documento, etc>

Referencia Fuente: <Nombre de Minuta, Documento, etc. >

Asociaciones de Inclusión: No Aplica

Asociaciones de Extensión: No Aplica

Use case donde se incluye: No Aplica

Use Case al que extiende: No Aplica

Use Case de Generalización: No Aplica



ALTA CARRERA

NRO. DE ORDEN: 004

PAQUETE:
ADMINISTRADOR

Actor Principal: Administrador

Actor Secundario: No Aplica

Nivel de Caso de Uso: Negocio Sistema de InformaciónTipo de Caso de Uso: Concreto AbstractoPrioridad: Esencial Útil DeseableCategoría: Esencial Soporte Significativo para la Arquitectura: Si NoComplejidad: Simple Mediano Complejo Muy Complejo Extremadamente Complejo

Objetivo: Registrar una nueva carrera en el sistema

Precondiciones: No Aplica

Post- Condiciones: Éxito 1: El administrador registra exitosamente una nueva carrera en el sistema.

Fracaso 1: El administrador no completa algún campo obligatorio.

Fracaso 2: El administrador no confirma los datos de la carrera a registrar.

Curso Normal	Alternativas
32. Actor: El caso de uso comienza cuando el súper usuario selecciona la opción “Aregar Carrera” en la pestaña “Carreras”.	
33. Sistema: Muestra un formulario con los datos necesarios para dar de alta una nueva carrera. Estos datos son: Cod_Carrera, Nombre, Resolución, Duración, Alumnos	
34. Actor: El súper usuario llena el formulario y presiona “Guardar”.	3 .A: El súper usuario no ingresa algún campo obligatorio. 3.A.1: El sistema informa de tal situación. 3.A.2: Se muestra el formulario con los datos cargados.
35. Sistema: Muestra el detalle de todos los datos ingresados, en el caso de necesitar alguna modificación antes de registrar los datos de la nueva carrera en el sistema.	

**36. Fin del caso de uso.****Observaciones:**

9. El administrador puede cancelar la operación en cualquier momento.

Requerimientos no Funcionales Asociados: No Aplica

Reglas de Negocio Asociadas: No Aplica

Fuente: <Minuta, Reunión, Entrevista, Documento, etc>

Referencia Fuente: <Nombre de Minuta, Documento, etc. >

Asociaciones de Inclusión: No Aplica

Asociaciones de Extensión: No Aplica

Use case donde se incluye: No Aplica

Use Case al que extiende: No Aplica

Use Case de Generalización: No Aplica



BAJA CARRERA		NRO. DE ORDEN: 005
		PAQUETE: ADMINISTRADOR
Actor Principal: Administrador		Actor Secundario: No Aplica
Nivel de Caso de Uso:		<input type="checkbox"/> Negocio <input checked="" type="checkbox"/> Sistema de Información
Tipo de Caso de Uso:		<input checked="" type="checkbox"/> Concreto <input type="checkbox"/> Abstracto
Prioridad:		<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Útil <input type="checkbox"/> Deseable
Categoría: <input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Soporte		Significativo para la Arquitectura: <input checked="" type="checkbox"/> Si <input type="checkbox"/> No
Complejidad: <input type="checkbox"/> Simple <input checked="" type="checkbox"/> Mediano <input type="checkbox"/> Complejo <input type="checkbox"/> Muy Complejo <input type="checkbox"/> Extremadamente Complejo		
Objetivo: Eliminar una carrera del sistema.		
Precondiciones: Carrera registrada en la base de datos del sistema.		
Post- Condiciones:	Éxito 1: El administrador elimino exitosamente la Carrera del sistema. Fracaso 1: El administrador selecciona la opción cancelar. Fracaso 2: El profesor no se encuentra registrado en el sistema.	
Curso Normal		Alternativas
37. Actor: El caso de uso comienza cuando el administrador selecciona la pestaña “Carreras”		
38. Sistema: Muestra una lista de carreras registradas para que el administrador busque cual es la que desea eliminar		
39. Actor: El Administrador selecciona la carrera a eliminar, en la parte superior elige la opción “Eliminar carrera seleccionada” y luego presiona el botón “Ejecutar”.		.
40. Sistema: Muestra un mensaje preguntando si esta seguro de eliminar la carrera.		
41. Actor: El actor presiona la opción “Si, estoy seguro” y la carrera se elimina.		
42. Fin del caso de uso.		

**Observaciones:**

10. El administrador puede cancelar la operación en cualquier momento.

Requerimientos no Funcionales Asociados: No Aplica

Reglas de Negocio Asociadas: No Aplica

Fuente: <Minuta, Reunión, Entrevista, Documento, etc>

Referencia Fuente: <Nombre de Minuta, Documento, etc. >

Asociaciones de Inclusión: No Aplica

Asociaciones de Extensión: No Aplica

Use case donde se incluye: No Aplica

Use Case al que extiende: No Aplica

Use Case de Generalización: No Aplica



MODIFICAR CARRERA

NRO. DE ORDEN: 006

PAQUETE:
ADMINISTRADOR

Actor Principal: Administrador

Actor Secundario: No Aplica

Nivel de Caso de Uso: Negocio Sistema de InformaciónTipo de Caso de Uso: Concreto AbstractoPrioridad: Esencial Útil DeseableCategoría: Esencial Soporte Significativo para la Arquitectura: Si NoComplejidad: Simple Mediano Complejo Muy Complejo Extremadamente Complejo

Objetivo:

Precondiciones: No Aplica

Post- Condiciones:

Éxito 1:

Fracaso 1: El <Actor> selecciona la opción cancelar.

Fracaso 2:

Curso Normal	Alternativas
43. Actor: El caso de uso comienza cuando el administrador selecciona la pestaña “Carreras”	
44. Sistema: Muestra una lista de carreras registrados para que el administrador busque cual es la que desea modificar	
45. Actor: El Administrador selecciona la carrera a modificar y hace click en su nombre	
46. Sistema: Muestra un formulario con los datos del usuario que se hizo click para que el Super Usuario los modifique Cod_Carrera, Nombre, Resolución, Duración, Alumnos	
47. Actor: El actor presiona la “Guardar”	
48. Fin del caso de uso.	

Observaciones:

11. El super usuario puede cancelar la operación en cualquier momento.

Requerimientos no Funcionales Asociados: No Aplica



Reglas de Negocio Asociadas: No Aplica

Fuente: Reunión, Entrevista, Documento

Referencia Fuente:

Asociaciones de Inclusión: No Aplica

Asociaciones de Extensión: No Aplica

Use case donde se incluye: No Aplica

Use Case al que extiende: No Aplica

Use Case de Generalización: No Aplica



ALTA MATERIA		NRO. DE ORDEN: 007
		PAQUETE: ADMINISTRADOR
Actor Principal: Administrador		Actor Secundario: No Aplica
Nivel de Caso de Uso:		<input type="checkbox"/> Negocio <input checked="" type="checkbox"/> Sistema de Información
Tipo de Caso de Uso:		<input checked="" type="checkbox"/> Concreto <input type="checkbox"/> Abstracto
Prioridad:		<input type="checkbox"/> Esencial <input type="checkbox"/> Útil <input type="checkbox"/> Deseable
Categoría: <input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Soporte		Significativo para la Arquitectura: <input checked="" type="checkbox"/> Si <input type="checkbox"/> No
Complejidad: <input type="checkbox"/> Simple <input checked="" type="checkbox"/> Mediano <input type="checkbox"/> Complejo <input type="checkbox"/> Muy Complejo <input type="checkbox"/> Extremadamente Complejo		
Objetivo: Registrar una nueva materia en el sistema		
Precondiciones: No Aplica		
Post- Condiciones:	Éxito 1: El Administrador registra una nueva matrícula (espacio curricular) en el sistema exitosamente. Fracaso 1: No se ingreso algún dato obligatorio Fracaso 2: El administrador no confirma los datos de la materia a registrar	
Curso Normal		Alternativas
49. Actor: El caso de uso comienza cuando el super usuario selecciona la opción “Aregar Materia” en la pestaña “Materias”.		
50. Sistema: Muestra un formulario para que el super usuario llene con los datos del alumno a registrar (Cod_Materia, Nombre, Carrera, Duración, Año a que pertenece, Correlativas)		
51. Actor: El super usuario llena el formulario y presiona “Guardar”.		3.A: El administrador no ingresa algún campo obligatorio. 3.A.1: El sistema informa de tal situación. 3.A.2: Se muestra el formulario con los datos cargados.
52. Sistema: Muestra el detalle de todos los datos ingresados.		

**53. Fin del caso de uso.****Observaciones:**

12. El Administrador puede cancelar la operación en cualquier momento.

Requerimientos no Funcionales Asociados: No Aplica

Reglas de Negocio Asociadas: No Aplica

Fuente: <Minuta, Reunión, Entrevista, Documento, etc>

Referencia Fuente: <Nombre de Minuta, Documento, etc. >

Asociaciones de Inclusión: No Aplica

Asociaciones de Extensión: No Aplica

Use case donde se incluye: No Aplica

Use Case al que extiende: No Aplica

Use Case de Generalización: No Aplica



BAJA MATERIA		NRO. DE ORDEN: 008			
		PAQUETE: ADMINISTRADOR			
Actor Principal: Administrador	Actor Secundario: No Aplica				
Nivel de Caso de Uso:	<input type="checkbox"/> Negocio	<input checked="" type="checkbox"/> Sistema de Información			
Tipo de Caso de Uso:	<input checked="" type="checkbox"/> Concreto	<input type="checkbox"/> Abstracto			
Prioridad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Útil	<input type="checkbox"/> Deseable		
Categoría:	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Soporte	Significativo para la Arquitectura: <input checked="" type="checkbox"/> Si <input type="checkbox"/> No		
Complejidad:	<input type="checkbox"/> Simple	<input checked="" type="checkbox"/> Mediano	<input type="checkbox"/> Complejo	<input type="checkbox"/> Muy Complejo	<input type="checkbox"/> Extremadamente Complejo
Objetivo: Eliminar una materia del sistema					
Precondiciones: La materia debe existir en la base de datos del sistema.					
Post- Condiciones:	Éxito 1: El Administrador elimino una matéría del sistema correctamente. Fracaso 1: La materia que se desea eliminar no se encuentra registrado				
Curso Normal		Alternativas			
54. Actor: El caso de uso comienza cuando el administrador selecciona la pestaña “Materias”					
55. Sistema: Muestra una lista de materias registradas para que el administrador busque cual es la que desea eliminar					
56. Actor: El Administrador selecciona la materia a eliminar, en la parte superior elije la opción “Eliminar materia seleccionada” y luego presiona el botón “Ejecutar”.					
57. Sistema: Muestra un mensaje preguntando si esta seguro de eliminar la Materia.					
58. Actor: El actor presiona la opción “Si” y la materia se elimina.					
59. Fin del caso de uso.					
Observaciones:					
13. El Administrador puede cancelar la operación en cualquier momento.					
Requerimientos no Funcionales Asociados: No Aplica					



Reglas de Negocio Asociadas: No Aplica

Fuente: <Minuta, Reunión, Entrevista, Documento, etc>

Referencia Fuente: <Nombre de Minuta, Documento, etc. >

Asociaciones de Inclusión: No Aplica

Asociaciones de Extensión: No Aplica

Use case donde se incluye: No Aplica

Use Case al que extiende: No Aplica

Use Case de Generalización: No Aplica



MODIFICAR MATERIA

NRO. DE ORDEN: 009

PAQUETE:
ADMINISTRADOR

Actor Principal: Administrador

Actor Secundario: No Aplica

Nivel de Caso de Uso: Negocio Sistema de InformaciónTipo de Caso de Uso: Concreto AbstractoPrioridad: Esencial Útil DeseableCategoría: Esencial Soporte Significativo para la Arquitectura: Si NoComplejidad: Simple Mediano Complejo Muy Complejo Extremadamente Complejo

Objetivo: Modificar los datos de una materia registrada en el sistema, en el caso de error o cambios necesarios.

Precondiciones: La materia debe existir en la base de datos del sistema.

Post- Condiciones: Éxito 1: La materia es modificada correctamente del sistema

Fracaso 1: La materia no está previamente registrada en el sistema.

Fracaso 2: El administrador selecciona la opción cancelar.

Curso Normal	Alternativas
60. Actor: El caso de uso comienza cuando el administrador selecciona la pestaña “Materias”	
61. Sistema: Muestra una lista de materias registradas para que el super usuario busque cual es la que desea modificar	
62. Actor: El super usuario selecciona la materia a modificar y hace click en su nombre	
63. Sistema: Muestra un formulario con los datos del usuario que se hizo click para que el Super Usuario los modifique (Cod_Materia, Nombre, Carrera, Duración, Año a que pertenece, Correlativas)	
64. Actor: El actor presiona la “Guardar”	
65. Fin del caso de uso.	

**Observaciones:**

14. El Super usuario puede cancelar la operación en cualquier momento.

Requerimientos no Funcionales Asociados: No Aplica

Reglas de Negocio Asociadas: No Aplica

Fuente: <Minuta, Reunión, Entrevista, Documento, etc>

Referencia Fuente: <Nombre de Minuta, Documento, etc. >

Asociaciones de Inclusión: No Aplica

Asociaciones de Extensión: No Aplica

Use case donde se incluye: No Aplica

Use Case al que extiende: No Aplica

Use Case de Generalización: No Aplica



ALTA MESA DE EXAMEN		NRO. DE ORDEN: 010			
		PAQUETE: ADMINISTRADOR			
Actor Principal: Administrador	Actor Secundario: No Aplica				
Nivel de Caso de Uso:	<input type="checkbox"/> Negocio	<input checked="" type="checkbox"/> Sistema de Información			
Tipo de Caso de Uso:	<input checked="" type="checkbox"/> Concreto	<input type="checkbox"/> Abstracto			
Prioridad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Útil	<input type="checkbox"/> Deseable		
Categoría:	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Soporte	Significativo para la Arquitectura: <input checked="" type="checkbox"/> Si <input type="checkbox"/> No		
Complejidad:	<input type="checkbox"/> Simple	<input checked="" type="checkbox"/> Mediano	<input type="checkbox"/> Complejo	<input type="checkbox"/> Muy Complejo	<input type="checkbox"/> Extremadamente Complejo
Objetivo: Registrar una nueva mesa de examen en el sistema					
Precondiciones: No Aplica					
Post- Condiciones:	Éxito 1: El Administrador registra una nueva mesa de examen en el sistema exitosamente. Fracaso 1: No se ingreso algún dato obligatorio Fracaso 2: El administrador no confirma los datos del usuario a registrar				
Curso Normal		Alternativas			
66. Actor: El caso de uso comienza cuando el Administrador selecciona la opción “Aregar Mesa de Examen” en la pestaña “Mesa Finales”.					
67. Sistema: Muestra un formulario para que el administrador llene con los datos del alumno a registrar (Fecha, Hora, Materia, Turno, Cod_Mesa)					
68. Actor: El Administrador llena el formulario y presiona “Guardar”.	3.A: El administrador no ingresa algún campo obligatorio. 3.A.1: El sistema informa de tal situación. 3.A.2: Se muestra el formulario con los datos cargados.				
69. Sistema: Muestra el detalle de todos los datos ingresados.					



70. Fin del caso de uso.

Observaciones:

15. El Administrador puede cancelar la operación en cualquier momento.

Requerimientos no Funcionales Asociados: No Aplica

Reglas de Negocio Asociadas: No Aplica

Fuente: <Minuta, Reunión, Entrevista, Documento, etc>

Referencia Fuente: <Nombre de Minuta, Documento, etc. >

Asociaciones de Inclusión: No Aplica

Asociaciones de Extensión: No Aplica

Use case donde se incluye: No Aplica

Use Case al que extiende: No Aplica

Use Case de Generalización: No Aplica



BAJA MESA DE EXAMEN

NRO. DE ORDEN: 011

PAQUETE:
ADMINISTRADOR

Actor Principal: Administrador

Actor Secundario: No Aplica

Nivel de Caso de Uso: Negocio Sistema de InformaciónTipo de Caso de Uso: Concreto AbstractoPrioridad: Esencial Útil DeseableCategoría: Esencial Soporte Significativo para la Arquitectura: Si NoComplejidad: Simple Mediano Complejo Muy Complejo Extremadamente Complejo

Objetivo: Eliminar una mesa de examen del sistema.

Precondiciones: Mesa de Examen registrada en la base de datos del sistema.

Post- Condiciones: Éxito 1: El administrador elimino exitosamente la mesa de examen del sistema.

Fracaso 1: El administrador selecciona la opción cancelar.

Fracaso 2: La mesa de examen no se encuentra registrado en el sistema.

Curso Normal	Alternativas
71. Actor: El caso de uso comienza cuando el administrador selecciona la pestaña “Mesa Finales”	
72. Sistema: Muestra una lista de mesas de exámenes finales registradas para que el administrador busque cual es la que desea eliminar	
73. Actor: El Administrador selecciona la mesa de examen final a eliminar, en la parte superior elije la opción “Eliminar materia seleccionada” y luego presiona el botón “Ejecutar”.	
74. Sistema: Muestra un mensaje preguntando si esta seguro de eliminar la mesa de examen final.	
75. Actor: El actor presiona la opción “Si” y la mesa de examen final se elimina.	
76. Fin del caso de uso.	

Observaciones:

16. El administrador puede cancelar la operación en cualquier momento.



Requerimientos no Funcionales Asociados: No Aplica

Reglas de Negocio Asociadas: No Aplica

Fuente: <Minuta, Reunión, Entrevista, Documento, etc>

Referencia Fuente: <Nombre de Minuta, Documento, etc. >

Asociaciones de Inclusión: No Aplica

Asociaciones de Extensión: No Aplica

Use case donde se incluye: No Aplica

Use Case al que extiende: No Aplica

Use Case de Generalización: No Aplica



MODIFICAR MESA DE EXAMEN

NRO. DE ORDEN: 012

PAQUETE:
ADMINISTRADOR

Actor Principal: Administrador

Actor Secundario: No Aplica

Nivel de Caso de Uso: Negocio Sistema de InformaciónTipo de Caso de Uso: Concreto AbstractoPrioridad: Esencial Útil DeseableCategoría: Esencial Soporte Significativo para la Arquitectura: Si NoComplejidad: Simple Mediano Complejo Muy Complejo Extremadamente Complejo**Objetivo:** Modificar los datos de una mesa de examen registrada en el sistema, en el caso de error o cambios necesarios.**Precondiciones:** Mesa de Examen existente en la base de datos del sistema.**Post- Condiciones:** **Éxito 1:** La Mesa de Examen es modificado correctamente del sistema**Fracaso 1:** La Mesa de Examen no está previamente registrado en el sistema.**Fracaso 2:** El administrador selecciona la opción cancelar.

Curso Normal	Alternativas
77. Actor: El caso de uso comienza cuando el administrador selecciona la pestaña “Mesa Finales”	
78. Sistema: Muestra una lista de mesas de exámenes finales para que el super usuario busque cual es la que desea modificar	
79. Actor: El super usuario selecciona la mesa de examen final a modificar y hace click en su nombre	
80. Sistema: Muestra un formulario con los datos del usuario que se hizo click para que el Super Usuario los modifique (Fecha, Hora, Materia, Turno, Cod_Mesa)	
81. Actor: El actor presiona la “Guardar”	
82. Fin del caso de uso.	

Observaciones:

17. El Administrador puede cancelar la operación en cualquier momento.



Requerimientos no Funcionales Asociados: No Aplica

Reglas de Negocio Asociadas: No Aplica

Fuente: Reunión, Entrevista, Documento

Referencia Fuente:

Asociaciones de Inclusión: No Aplica

Asociaciones de Extensión: No Aplica

Use case donde se incluye: No Aplica

Use Case al que extiende: No Aplica

Use Case de Generalización: No Aplica



Plantillas Trazo Grueso

CONSULTAR ALUMNOS		NRO. DE ORDEN: 013	
		PAQUETE: ADMINISTRADOR	
Actor Principal: Administrador	Actor Secundario: No Aplica		
Nivel de Caso de Uso:	<input type="checkbox"/> Negocio	<input checked="" type="checkbox"/> Sistema de Información	
Tipo de Caso de Uso:	<input checked="" type="checkbox"/> Concreto	<input type="checkbox"/> Abstracto	
Prioridad:	<input type="checkbox"/> Esencial	<input checked="" type="checkbox"/> Útil	<input type="checkbox"/> Deseable
Categoría: <input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Soporte	Significativo para la Arquitectura: <input checked="" type="checkbox"/> Si <input type="checkbox"/> No		
Complejidad: <input checked="" type="checkbox"/> Simple <input type="checkbox"/> Mediano <input type="checkbox"/> Complejo <input type="checkbox"/> Muy Complejo <input type="checkbox"/> Extremadamente Complejo			
Objetivo: El Super Usuario consulta la cantidad de alumnos registrados en el sistema para realizar algún cambio en caso de ser necesario			
Precondiciones: Debe haber alumnos registrados en el sistema			
Post- Condicioness:	Éxito 1: La consulta sobre los alumnos registrados en El sistemal fue realizada de manera correcta Fracaso 1: No existe ningún alumno registrado en el sistema		
El caso de uso comienza cuando el Super Usuario selecciona la opción “Lista de Alumnos” en la pestaña “Alumnos”.			
Fin del caso de uso.			
Observaciones:			
18. El Super Usuario puede cancelar la operación en cualquier momento.			
Requerimientos no Funcionales Asociados: No Aplica			
Reglas de Negocio Asociadas: No Aplica			
Fuente: <Minuta, Reunión, Entrevista, Documento, etc>	Referencia Fuente: <Nombre de Minuta, Documento, etc. >		



Asociaciones de Inclusión: No Aplica

Asociaciones de Extensión: No Aplica

Use case donde se incluye: No Aplica

Use Case al que extiende: No Aplica

Use Case de Generalización: No Aplica



CONSULTAR CARRERA		NRO. DE ORDEN: 014
		PAQUETE: ADMINISTRADOR
Actor Principal: Administrador		Actor Secundario: No Aplica
Nivel de Caso de Uso: <input type="checkbox"/> Negocio <input checked="" type="checkbox"/> Sistema de Información		
Tipo de Caso de Uso: <input checked="" type="checkbox"/> Concreto <input type="checkbox"/> Abstracto		
Prioridad: <input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Útil <input type="checkbox"/> Deseable		
Categoría: <input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Soporte		Significativo para la Arquitectura: <input checked="" type="checkbox"/> Si <input type="checkbox"/> No
Complejidad: <input checked="" type="checkbox"/> Simple <input type="checkbox"/> Mediano <input type="checkbox"/> Complejo <input type="checkbox"/> Muy Complejo <input type="checkbox"/> Extremadamente Complejo		
Objetivo: El Bedel consulta una Carrera registrada en el sistema para realizar algún cambio en caso de ser necesario		
Precondiciones: Debe haber Carreras registrados en el sistema		
Post- Condiciones:	Éxito 1: La consulta sobre La Carrera registrada en el sistema fue realizada de manera correcta Fracaso 1: No existe ninguna carrera registrada en el sistema	
El caso de uso comienza cuando el Bedel selecciona la opción “Lista Carreras” en la pestaña “Carreras”, pudiendo así consultar las diferentes carreras registradas.		
Fin del caso de uso.		
Observaciones:		
19. El Bedel puede cancelar la operación en cualquier momento.		
Requerimientos no Funcionales Asociados: No Aplica		
Reglas de Negocio Asociadas: No Aplica		
Fuente: Reunión, Entrevista, Documento	Referencia Fuente:	
Asociaciones de Inclusión: No Aplica		



Asociaciones de Extensión: No Aplica

Use case donde se incluye: No Aplica

Use Case al que extiende: No Aplica

Use Case de Generalización: No Aplica



CONSULTAR INSCRIPCIONES

NRO. DE ORDEN: 015

PAQUETE:
ADMINISTRADOR**Actor Principal:** Administrador**Actor Secundario:** No Aplica**Nivel de Caso de Uso:** Negocio Sistema de Información**Tipo de Caso de Uso:** Concreto Abstracto**Prioridad:** Esencial Útil Deseable**Categoría:** Esencial Soporte
Significativo para la Arquitectura: Si No**Complejidad:** Simple Mediano Complejo Muy Complejo Extremadamente Complejo**Objetivo:** El alumno debe poder consultar las materias en las que ya está inscripto, su fecha y el aula en donde se rendirá el examen final.**Precondiciones:** El alumno debe haberse inscripto previamente en alguna mesa de examen final.**Post- Condiciones:** **Éxito 1:** El alumno visualiza las materias a las cuales está inscripto de manera satisfactoria.**Fracaso 1:** El alumno no está inscripto en ninguna materia por rendir.

El caso de uso comienza cuando el alumno selecciona la opción "Consultar inscripción a materias por cursar" en la pestaña "Consultar Inscripciones".

Fin del caso de uso.**Observaciones:** No aplica.**Requerimientos no Funcionales Asociados:** No Aplica**Reglas de Negocio Asociadas:** No Aplica**Fuente:** <Minuta, Reunión, Entrevista, Documento, etc>**Referencia Fuente:** <Nombre de Minuta, Documento, etc. >**Asociaciones de Inclusión:** No Aplica



ENTORNO DE GESTION ACADEMICA

FECHA: 26/08/14

DOCUMENTO: ESPECIFICACION DE REQUERIMIENTOS DE SOFTWARE

Asociaciones de Extensión: No Aplica

Use case donde se incluye: No Aplica

Use Case al que extiende: No Aplica

Use Case de Generalización: No Aplica



CONSULTAR MATERIA		NRO. DE ORDEN: 016
		PAQUETE: ADMINISTRADOR
Actor Principal: Administrador		Actor Secundario: No Aplica
Nivel de Caso de Uso: <input type="checkbox"/> Negocio <input checked="" type="checkbox"/> Sistema de Información		
Tipo de Caso de Uso: <input checked="" type="checkbox"/> Concreto <input type="checkbox"/> Abstracto		
Prioridad: <input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Útil <input type="checkbox"/> Deseable		
Categoría: <input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Soporte		Significativo para la Arquitectura: <input checked="" type="checkbox"/> Si <input type="checkbox"/> No
Complejidad: <input checked="" type="checkbox"/> Simple <input type="checkbox"/> Mediano <input type="checkbox"/> Complejo <input type="checkbox"/> Muy Complejo <input type="checkbox"/> Extremadamente Complejo		
Objetivo: El Bedel consulta una Materia registrada en el sistema para realizar algún cambio en caso de ser necesario		
Precondiciones: Debe haber Carreras registrados en el sistema, las cuales a su vez tengan materias registradas.		
Post- Condiciones:	Éxito 1: La consulta sobre La materia registrada en el sistema fue realizada de manera correcta Fracaso 1: No existe ninguna materia registrada en el sistema	
El caso de uso comienza cuando el Bedel selecciona la opción “Lista Materias” en la pestaña “Materias”.		
Fin del caso de uso.		
Observaciones:		
20. El Super Usuario puede cancelar la operación en cualquier momento.		
Requerimientos no Funcionales Asociados: No Aplica		
Reglas de Negocio Asociadas: No Aplica		
Fuente: <Minuta, Reunión, Entrevista, Documento, etc>		Referencia Fuente: <Nombre de Minuta, Documento, etc. >



Asociaciones de Inclusión: No Aplica

Asociaciones de Extensión: No Aplica

Use case donde se incluye: No Aplica

Use Case al que extiende: No Aplica

Use Case de Generalización: No Aplica



CONSULTAR MESA DE EXAMEN FINAL		NRO. DE ORDEN: 017	
		PAQUETE: ADMINISTRADOR	
Actor Principal: Administrador	Actor Secundario: No Aplica		
Nivel de Caso de Uso:	<input type="checkbox"/> Negocio	<input checked="" type="checkbox"/> Sistema de Información	
Tipo de Caso de Uso:	<input checked="" type="checkbox"/> Concreto	<input type="checkbox"/> Abstracto	
Prioridad:	<input type="checkbox"/> Esencial	<input checked="" type="checkbox"/> Útil	<input type="checkbox"/> Deseable
Categoría: <input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Soporte	Significativo para la Arquitectura: <input checked="" type="checkbox"/> Si <input type="checkbox"/> No		
Complejidad: <input checked="" type="checkbox"/> Simple <input type="checkbox"/> Mediano <input type="checkbox"/> Complejo <input type="checkbox"/> Muy Complejo <input type="checkbox"/> Extremadamente Complejo			
Objetivo: El bedel consulta la fecha de una determinada mesa de examen final			
Precondiciones: La mesa de examen final debe estar registrada			
Post- Condiciones:	Éxito 1: La consulta sobre La mesa se examen final fue realizada de manera correcta		
	Fracaso 1: No se encuentra la mesa de examen final debido a que no esta registrada en el sistema		
El caso de uso comienza cuando el Bedel selecciona la opción “Lista de mesas de exámenes finales” en la pestaña “Finales”, pudiendo ver todas las mesas de exámenes finales registradas en el sistema			
Fin del caso de uso.			
Observaciones:			
21. El <Actor> puede cancelar la operación en cualquier momento.			
Requerimientos no Funcionales Asociados: No Aplica			
Reglas de Negocio Asociadas: No Aplica			
Fuente: Reunión, Entrevista, Documento,	Referencia Fuente:		
Asociaciones de Inclusión: No Aplica			



Asociaciones de Extensión: No Aplica

Use case donde se incluye: No Aplica

Use Case al que extiende: No Aplica

Use Case de Generalización: No Aplica



ENTORNO DE GESTION ACADEMICA

DIAGRAMAS DE ACTIVIDAD

VERSION 1.0



DIAGRAMA DE ACTIVIDAD: "PREINSCRIPCION"

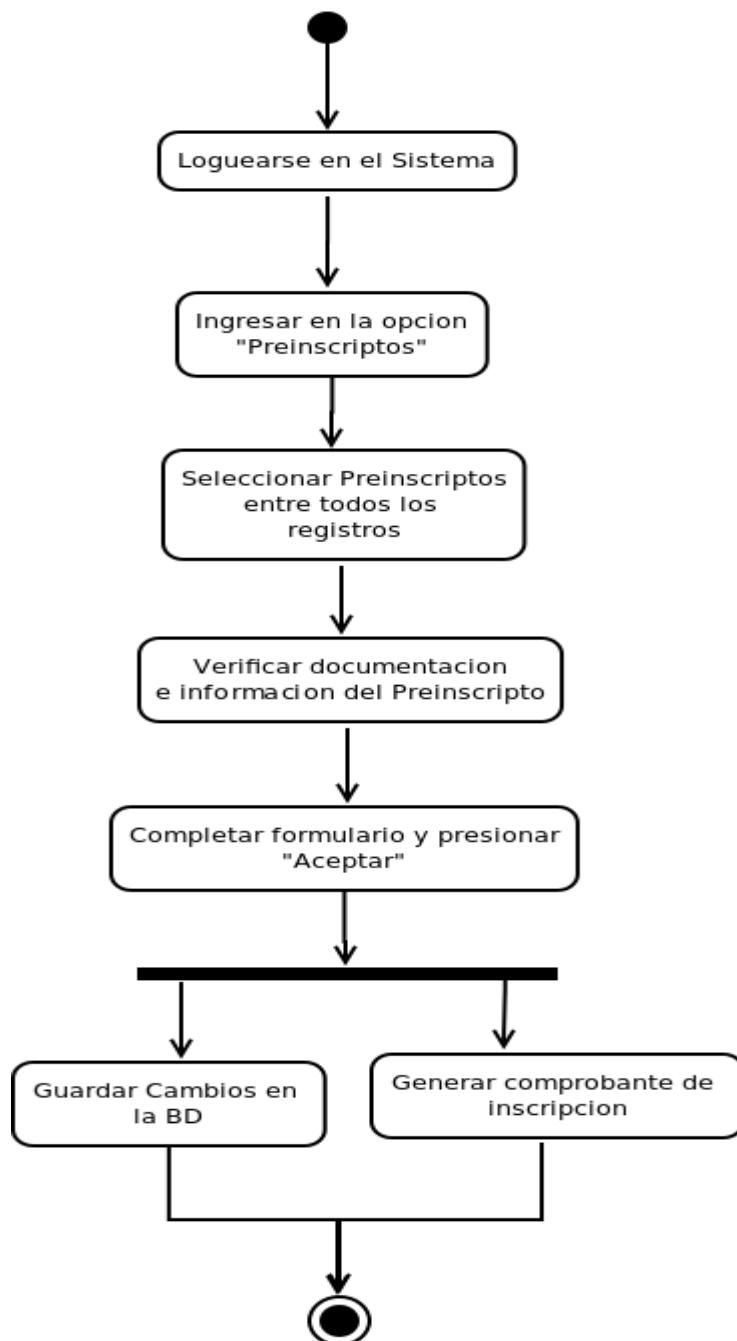




DIAGRAMA DE ACTIVIDAD: "AGREGAR ALUMNO"

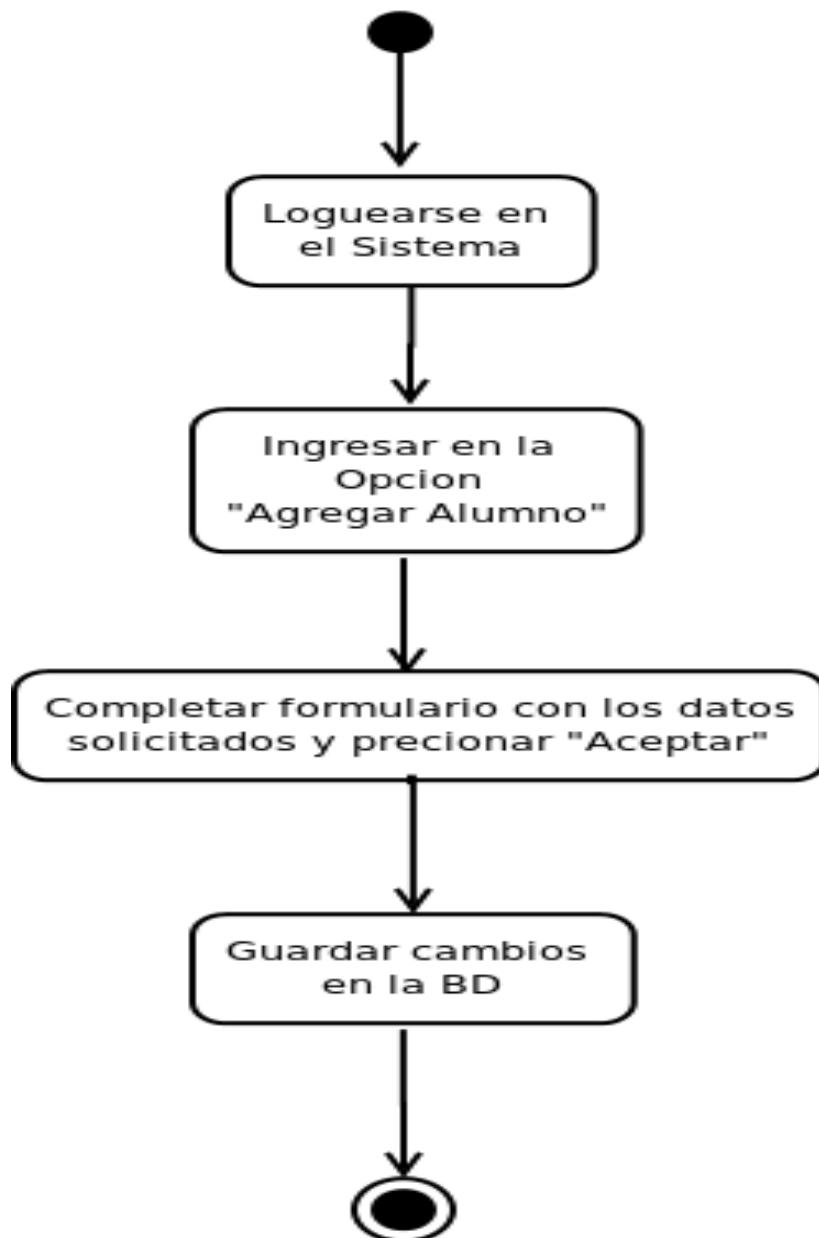




DIAGRAMA DE ACTIVIDAD: "ELIMINAR ALUMNO"





DIAGRAMA DE ACTIVIDAD: "AGREGAR PROFESOR"

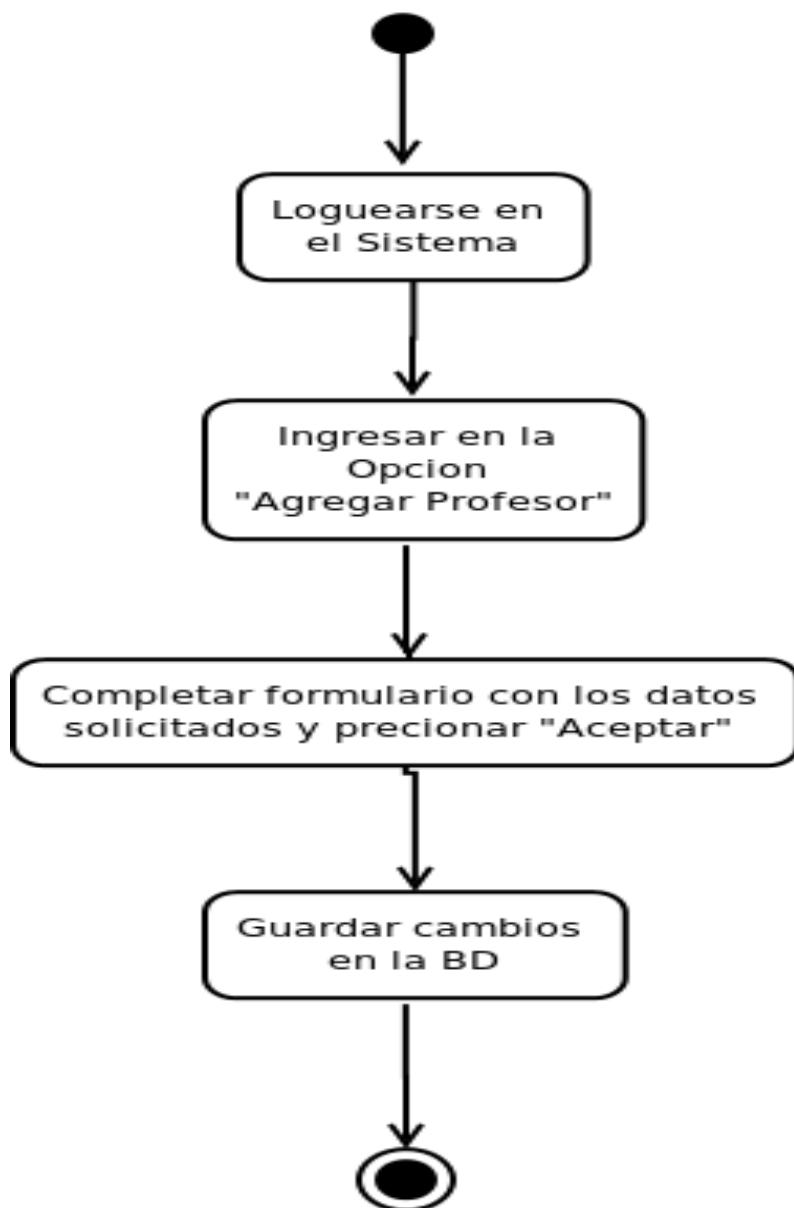




DIAGRAMA DE ACTIVIDAD: "ELIMINAR PROFESOR"

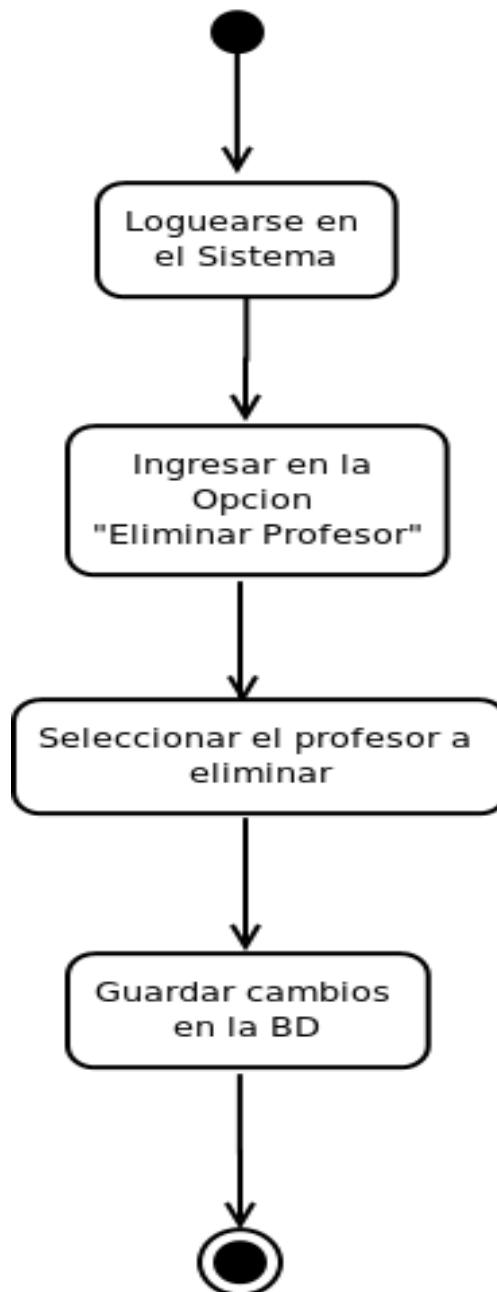




DIAGRAMA DE ACTIVIDAD: "INSCRIPCION A EXAMEN FINAL"

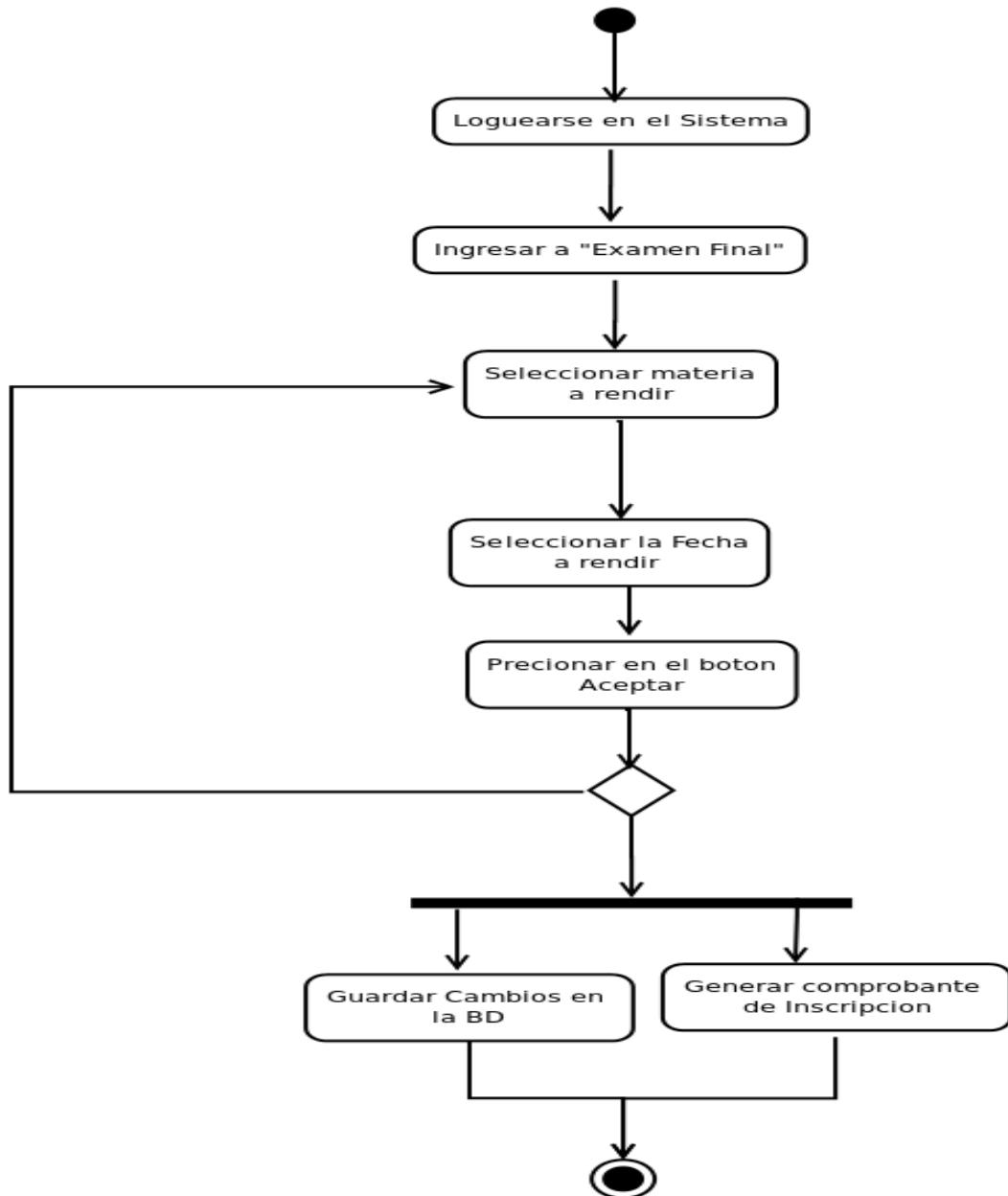




DIAGRAMA DE ACTIVIDAD: "INSCRIPCION A MATERIA POR CURSAR"

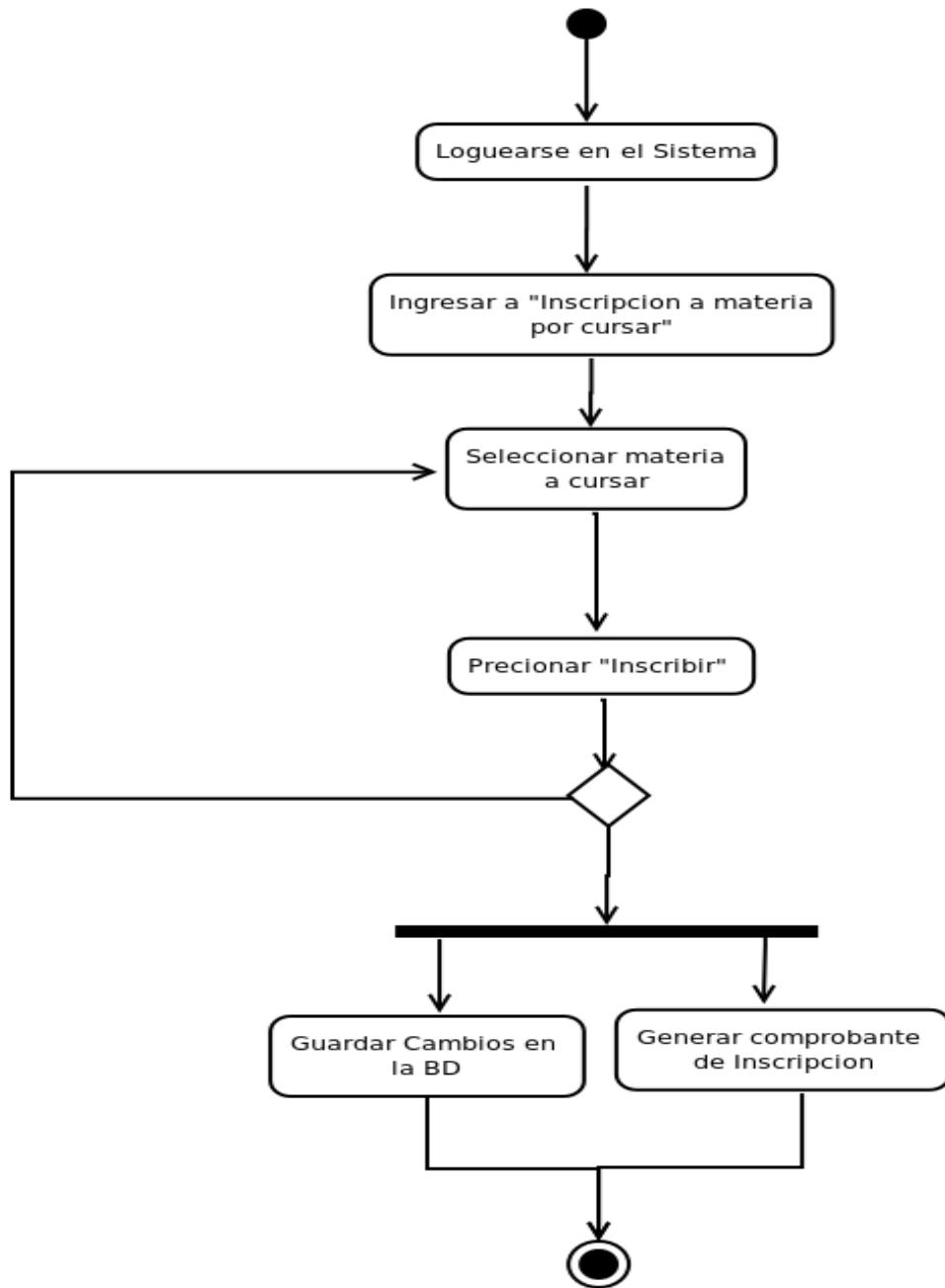




DIAGRAMA DE ACTIVIDAD: "CERTIFICADO DE ALUMNO REGULAR"

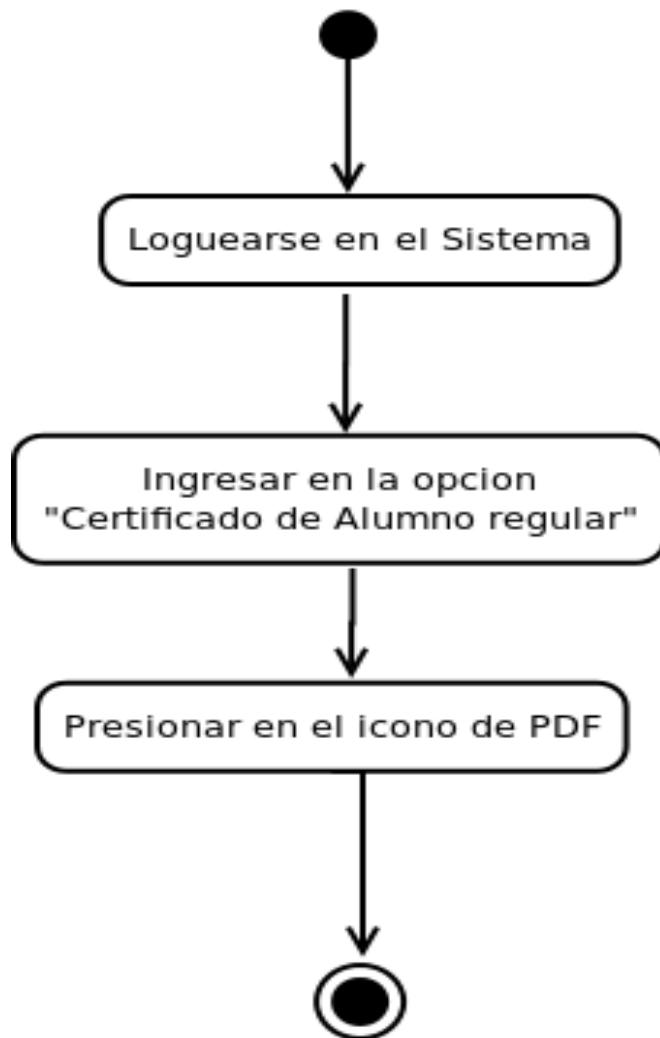




DIAGRAMA DE ACTIVIDAD: "AGREGAR MATERIA"

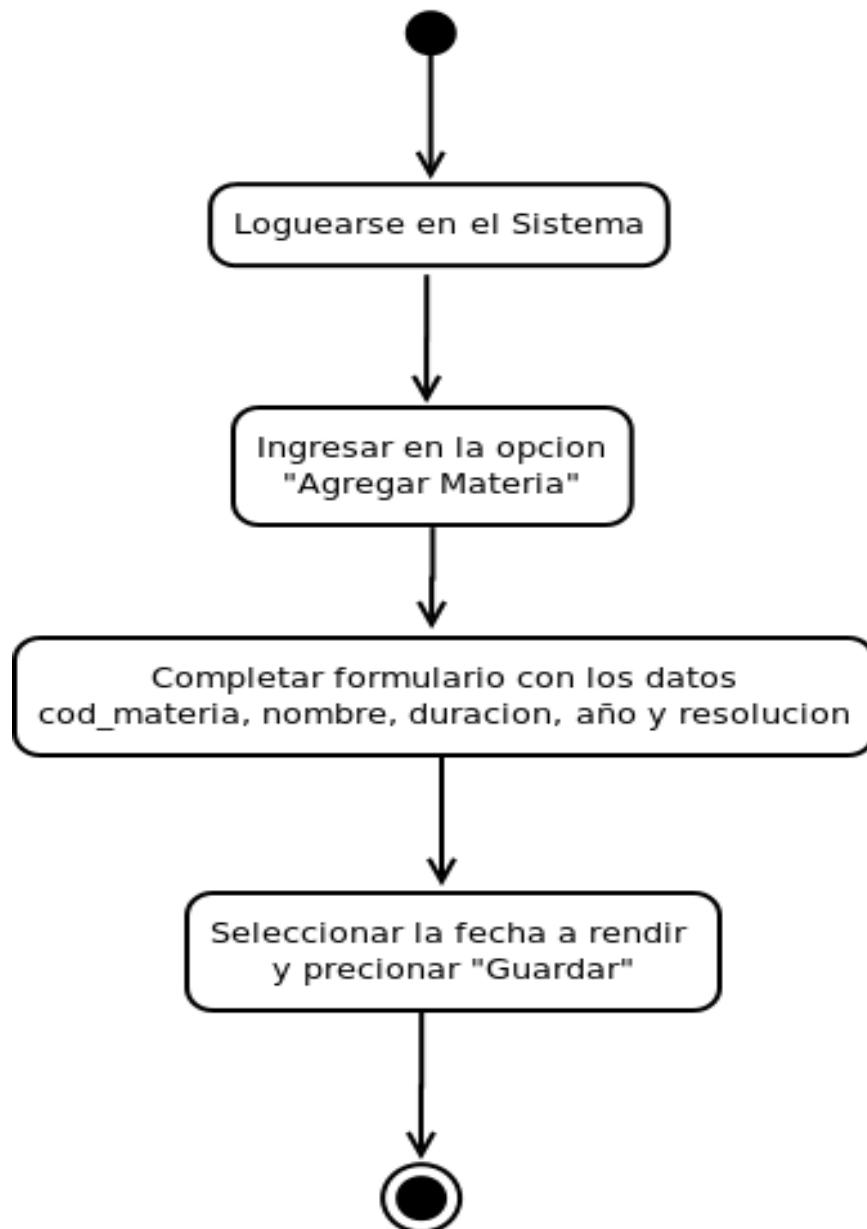




DIAGRAMA DE ACTIVIDAD: "AGREGAR CARRERA"

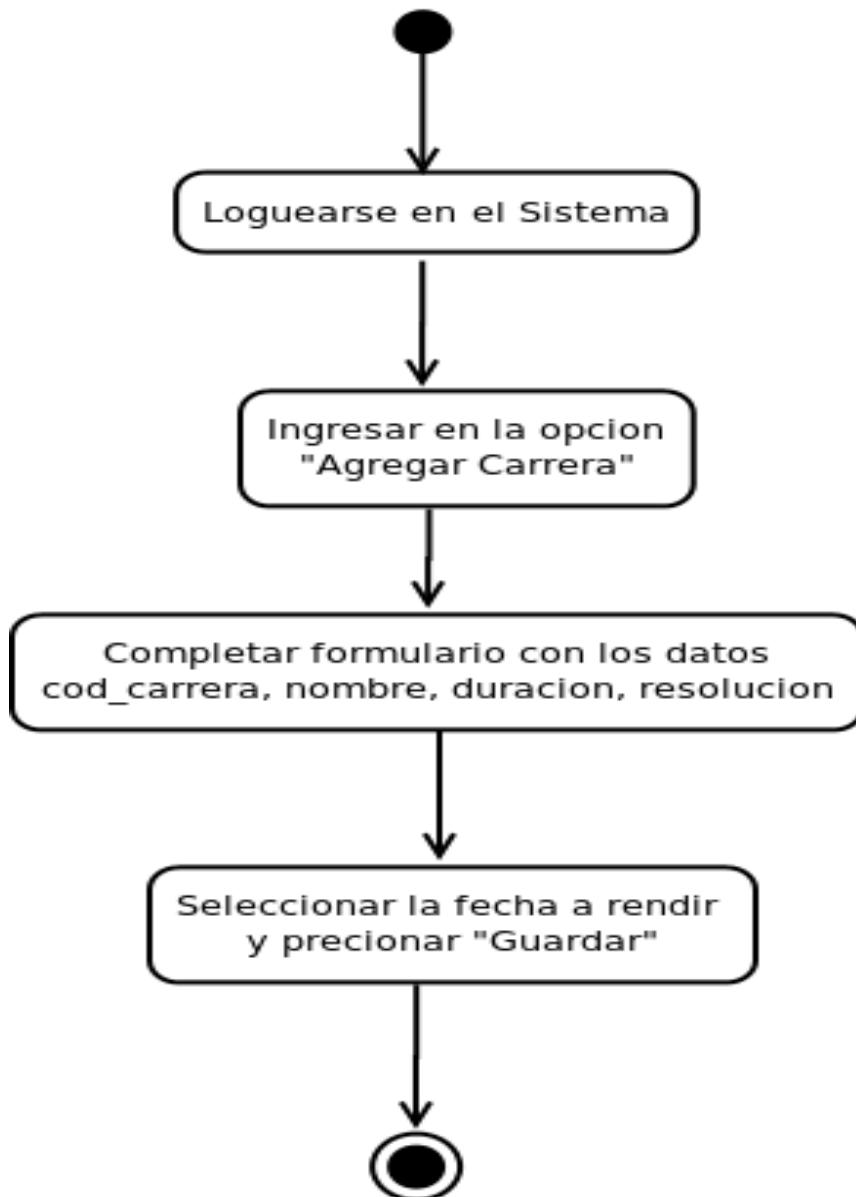
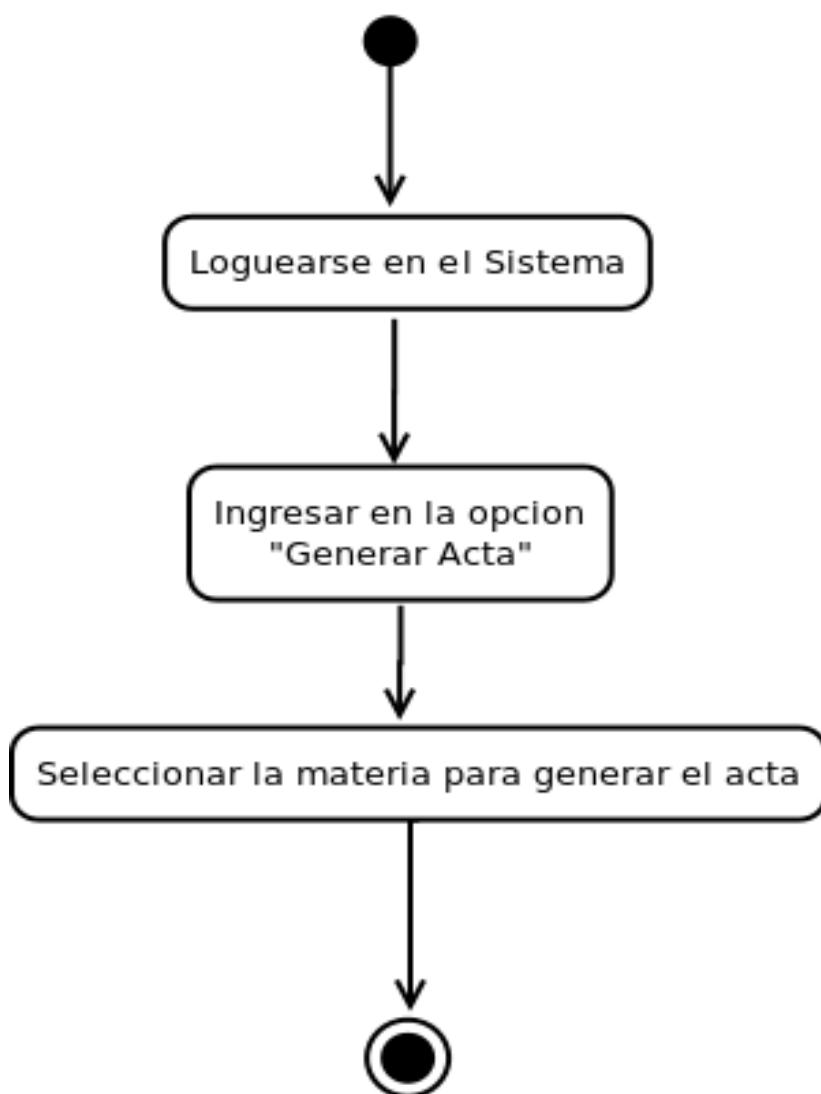




DIAGRAMA DE ACTIVIDAD: "GENERAR ACTA"







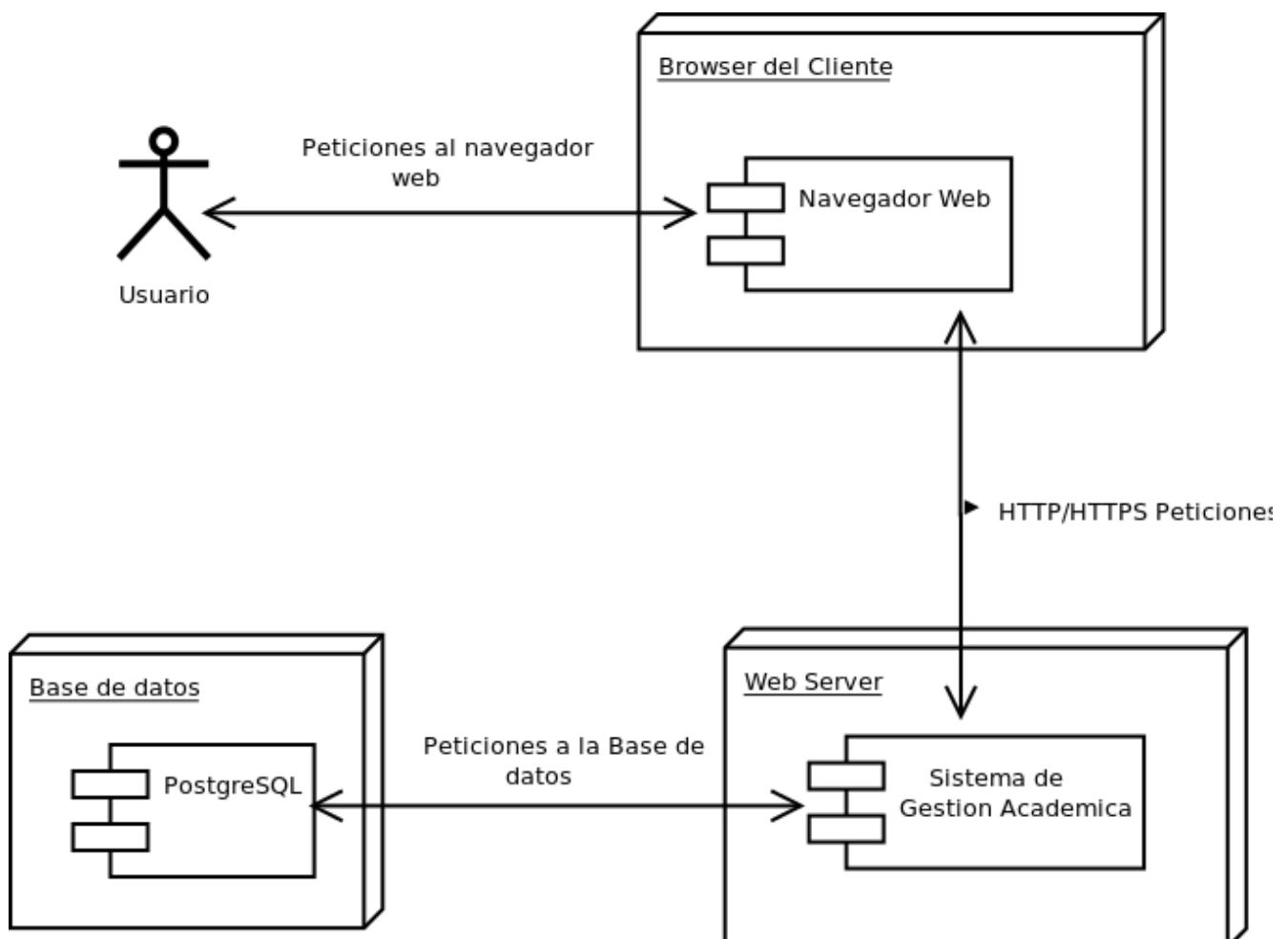
ENTORNO DE GESTION ACADEMICA

DIAGRAMA DE DESPLIEGUE

VERSION 1.0



DIAGRAMA DE DESPLIEGUE “EGA”







ENTORNO DE GESTION ACADEMICA

CODIGO FUENTE

VERSION 1.0



ESTRUCTURA DEL PROYECTO

FOLDERS	SistemaEGA	templates
▼ SistemaEGA	► SistemaEGA	▼ alumno
▼ SistemaEGA	► SistemaEGA	certificado.html
▼ settings	► apps	contacto.html
__init__.py	▼ alumno	historial_academico.html
base.py	► migrations	index.html
email_info.py	__init__.py	lista_finales.html
local.py	admin.py	lista_materias.html
production.py	forms.py	login.html
staging.py	models.py	preinscripcion.html
test.py	tests.py	user_update_form.html
__init__.py	urls.py	user_detail.html
urls.py	views.py	useralumno_update_form.html
wsgi.py	▼ bedel	▼ bedel
▼ apps	__init__.py	agregar_carrera.html
► alumno	admin.py	agregar_materias.html
► bedel	forms.py	agregar_mesa_final.html
► home	models.py	agregar_nota_final.html
__init__.py	tests.py	busqueda.html
▼ media	urls.py	lista_alumnos.html
► perfiles	views.py	lista_alumnos_acta.html
▼ static	▼ home	lista_carreras.html
► css	__init__.py	lista_materias.html
► css2	admin.py	lista_materias_acta.html
► fonts	forms.py	lista_mesa_final.html
► ima	models.py	menu_bedel.html
► js	tests.py	► errores
▼ templates	urls.py	▼ home
► alumno	views.py	carrera_update_form.html
► bedel	__init__.py	carrera_confirm_delete.html
► errores	▼ media	create_final.html
► home	► perfiles	inscripcionfinal_confirm_delete.html
► pdf	▼ static	inscripcionmateria_form.html
base.html	► css	materia_update_form.html
base2.html	▼ css2	materia_confirm_delete.html
dudas.txt		materia_detail.html
manage.py		materia_form.html



CODIGO FUENTE: MODELOS

MODELO “USER”

```
1 #encoding:utf-8
2 from django.db import models
3 from django.contrib.auth.models import AbstractBaseUser, BaseUserManager, PermissionsMixin
4 from PIL import Image
5 from django.core.mail import EmailMessage
6 from django.core.validators import RegexValidator
7
8
9 class UserManager(BaseUserManager):
10     def _create_user(self, username, email, password, is_staff,
11                     is_superuser, **extra_fields):
12         if not email:
13             raise ValueError('el email debe ser obligado')
14         email = self.normalize_email(email)
15         user = self.model(username=username, email=email, is_active=True,
16                           is_staff=is_staff, is_superuser=is_superuser, **extra_fields)
17         user.set_password(password)
18         user.save(using = self._db)
19         return user
20
21     def create_user(self, username, email, password, **extra_fields):
22         return self._create_user(username, email, password, False, False, **extra_fields)
23
24     def create_superuser(self, username, email, password, **extra_fields):
25         return self._create_user(username, email, password, True, True, **extra_fields)
26
27
28 class User(AbstractBaseUser,PermissionsMixin):
29
30     nombre_apellido = models.CharField(max_length=800)
31     username = models.CharField(max_length=50, unique=True)
32     email = models.EmailField(max_length=50, unique=True)
33     cod_alumno = models.CharField(max_length=60, blank=True, null=True)
34     #carrera = models.CharField(max_length=50)
35     dni = models.CharField(max_length=10, blank=True, null=True, validators=[
36         RegexValidator(
37             regex = '^[0-9]*$',
38             message ='Solamente puede Ingresar Numeros'
39         )
40     ])
41     lugar_nacimiento = models.CharField(max_length=500)
42     fecha_nacimiento = models.DateField(blank=True, null=True)
43     ciudad_actual = models.CharField(max_length=500)
44     domicilio_actual = models.CharField(max_length=500)
45     tipo_usuario = models.BooleanField(default=False)
46     documentacion_completa = models.BooleanField(default=False)
47     fecha = models.DateField(auto_now=True, auto_now_add=True)
48     imagen = models.ImageField(upload_to='perfiles/', blank=True, null=True)
49
50     is_active = models.BooleanField(default= True)
51     is_staff = models.BooleanField(default= False)
52
53     USERNAME_FIELD = 'username'
54     REQUIRED_FIELDS = ['email']
55     objects = UserManager()
56
57     def get_short_name(self):
58         return self.username
59
```



MODELO “APP HOME”

```
1 |from django.db import models
2 |from apps.alumno.models import User
3 |
4 |
5 |class Carrera(models.Model):
6 |
7 |    cod_carrera = models.IntegerField()
8 |    nombre = models.CharField(max_length=500)
9 |    resolucion = models.CharField(max_length=500, blank=True, null=True)
10 |   duracion = models.IntegerField()
11 |   alumno = models.ManyToManyField(User)
12 |
13 |def __unicode__(self):
14 |
15 |    return self.nombre
16 |
17 |
18 |class Materia(models.Model):
19 |
20 |    cod_materia = models.IntegerField()
21 |    nombre = models.CharField(max_length=500)
22 |    carrera = models.ForeignKey(Carrera)
23 |    duracion = models.CharField(max_length=25)
24 |    anio = models.IntegerField()
25 |    correlativa = models.ManyToManyField('Materia', blank=True, null=True)
26 |
27 |def __unicode__(self):
28 |    return self.nombre
29 |
30 |
31 |class Profesor(models.Model):
32 |
33 |    nombre = models.CharField(max_length=500)
34 |    materia = models.ForeignKey(Materia)
35 |
36 |    def __unicode__(self):
37 |        return self.nombre
38 |
39 |class MesaFinal(models.Model):
40 |
41 |    fecha = models.DateField()
42 |    hora = models.TimeField(blank=True, null=True)
43 |    materia = models.ForeignKey(Materia)
44 |    turno = models.CharField(max_length=100)
45 |    cod_mesa = models.IntegerField()
46 |
47 |def __unicode__(self):
48 |    return self.turno
49 |
50 |
51 |from apps.alumno.models import User
52 |
53 |
54 |
55 |class InscripcionMateria(models.Model):
56 |
57 |    alumno = models.ForeignKey(User)
58 |    materia = models.ForeignKey(Materia)
59 |    regular = models.BooleanField(default=False)
60 |
61 |
62 |class InscripcionFinal(models.Model):
63 |
64 |    cod_inscripcion = models.CharField(max_length=60, unique=True)
65 |    alumno = models.ForeignKey(User)
66 |    materia = models.ForeignKey(Materia)
67 |    mesa = models.ForeignKey(MesaFinal)
68 |
69 |
70 |class HistorialAcademico(models.Model):
71 |
72 |    alumno = models.ForeignKey(User)
73 |    materia = models.ForeignKey(Materia)
74 |    nota = models.IntegerField()
75 |    fecha = models.DateField()
76 |
```



CODIGO FUENTE: VISTAS

VISTA “APP ALUMNOS”

```
20 #Vista de Login
21 def index(request):
22     if request.method == 'POST':
23         form = LoginForm(request.POST)
24         username = request.POST['username']
25         password = request.POST['password']
26         user = authenticate(username=username, password=password)
27         if user is not None: #si los datos son correctos y no estan vacios
28             if user.is_active and user.documentacion_completa == True and user.tipo_usuario == True:
29                 login(request,user) #se loguea como bedel
30                 return redirect('/index-bedel') #lo redirecciona al index
31             else:
32                 if user.is_active and user.documentacion_completa == True:
33                     login(request,user) #se loguea como alumno
34                     return redirect('/index') #lo redirecciona al index
35                 else:
36                     return redirect('/documentacion')
37
38         else:
39             return redirect('/error')
40
41     else:
42
43         form = LoginForm()
44         ctx = {'form':form}
45         return render(request, 'alumno/login.html', ctx )
46
47
48
49 #Vista para Logout
50 @login_required(login_url='/')
51 def cerrar(request):
52     logout(request)
53     return HttpResponseRedirect('/')
54
55 #Vista de Formulario de Preinscripcion
56 class PreinscripcionView(FormView):
57
58     template_name = 'alumno/preinscripcion.html'
59     form_class = UserForm
60     success_url = '/'
61
62     def form_valid(self, form):
63
64         user = form.save() #asignamos a la variable user el formulario
65         user.set_password(form.cleaned_data['password']) #guardamos la clave encriptada
66         user.save() #guardamos el formulario
67         return super(PreinscripcionView, self).form_valid(form)
68
69     def form_invalid(self, form):
70
71         return super(PreinscripcionView, self).form_invalid(form)
72
73
74
75 #Index Alumno
76 class IndexView(LoginRequiredMixin, TemplateView):
77
78     template_name = 'alumno/index.html'
79     login_url = '/'
80     model = User
81     context_object_name = 'usuario'
82
83     def get_context_data(self, **kwargs):
84
85         context = super(IndexView, self).get_context_data(**kwargs)
86         context['fecha'] = datetime.datetime.now().utcnow().replace(tzinfo=utc)
87         return context
88
89
90
91 class PerfilView(LoginRequiredMixin, TemplateView):
92
93     template_name = 'alumno/perfil.html'
94     login_url = '/'
```



VISTA “APP ALUMNOS” (CONTINUACION)

```
101 #Vista para el Perfil de Usuario
102 class UserDetailView(LoginRequiredMixin, DetailView):
103
104     model = User
105     context_object_name = 'usuario'
106     slug_field = 'username'
107
108     def get_context_data(self, **kwargs):
109
110         context = super(UserDetailView, self).get_context_data(**kwargs)
111         context['carrera'] = Carrera.objects.get(alumno = self.request.user)
112         return context
113
114
115 #Vista para mandar Correo al Administrador
116 @login_required(login_url='/')
117 def contacto_view(request):
118     form_class = ContactForm
119     email = ""
120     titulo = ""
121     texto = ""
122     if request.method == "POST":
123         formulario = ContactForm(request.POST)
124         if formulario.is_valid():
125             email = formulario.cleaned_data['Email']
126             titulo = formulario.cleaned_data['Titulo']
127             texto = formulario.cleaned_data['Texto']
128             message = "Mensaje de : [%s][%s]<br> Texto : [%s]"%(request.user.nombre_apellido, email, texto)
129             msg = EmailMessage( subject= titulo,
130                                 body = message,
131                                 from_email= email,
132                                 to =['fabian.falon@gmail.com'])
133
134             msg.send()
135             return redirect('/index')
136     else:
137         formulario = ContactForm()
138         ctx = {'form':formulario, 'email':email, 'titulo':titulo, 'texto':texto}
139         return render_to_response('alumno/contacto.html',ctx,context_instance=RequestContext(request))
140
141 #Vista que muestra un error en el login si no presento la documentacion
142 class ErrorDocumentacionView(TemplateView):
143
144     template_name = 'errores/documentacion_incompleta.html'
145
146
147
148 #Vista datos incorrectos en el form de login
149 class ErrorDatosView(TemplateView):
150
151     template_name = 'errores/error_datos.html'
152
153
154 #Visualizar el Historial Academico
155 class HistorialAcademicoView(TemplateView):
156
157     models = HistorialAcademico
158     template_name = 'alumno/historial_academico.html'
159
160     def get_context_data(self, **kwargs):
161
162         context = super(HistorialAcademicoView, self).get_context_data(**kwargs)
163         context['historial_materias'] = HistorialAcademico.objects.filter(alumno = self.request.user)
164         #Filtramos en HistorialAcademico por alumno mientras sea igual a alumno logueado
165         return context
166
```



VISTA “APP ALUMNOS IMPORT”

```
1 from django.contrib import messages
2 from django.core.mail import EmailMessage, EmailMultiAlternatives
3 from django.core.urlresolvers import reverse_lazy
4 from django.http import HttpResponseRedirect, HttpResponse
5 from django.shortcuts import render_to_response, redirect, render, RequestContext
6 from django.views.generic import TemplateView, FormView, DetailView, UpdateView, DeleteView
7 from wkhtmltopdf.views import PDFTemplateView, PDFTemplateResponse
8
9 from django.contrib.auth.forms import AuthenticationForm
10 from django.contrib.auth import authenticate, login, logout
11 from django.contrib.auth.decorators import login_required
12 from braces.views import LoginRequiredMixin
13 from apps.alumno.models import User
14 from apps.home.models import HistorialAcademico, InscripcionMateria, InscripcionFinal, Carrera
15 from .forms import UserForm, LoginForm, ContactForm, EditForm
16 from django.utils.timezone import utc
17 import datetime
18
```



VISTA “APP HOME”

```
1 import random
2 from django.shortcuts import redirect, HttpResponseRedirect, render
3 from braces.views import LoginRequiredMixin
4 from django.views.generic import FormView, CreateView, DetailView
5 from django.views.generic import TemplateView, DeleteView
6 from .models import Materia, User, InscripcionFinal
7 from .models import InscripcionMateria, Carrera, MesaFinal
8 from .forms import FinalForm, MateriaForm
9
10
11 #Mostrar lista de materias para rendir examen Final
12 class FinalCreateView(LoginRequiredMixin, CreateView):
13
14     model = Materia
15
16     def get_context_data(self, **kwargs):
17
18         context = super(FinalCreateView, self).get_context_data(**kwargs)
19         #Consultamos en que carrera esta inscripto el Alumno logueado
20         carrera = Carrera.objects.get(alumno_in = [self.request.user])
21         #Traemos la lista de materias de la carrera en la que esta inscripto el Usuario
22         context['total_materias_finales'] = Materia.objects.filter(carrera = carrera).order_by('anio')
23         return context
24
25
26 #Muestra el detalle de la materia para inscribirse al examen final
27 class FinalDetailView(LoginRequiredMixin, DetailView):
28
29     template_name = 'home/create_final.html'
30     model = Materia
31     form_class = FinalForm
32     context_object_name = 'materia'
33
34     def get_context_data(self, *args, **kwargs):
35
36         mesa = MesaFinal()
37         form_class = FinalForm()
38         context = super(FinalDetailView, self).get_context_data(**kwargs)
39         #Filtramos la materia en la que hicimos click para que nos muestre su detalle
40         context['mesa'] = MesaFinal.objects.filter(materia = context['object'])
41         return context
```



VISTA “APP HOME (CONTINUACION)”

```
43     def post(self, request, *args, **kwargs):
44
45         inscripcionfinal = InscripcionFinal()
46         inscripcionfinal.alumno = request.user# Guardamos el usuario que esta haciendo la peticion
47         inscripcionfinal.materia = Materia.objects.get(pk = kwargs['pk'])
48         # Buscamos la materia que sea igual pk pasada por parametro
49         inscripcionfinal.mesa = MesaFinal.objects.get(fecha = request.POST['fecha'])
50         # Filtramos mesa final segun la mesa que se paso por POST
51         cpu = random.choice(range(100000))#Generamos numero aleatorio
52         materia = inscripcionfinal.materia
53         numero = cpu , inscripcionfinal.materia
54         inscripcionfinal.cod_inscripcion = numero
55         inscripcioncorrelativa = Materia.objects.filter(materia_pk = kwargs['pk'])
56         #Consulta: Me trae una lista de materias correlativas de la materia en la que hice click
57         validacion = InscripcionMateria.objects.filter(materia = inscripcioncorrelativa, alumno = inscripcionfinal.alumno)
58         # Preguntamos en InscripcionMateria si hay registro cuya materia sea igual a la consulta anterior
59         #y si ese registro pertenece al alumno logueado
60         if validacion:
61             inscripcionfinal.save()
62             return redirect('/inscripcion-finales')
63         else:
64             contexto = {
65                 'inscripcioncorrelativa': inscripcioncorrelativa,
66             }
67             return render(request, 'errores/error-inscripcion.html', contexto)
68
69
70     # Muestra lista de materias para inscribirse a cursar
71 class MateriaCreateView(LoginRequiredMixin, CreateView):
72
73     model = InscripcionMateria
74     success_url = '/index'
75
76     def get_context_data(self, **kwargs):
77
78         context = super(MateriaCreateView, self).get_context_data(**kwargs)
79         carrera = Carrera.objects.get(alumno__in = [self.request.user])
80         # Filtramos en la clase Carrera segun el usuario registrado
81         context['total_materias'] = Materia.objects.filter(carrera = carrera).order_by('anio')
82         # Filtramos las materias de esa carrera
83         return context
```



VISTA “APP HOME (CONTINUACION)”

```
88 # Muestra el detalle de las materias a inscribirse a cursar
89 class MateriaDetailView(LoginRequiredMixin, DetailView):
90
91     model = Materia
92     form_class = MateriaForm
93     context_object_name = 'materia'
94
95     def post(self, request, *args, **kwargs):
96
97         inscripcion = InscripcionMateria()
98         inscripcion.alumno = request.user
99         inscripcion.materia = Materia.objects.get(pk = kwargs['pk'])
100        inscripcioncorrelativa = Materia.objects.filter(materia_pk = kwargs['pk'])
101        inscripcion.regular = True
102        validacion = InscripcionMateria.objects.filter(materia = inscripcioncorrelativa, alumno = inscripcion.alumno)
103        if validacion:
104            inscripcion.save()
105            return redirect('/inscripcion-materias')
106        else:
107            contexto = {
108                'inscripcioncorrelativa': inscripcioncorrelativa,
109            }
110            return render(request, 'errores/error-inscripcion.html', contexto)
111
112
113 # Muestra el template de Error
114 class ErrorInscripcionView(LoginRequiredMixin, TemplateView):
115
116     template_name = 'errores/error-inscripcion.html'
117
118
119 # Muestra el template del menu para el bedel
120 class BedelView(LoginRequiredMixin, TemplateView):
121
122     template_name = 'bedel/menu_bedel.html'
123
```



VISTA “APP HOME (CONTINUACION)”

```
125 # Eliminar Final
126 class FinalDeleteView(LoginRequiredMixin, DeleteView):
127
128     model = InscripcionFinal
129     success_url = '/inscripcion-finales'
130
131     def delete(self, request, *args, **kwargs):
132
133         self.object = self.get_object()
134         self.object.delete()
135
136         return HttpResponseRedirect(self.get_success_url())
```

CODIGO FUENTE: URLs

URL “APP ALUMNO”

```
1 from django.conf.urls import patterns, include, url
2 from .views import IndexView, PerfilView, LoginView, PreinscripcionView, AlumnoUpdateView, ListaMateriasView, Cer-
3 from .views import ErrorDocumentacionView, UserDetailView, ErrorDatosView, HistorialAcademicoView, ListaFinalView
4 from apps.home.views import BedelView
5
6 urlpatterns = patterns('',
7
8     url(r'^index/$', IndexView.as_view(), name='index'),
9     url(r'^$', LoginView.as_view()),
10    url(r'^documentacion/$', ErrorDocumentacionView.as_view(), name='documentacion'),
11    url(r'^error/$', ErrorDatosView.as_view(), name='error'),
12    url(r'^preinscripcion/$', PreinscripcionView.as_view() , name='preinscripcion'),
13    url(r'^perfil/$', PerfilView.as_view(), name='perfil'),
14
15    url(r'^usuario/(?P<slug>[-\w]+)/$', UserDetailView.as_view(), name='user_detail'),
16    url(r'^usuario/(?P<slug>[-\w]+)/edit/$', AlumnoUpdateView.as_view(), name='user_update'),
17
18    url(r'^historial-academico/$', HistorialAcademicoView.as_view(), name='historial'),
19    url(r'^consulta-materias/$', ListaMateriasView.as_view(), name='materias-cursar'),
20    url(r'^consulta-finales/$', ListaFinalView.as_view(), name='materias-final'),
21    url(r'^certificado-alumno-regular/$', CertRegularView.as_view(), name='certificado-regular'),
22
23    url(r'^contacto/$', 'apps.alumno.views.contacto_view', name='contacto'),
24
25    #operaciones del bedel
26    url(r'^index-bedel/$', BedelView.as_view() , name='bedel'),
27
28 )
```



ENTORNO DE GESTION ACADEMICA

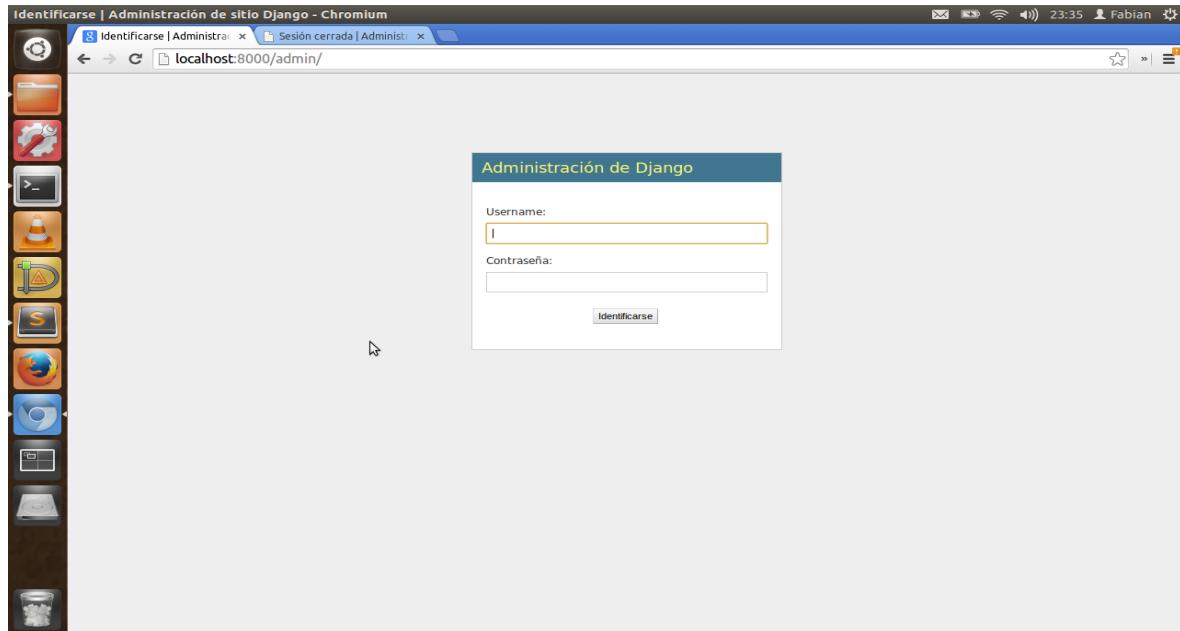
INTERFAZ GRAFICA

VERSION 1.0

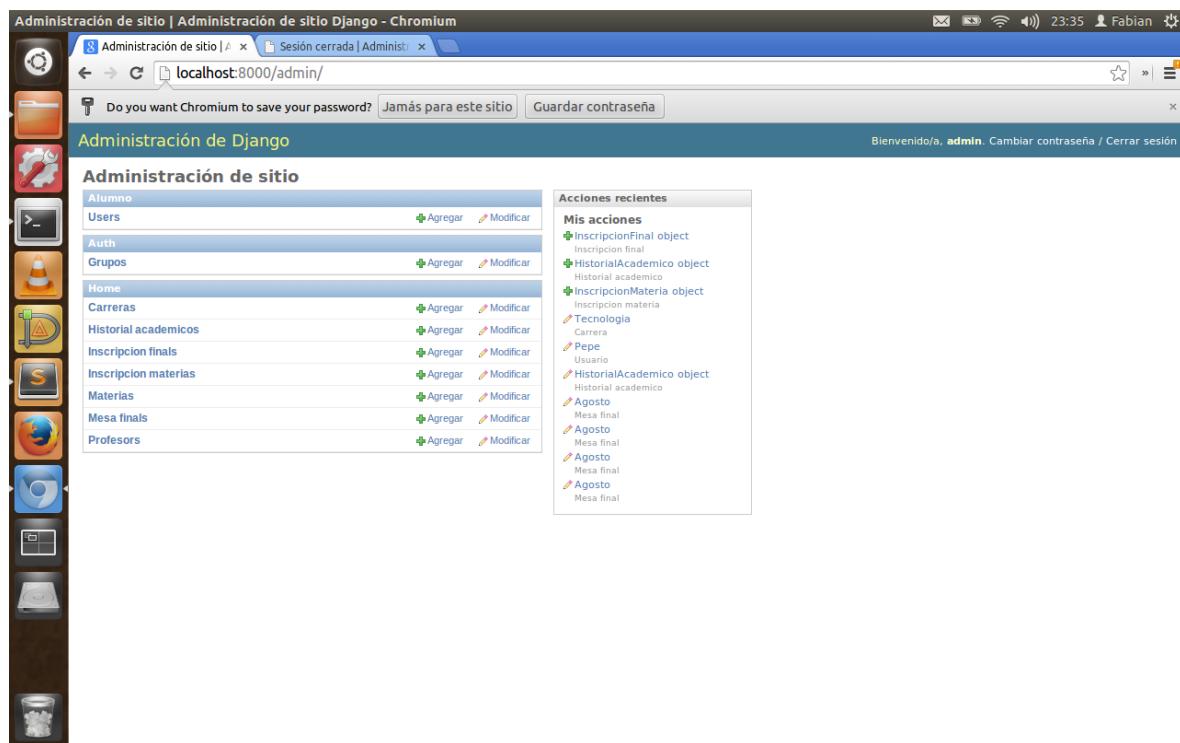


INTERFAZ GRAFICA MODULO “SUPERUSER”

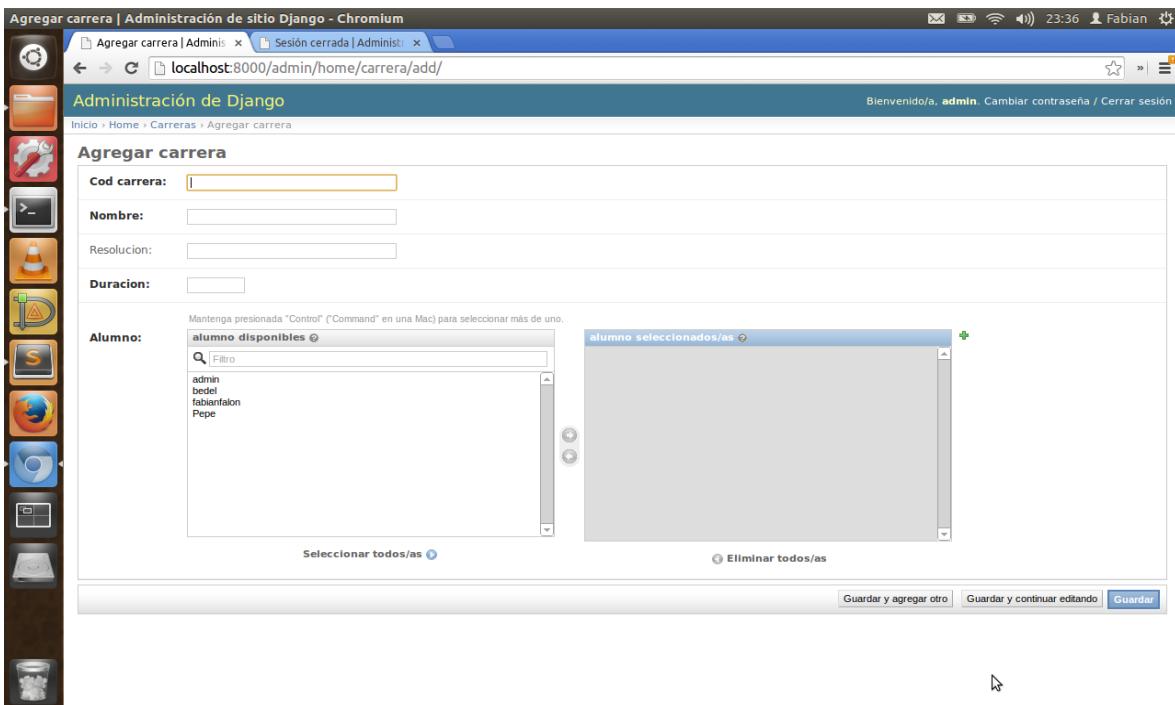
LOGUEARSE EN EL SISTEMA



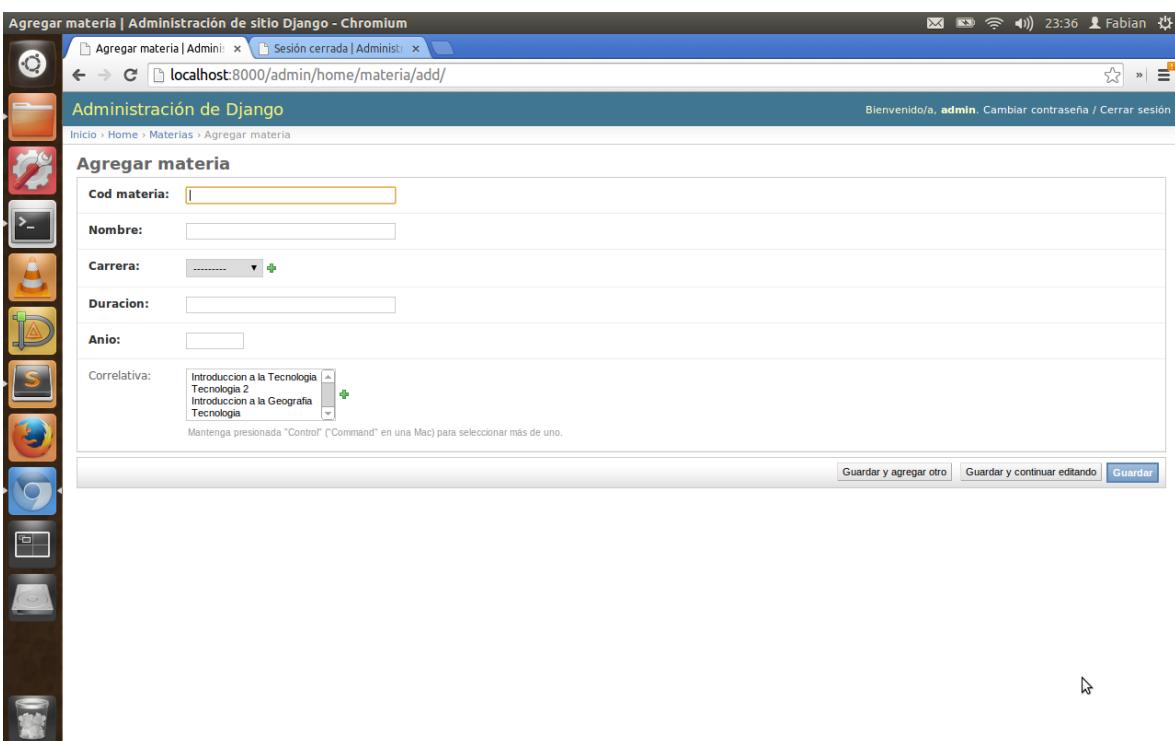
MOSTRAR MENÚ ADMIN



AGREGAR UNA CARRERA



AGREGAR UNA MATERIA





LISTAR MATERIAS

Nombre	Carrera	Duración	Cod materia
Introducción a la Tecnología	Tecnología	Anual	001
Tecnología 2	Tecnología	Cuatri	004
Introducción a la Geografía	Geografía	Anual	1
Tecnología	Tecnología	Anual	2



INTERFAZ GRAFICA MODULO “BEDEL”

MOSTRAR MENÚ BEDEL

The screenshot shows a blue-themed web application interface. At the top center is the EGA logo with the text "ENTORNO DE GESTION ACADEMICA". Below the logo is a navigation bar with links: Perfil, Carreras, Materias, Finales, Alumnos, Inscripción, and Consultas. On the far right of the navigation bar is a user icon labeled "bedel". The main content area displays a message "Hola bedel" within a large blue rectangular box. At the bottom of the page, there is a footer bar with the same "ENTORNO DE GESTION ACADEMICA" text.

AGREGAR CARRERAS

The screenshot shows a modal dialog box titled "Ingresar los datos de la carrera". The dialog contains four input fields: "Cod carrera:" (with placeholder "Ingrese el codigo de la carrera"), "Nombre:" (with placeholder "Ingrese nombre de la carrera"), "Resolucion:" (with placeholder "Ingrese numero de resolucion de la carrera"), and "Duracion:" (with placeholder "Ingrese la duracion de la carrera"). At the bottom of the dialog are two buttons: a green "Guardar" button and a red "Cancelar" button.



LISTAR CARRERAS

Carreras	Eliminar	Modificar
Geografía	X	
Tecnología	X	

AGREGAR MATERIAS

Ingrese los datos de la Materia

Cod materia:

Nombre:

Carrera: -----

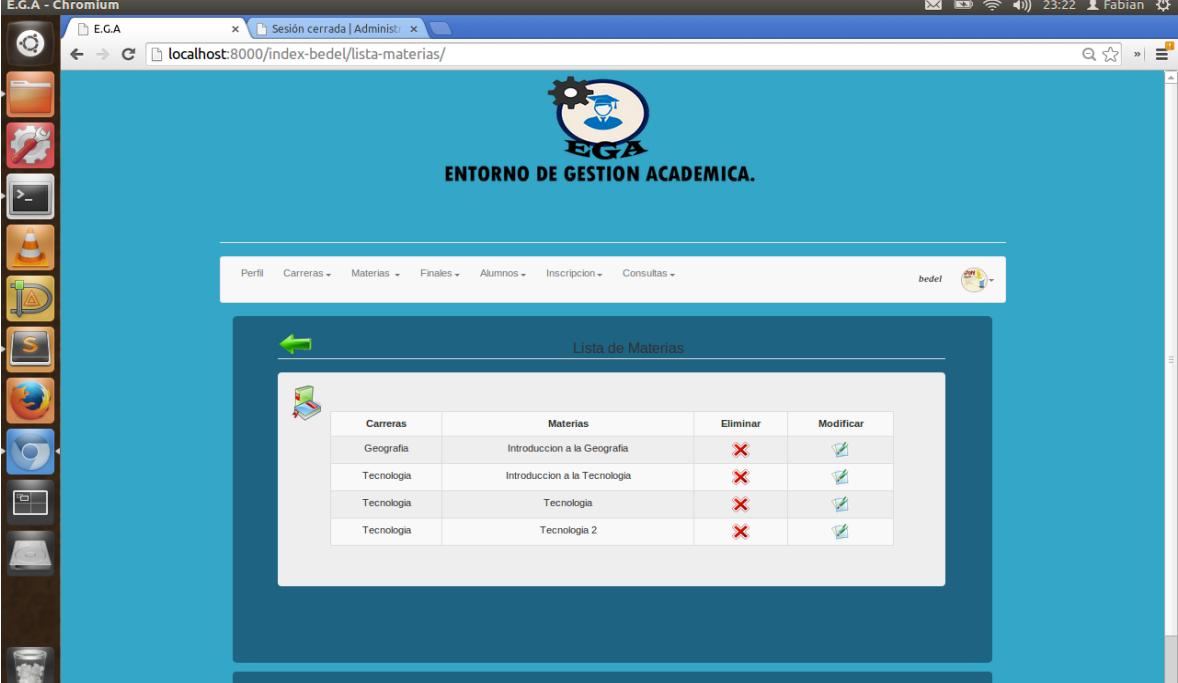
Duracion:

Anio:

Guardar Cancelar

	ENTORNO DE GESTION ACADEMICA FECHA:26/08/14 DOCUMENTO: INTERFAZ GRAFICA V.1
----------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------

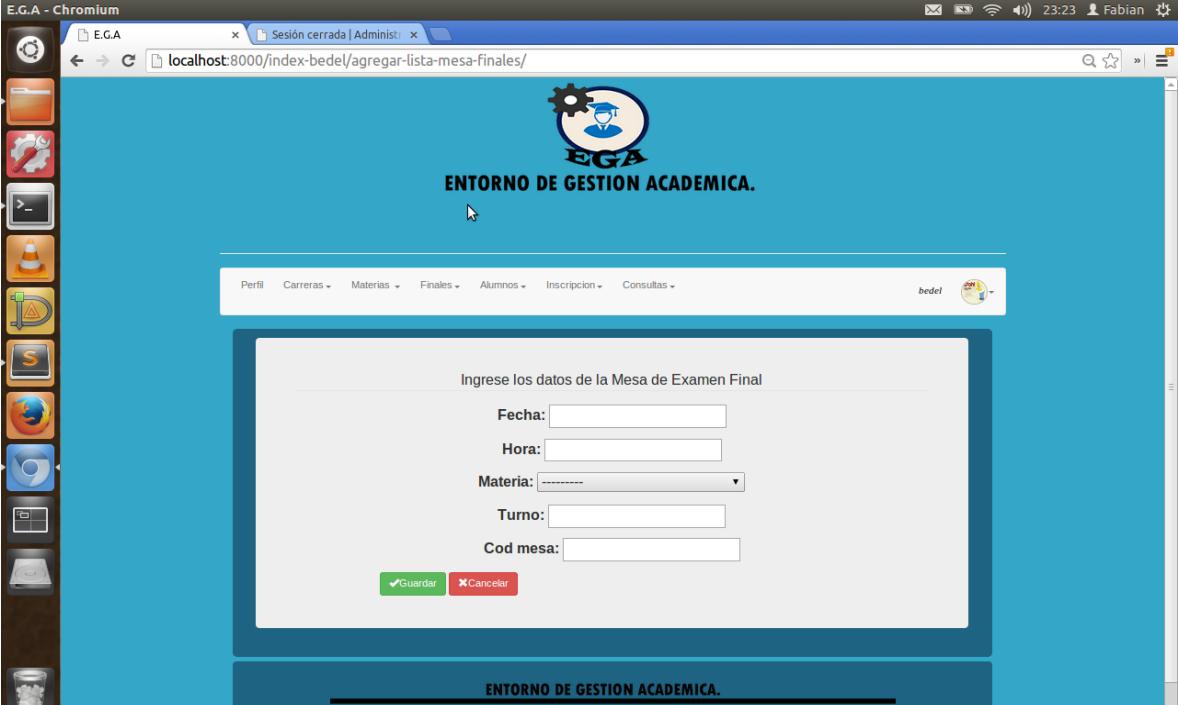
LISTAR MATERIAS



The screenshot shows a web browser window titled "E.G.A - Chromium". The URL is "localhost:8000/index-bedel/lista-materias/". The main content area displays a table titled "Lista de Materias" (List of Subjects). The table has columns for Carreras (Subjects) and Materias (Subjects). The data in the table is as follows:

Carreras	Materias	Eliminar	Modificar
Geografía	Introducción a la Geografía	X	
Tecnología	Introducción a la Tecnología	X	
Tecnología	Tecnología	X	
Tecnología	Tecnología 2	X	

AGREGAR MESAS DE EXAMEN FINAL



The screenshot shows a web browser window titled "E.G.A - Chromium". The URL is "localhost:8000/index-bedel/agregar-lista-mesa-finales/". The main content area displays a form titled "Ingrese los datos de la Mesa de Examen Final" (Enter the data of the Exam Table). The form fields are:

- Fecha:
- Hora:
- Materia:
- Turno:
- Cod mesa:

At the bottom of the form are two buttons: a green "Guardar" (Save) button and a red "Cancelar" (Cancel) button.



LISTAR MESAS DE EXAMEN FINAL

The screenshot shows a Linux desktop environment with a window titled "Lista de Mesas de Examenes Finales". The window displays a table of exam schedules:

Fecha	Materia	Hora	Turno	Código de Mesa	Eliminar	Modificar
17 Julio 2014	Introducción a la Tecnología	17:30:00	Septiembre	1	X	
14 Ago. 2014	Tecnología	16:38:33	Agosto	agosto1	X	
6 Ago. 2014	Tecnología	16:38:33	Agosto	agosto12	X	

INSCRIBIR A MATERIA

The screenshot shows a Linux desktop environment with a window titled "Ingresar los datos de la carrera". The window contains fields for "Alumno" and "Materia" with "Guardar" and "Cancelar" buttons.

Ingresar los datos de la carrera

Alumno
Ingrese el nombre del alumno

Materia
Ingrese el nombre de la materia

✓Guardar ✗Cancelar



LISTAR INSCRIPCIONES A MATERIAS

Alumno	Materia	Modificar
fabianfalon	Introducción a la Tecnología	
fabianfalon	Tecnología	
Pepe	Introducción a la Tecnología	

AGREGAR HISTORIAL ACADÉMICO

Ingrese los datos del alumno a cargar en el historial

Alumno

Materia

Nota

Código de Acta

Fecha



LISTAR ALUMNOS INSCRIPTOS

Carrera	Nombre	Eliminar	Modificar
	Fabian Falon	X	X
	Pepe Pereyra	X	X

GENERAR ACTA VOLANTE

Carrera	Materias	Descargar
Geografía	Introducción a la Geografía	X
Tecnología	Introducción a la Tecnología	X
Tecnología	Tecnología	X
Tecnología	Tecnología 2	X



DETALLAR ALUMNO

E.G.A - Chromium 23:27 Fabian

Sesión cerrada | Administrador

localhost:8000/index-bedel/lista_alumnos/editar/fabianfalon/edit/

Datos del Alumno

Nombre apellido: Fabian Falon

Username: fabianfalon

Email: fabian.falon@gmail.com

Cod alumno: Tec001

Dni: 35541117

Lugar nacimiento: Aimogasta

Fecha nacimiento: 17/06/1991

Ciudad actual: La Rioja

Domicilio actual: La Rioja

Tipo usuario:

Documentacion completa:

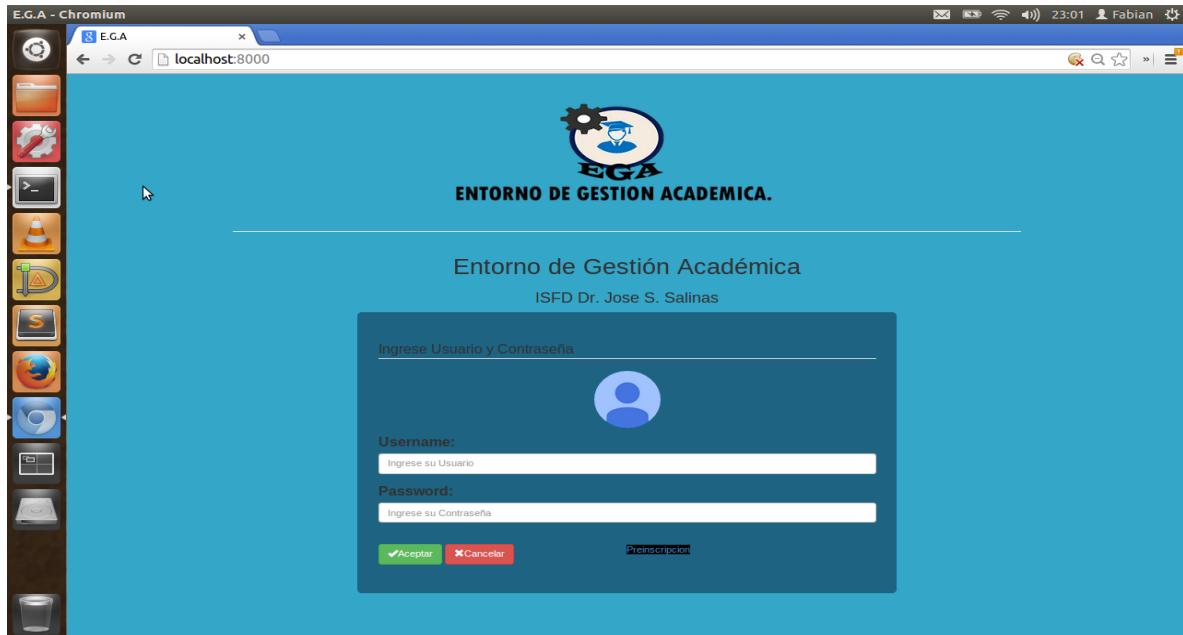
Imagen: Actualmente: [perfles/perfil_23.jpg](#) Eliminar
 Modificar:
[Seleccionar archivo](#) No se eligió archivo

Contraseña: pbkdf2_sha256\$12000\$II5

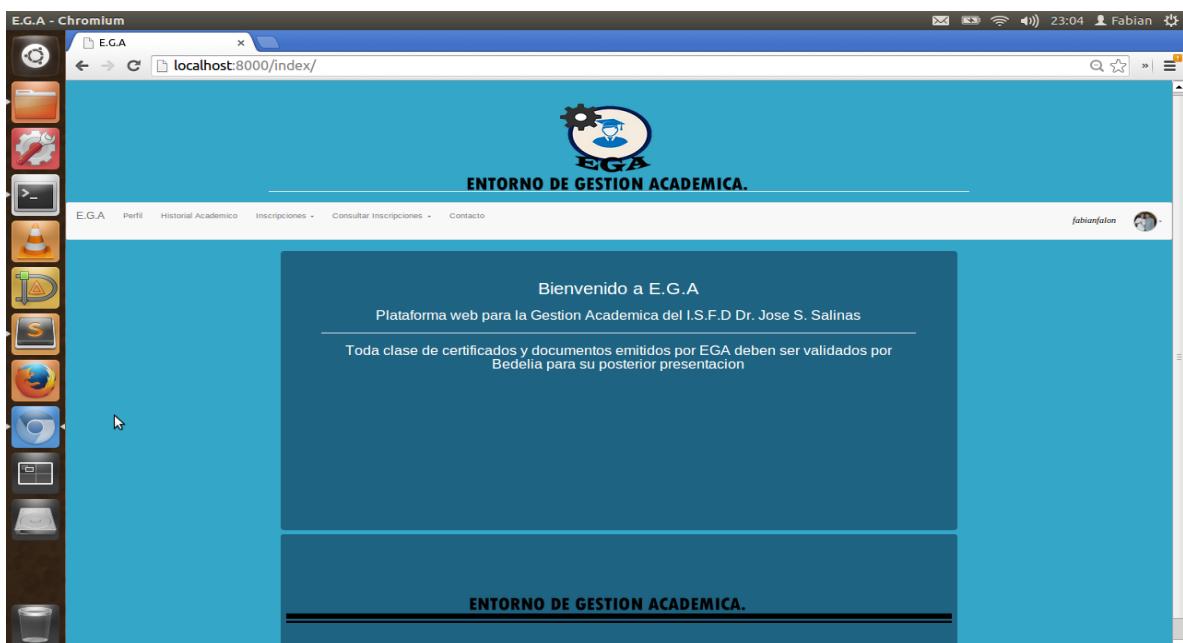
	ENTORNO DE GESTION ACADEMICA FECHA:26/08/14 DOCUMENTO: INTERFAZ GRAFICA V.1
----------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------

INTERFAZ GRAFICA MODULO “ALUMNO”

LOGUEARSE EN EL SISTEMA

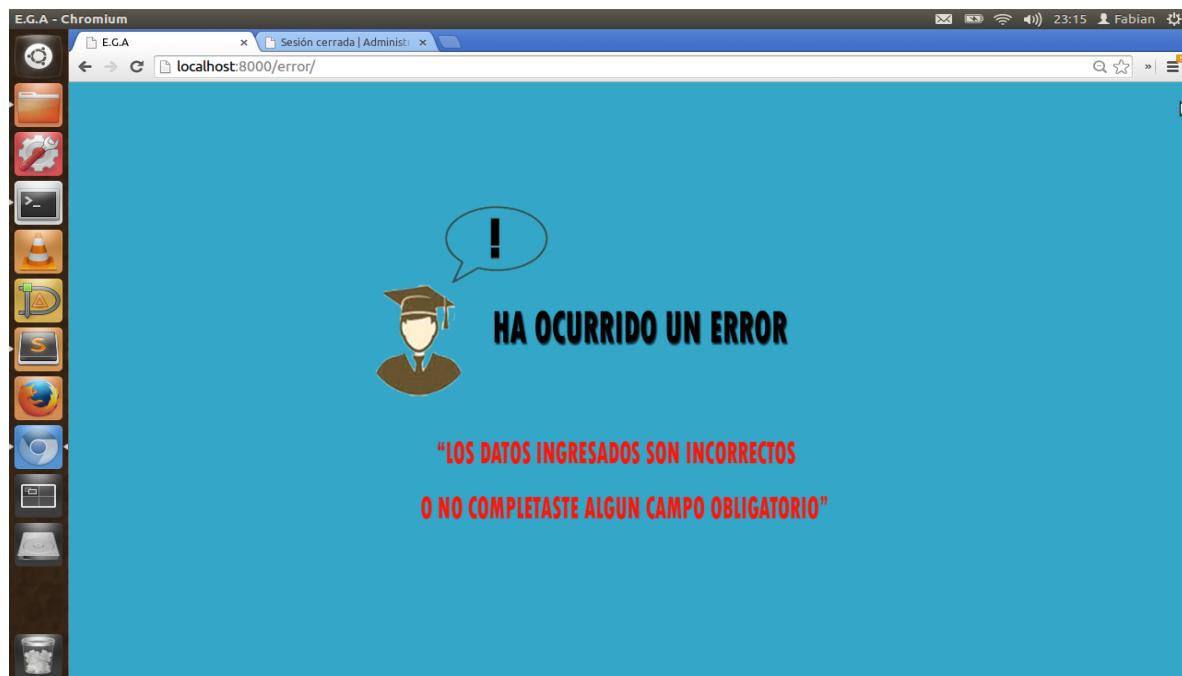


MOSTRAR MENU ALUMNO

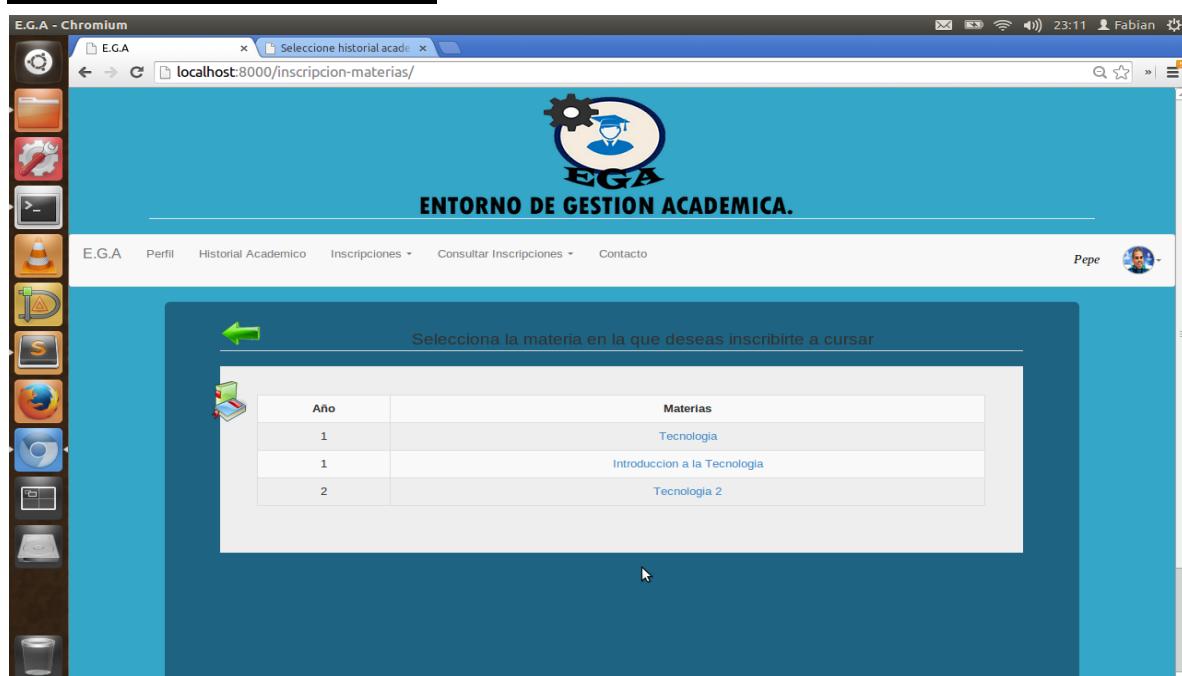




ERROR LOGIN



INSCRIBIRSE A MATERIA





DETALLAR INSCRIPCIÓN A MATERIA

Carrera	Materia	Duración
Tecnología	Tecnología	Anual

Inscribirse Cancelar

CONSULTAR MATERIAS A CURSAR

Alumno	Materia	Nota	Fecha de Final
Pepe	Introducción a la Tecnología		

PDF



DETALLAR INSCRIPCION A EXAMEN FINAL FINAL

E.G.A - Chromium

localhost:8000/incripcion-finales/7/

ENTORNO DE GESTION ACADEMICA.

Carrera	Materia	Duracion	Mesa	Hora
Tecnologia	Tecnologia	Anual	2014-08-06	

✓Inscribirse ✘Cancelar

CONSULTAR INSCRIPCIONES A EXÁMENES FINALES

E.G.A - Chromium

localhost:8000/consulta-finales/

ENTORNO DE GESTION ACADEMICA.

Alumno	Codigo	Materia	Nota	Fecha de Final	Hora	Eliminar Inscripción
Pepe	000000000123	Introducción a la Tecnología		14 Ago. 2014	16:38:33	X

PDF



IMPRIMIR PERMISO DE EXAMEN

PERMISO DE EXAMEN

Consté por la presente que el alumno **Fabian Falon**

Esta habilitado para rendir las Asignaturas que indica a continuación.

N de Orden	Asignatura	Curso	Fecha	Hora	Calificación	Firma presidente de mesa
1	Introducción a la Tecnología		17 Julio 2014	17:30:00		
2	Tecnología		6 Ago. 2014	16:38:33		

Firma del Bedel

NOTAS

1) Para poder rendir examen, el alumno deberá presentar a la mesa examinadora este permiso y su documento de Identidad
2) Los exámenes escritos deben ser tachados con tinta.

MOSTRAR HISTORIAL ACADEMICO

E.G.A - Chromium

localhost:8000/historial-academico/

ENTORNO DE GESTION ACADEMICA.

Historial Academico Pepe Pereyra

Alumno	Materia	Nota	Fecha de Final	Código de Acta
Pepe	Introducción a la Tecnología	8	4 Ago. 2014	tecn001

PDF



IMPRIMIR HISTORIAL ACADÉMICO

Visor de documentos

historial.pdf

Anterior Siguiente 1 (1 de 1) Ajustar al ancho de página

Miniaturas

Instituto Superior de Formacion Docente Dr. Jose S. Salinas

Conste por la presente que el alumno **Fabian Falon**
Rindio el examen final de las materias que a continuacion se detallan.

HISTORIAL ACADEMICO

Alumno	Materia	Nota	Fecha
fabianfalon	Introduccion a la Geografia	1111	29 Junio 2014

MOSTRAR PERFIL DEL ALUMNO

E.G.A - Chromium

localhost:8000/usuario/Pepe/

ENTORNO DE GESTION ACADEMICA.

E.G.A Perfil Historial Academico Inscripciones Consultar Inscripciones Contacto Pepe

Datos de Pepe Pereyra

Carrera: Tecnología
Nombre Y Apellido: Pepe Pereyra
Nombre de Usuario: Pepe
Matrícula: Tecno002
Fecha de Nacimiento: 17 Abril 1991
Email: pepe1@hotmail.com
DNI: 23352102
Usted Se logueo el dia: 4 Ago. 2014

Modificar Datos



MOSTRAR INFORMACION DE CONTACTO

The screenshot shows the 'Contactanos' (Contact us) page of the EGA website. On the left, there is a sidebar with various icons. The main content area features a logo at the top with the text 'ENTORNO DE GESTION ACADEMICA.' Below it is a contact form with fields for Email, Asunto (Subject), and Texto (Text). To the right of the form is a section titled 'Nuestro Establecimiento' (Our Establishment) containing address and contact details. It also includes social media links for Facebook and Twitter.

PREINSCRIPCION

The screenshot shows the 'Ingresar sus datos personales' (Enter your personal data) page for preinscription. The form contains fields for Username, Email, Nombre apellido (Name Surname), Dni (DNI), Lugar nacimiento (Place of birth), Fecha nacimiento (Date of birth), Ciudad actual (Current city), Domicilio actual (Current address), Imagen (Image), Password, and Carrera (Carrera). At the bottom are 'Guardar' (Save) and 'Cancelar' (Cancel) buttons.



ENTORNO DE GESTION ACADEMICA

FECHA:26/08/14

DOCUMENTO: INTERFAZ GRAFICA V.1

MANUAL DE SISTEMA



ENTORNO DE GESTION ACADEMICA

SEGUNDA ITERACION



ENTORNO DE GESTION ACADEMICA

INTERFAZ GRAFICA

VERSION 2.0



ENTORNO DE GESTION ACADEMICA

FECHA:26/08/14

DOCUMENTO: INTERFAZ GRAFICA V.2

MOSTRAR PERFIL DEL ALUMN

The screenshot shows the EGA interface with a blue header bar. The logo is at the top left, followed by the text "ENTORNO DE GESTION ACADEMICA.". Below the header, there's a navigation bar with links: "E.G.A.", "Perfil", "Historial Academico", "Inscripciones", "Consultar Inscripciones", and "Contacto". On the right side of the header, there's a user profile picture and the name "Isabianfalon". The main content area has a dark blue background. It displays a profile picture of a man and a woman holding popcorn buckets. To the right of the picture, there's a list of student details: Carrera: Tecnología; Nombre Y Apellido: Fabian Falon; Nombre de Usuario: fabianfalon; Matrícula: Tec001; Fecha de Nacimiento: 17 Junio 1991; Email: fabian.falon@gmail.com; DNI: 35541117. Below this information is a blue button labeled "Modificar Datos". At the bottom of the content area, it says "ENTORNO DE GESTION ACADEMICA.".

IMPRIMIR CERIFICADO DE ALUMNO REGULAR

Instituto Superior de Formación Docente Dr. José S. Salinas



Certificado de Alumno Regular

Certificamos por este medio que el alumno Fabian Falon con matrícula Tec001, se encuentra en condición de regular en la Carrera Tecnología, pudiendo presentar este certificado ante cualquier autoridad que lo requiera.

Firma del Bedel

Falón Fabián
Peralta Alejandro
Rodríguez Brenda



IMPRIMIR HISTORIAL ACADEMICO

Instituto Superior de Formación Docente Dr. Jose S. Salinas



Conste por la presente que el alumno Fabian Falon
Rindió el examen final de las materias que a continuación se detallan.

HISTORIAL ACADEMICO

Alumno	Materia	Nota	Fecha
fabianfalon	Introducción a la Geografía	9	29 Junio 2014



ENTORNO DE GESTION ACADEMICA

FECHA:26/08/14

DOCUMENTO: INTERFAZ GRAFICA V.2

INDEX DEL SISTEMA DE ALUMNO



ENTORNO DE GESTION ACADEMICA

FECHA:26/08/14

DOCUMENTO: INTERFAZ GRAFICA V.2

CAMBIO INTERFAZ BEDEL

AGREGAR CARRERA

Ingrese los datos de la carrera

Cod carrera:

Nombre:

Resolucion:

Duracion:

AGREGAR A HISTORIAL

Lista de alumnos

Ingrese el nombre del alumno

Fabian Falon
 Pepe Pereyra

Atrás

Materia

Ingrese el nombre de la materia

Nota

Ingrese la nota del alumno

Código de Acta

Ingrese el código del acta

Fecha



AGREGAR A HISTORIAL #2

Ingrese los datos del alumno a cargar en el historial

Alumno

Ingrese el nombre del alumno

Materia

Ingrese el nombre de la materia

Nota

Ingrese la nota del alumno

Código de Acta

Ingrese el código del acta

Fecha dd/mm/aaaa

Guarda Cancelar

agosto de 2014

dom	lun	mar	mié	jue	vie	sáb
27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31	1	2	3	4	5	6

BUSCADOR DE MATERIAS

E.G.A Carreras ▾ Materias ▾ Finales ▾ Alumnos ▾ Inscripcion ▾ Consultas ▾ bedel

Lista de Materias

Escriba el nombre de la Materia y presione enter...

Carreras	Materias	Modificar
Geografía	Introducción a la Geografía	
Tecnología	Introducción a la Tecnología	
Tecnología	Tecnología 2	
Tecnología	Tecnología	

ENTORNO DE GESTION ACADEMICA.



IMPRIMIR ACTA VOLANTE DE EXAMENES

Instituto Superior de Formación Docente Dr. Jose S. Salinas

ACTA VOLANTE DE EXAMENES

Establecimiento: Instituto Superior de Formación Docente Dr Jose S. Salinas

Exámenes de alumnos regulares

Folio N:

Asignatura	Alumno	Calificación Escrita	Calificación Oral	Número de boletas Oral	Número de boletas Escrito	Fecha	Firma alumno	DNI
Introducción a la Tecnología	Fabian Falon							35541117
Introducción a la Tecnología	Pepe Pereyra							23352102

Firmas Autoridades

Presidente : _____ Vocal : _____ Vocal : _____

Total de Alumnos : _____

Aprobados : _____

Aplazados : _____

Ausentes : _____



ENTORNO DE GESTION ACADEMICA

GLOSARIO



A

ACTA VOLANTE: es el documento entregable por el sistema en donde consta la totalidad de alumnos inscriptos para rendir un rendir.

ALUMNO: actor del sistema, que embiste las características de alumno, siendo que hizo la preinscripción y presento la documentación en tiempo y forma.

ALUMNO REGULAR: alumno que embiste la característica de alumno regular en alguna asignatura, es decir, aprobó los parciales ya sea recuperando uno a no, y tiene el 80% de la asistencia a clases junto con los trabajos prácticos aprobados.

B

BASE.PY: archivo de configuración donde se declaran las configuraciones comunes para todas las instancias del proyecto.

BD: base de datos

BEDEL: actor del sistema perteneciente a la institución encargado de la manipulación del módulo “BEDEL”.

BROWSER DEL CLIENTE: es el navegador que posee un usuario del sistema.



C

CARRERA: hace referencia a las distintas carreras terciarias que dicta la institución.

CERTIFICADO DE ALUMNO REGULAR: es un documento entregable por el sistema en el cual consta que un alumno es regular, y que debe ser validado por bedelía.

CORRELATIVIDAD: condición comprobable para la inscripción a una materia por cursar o por rendir, hace referencia a las materias aprobadas o regulares con las que se debe contar para poder efectuar la operación.

D

DJANGO: es un framework web Python de alto nivel que fomenta el rápido desarrollo y diseño limpio y pragmático.

DJANGO-BRACE: incluye varios Mixins que nos van a permitir completar nuestras clases genéricas.

DJANGO-WKHTMLTOPDF: aplicaciones de terceros para poder crear reportes en “.pdf” con Django.



E

EGA: nombre de la aplicación construida “ENTORNO DE GESTION ACADEMICA”.

ERS: documento del sistema, hace referencia a la especificación de requerimientos de software.

EXAMEN FINAL: evaluación de conocimientos necesaria para poder aprobar o no, un espacio curricular y al cual se debe poder inscribir a través de “EGA”.

F

FRAMEWORK: es una estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos de software concretos, que puede servir de base para la organización y desarrollo de software. Típicamente, puede incluir soporte de programas, bibliotecas, y un lenguaje interpretado, entre otras herramientas, para así ayudar a desarrollar y unir los diferentes componentes de un proyecto.

G

GITHUB: repositorio de servicio basado en la web de hosting que ofrece todo el control de versiones distribuidos y gestión de código fuente (SCM) funcionalidad de Git, así como la adición de muchas de sus propias características.



H

HISTORIAL ACADEMICO: Hace referencia al historial académico del alumno donde figuran todas la veces que éste rindió un examen final y la nota obtuvo aprobando o no alguna asignatura, También es uno de los documentos entregables por el sistema y que debe ser validado por el Bedel.

I

INSCRIPCION A EXAMEN FINAL: es la inscripción necesaria efectuada por un alumno para poder rendir un examen final, y que se debe poder realizar a través del “EGA”.

L

LOCAL.PY: archivo de configuración del sistema que muestra cual es la configuración usada cuando se trabaja en un ambiente de desarrollo local.



M

MATERIA: hace referencia a un espacio curricular o asignatura de una carrera.

MESA DE EXAMEN FINAL: Hace referencia al periodo en el cual se llevaran a cabo los exámenes finales.

MODELO: Un modelo es la única fuente definitiva de información sobre sus datos. Contiene los campos esenciales y comportamientos de la información almacena. Generalmente, cada modelo se asigna a una única tabla de base de datos.

MTV: es el patrón de diseño de software que sigue el framework DJANGO

P

PERMISO DE EXAMEN FINAL: es un documento entregable por el sistema, el cual sirve como comprobante de que un alumno se inscribió a un examen final o a varios.

PHYTON: Lenguaje de programación utilizado para el desarrollo de “EGA”.

PILLOW: librería para el procesamiento de imágenes con Django.

PIP: herramienta para la instalación y gestión de paquetes de Python.

POSTGRESQL: SGBD (Sistema de gestión de bases de datos) relacional orientado a objetos y libre, publicado bajo la licencia BSD.



PREINSCRIPCION A CARRERA: Es el primer paso para poder hacer efectiva la inscripción a una carrera, y que puede realizarse a través de "EGA".

PRODUCTION.PY: archivo de configuración que se utiliza_ para la configuración del servidor de producción.

PSYCOPG2: psycopg2 es un adaptador de base de datos PostgreSQL para el lenguaje de programación Python.

PYTHON SETUP TOOLS: biblioteca estable diseñada para facilitar los proyectos de Python.

R

REGULARIDAD: condición que se debe embestir para poder inscribirse a rendir un final de una materia como alumno regular.

REQUEST: es una biblioteca HTTP, escrita en Python.

REQUIREMENTS: Hace referencia a las dependencias necesarias para que nuestro Proyecto pueda funcionar sin mayores complicaciones.

RUP: es un marco de proceso integral que ofrece prácticas de la industria a prueba de software y sistemas de entrega y ejecución y para la gestión eficaz de los proyectos.



S

SOUTH: aplicación para las migraciones de base de datos.

STAGING.PY: archivo de configuración completo de Django

SUPERUSER/SUPER USUARIO/ADMINISTRADOR: actor del Sistema, que ingresa con todos los permisos y puede realizar cambios que otros actores no pueden.

T

TEMPLATE: es un conjunto de archivos que determinan la estructura y el aspecto visual de un sitio web.

TEST.PY: archivo de configuración para realizar pruebas

U

UNIPATH: es un front-end orientada a objetos para las funciones de directorio / archivo dispersos por varios módulos de biblioteca de Python



V

VIRTUALENV: entorno virtual necesario para poder ejecutar la aplicación “EGA”.

W

WKHTMLTOPDF: Librería para poder trabajar con Django y poder generar reportes “.pdf”.

INIT .PY: archivo de configuración que convierte al directorio que lo contiene en modulo