

# UNIVERSIDAD SIMÓN BOLÍVAR

Proyecto Final de Aula



**decAid**

Decanature Aid - version 1.0

- Proyecto de Aula para Algoritmia y Programación I .
  - Profesor: Dr. Ing. Juan Manuel Rúa Ascar.
  - Estudiante: Favio Arturo Galvis Gamboa.
  - Grupo 1-O - I Semestre - Ingeniería de Sistemas - 2015.



**Software para el análisis estadístico de  
estudiantes de Ing. de Sistemas de la  
Universidad Simón Bolívar**

## Contenidos

### Contenidos

#### 1. Introduccion

##### 1.1. Planteamiento del Problema

##### 1.2. Descripcion del Problema

#### 2. Objetivos

##### 2.1. Objetivo General

##### 2.2. Objetivos Especificos

#### 3. Marco de Referencia

#### 4. Recursos

#### 5. Requerimientos Funcionales

#### 6. Modelado del Sistema Propuesto

#### 7. Diseño de Prototipos

#### 8. Conclusiones



## 1. Introduccion

### Que es decAid?

Decanature Aid - ó más corto decAid - es un software para el análisis estadístico de las calificaciones, promedios, cortes y desempeños de los estudiantes de la carrera de Ing. de Sistemas por medio de reportes infográficos y modernos que muestran la información en tiempo real y de manera expedita. Este proyecto fue creado en el marco del Proyecto Final de Aula para la materia de Algoritmos y Programación I del Doc. Ing. Juan Rua.

### 1.1. Planteamiento del Problema

En la universidad simón bolivar de la ciudad de barranquilla, el decano de la facultad de ingeniería de sistemas desea obtener cierta información relacionada con el rendimiento académico de sus estudiantes durante el periodo febrero-junio de 2012, tomando como base el reglamento estudiantil de la institución. Dentro del reglamento, el criterio para aprobar o reprobar el semestre que cursa cada estudiante es el promedio ponderado. Cada estudiante tiene sus datos básicos como el código, el nombre y una proyección de varias asignaturas a cursar en un semestre.

Para el cálculo del promedio ponderado se tienen en cuenta varios subconceptos: Cada asignatura tiene: descripción, tipo, horas teóricas (ht), horas prácticas (hp), horas independientes (hi) y un peso en créditos (x). Las hi son calculadas y para ello se tiene en cuenta el tipo de crédito de las asignaturas, así: si es de tipo a, hi es el doble de ht; si es de tipo b, hi es la suma de (ht + hp).

Créditos (x): es el peso en créditos de cada asignatura y para su calculo se aplica la siguiente formula:

$$X = (ht + hp + hi) * 16 / 48,$$

En donde 16 corresponde al número de semanas del semestre académico y 48 es el número de horas equivalentes a un crédito académico.

Nota\_definitiva (y): es la nota definitiva de cada asignatura en la cual se tienen en cuenta dos exámenes parciales y un final, cada uno con un peso porcentual determinado, así:

$$Y = ex1 * 30\% + ex2 * 30\% + ex3 * 40\%.$$

Las variables ex1, ex2 y ex3 son los tres exámenes para cada asignatura. Cada asignatura se aprueba cuando la nota definitiva es igual o superior a 3.0.

A continuación se presenta un ejemplo el cálculo del promedio ponderado:

Asignatura creditos (x) nota\_definitiva (y) x \* y álgebra y trigonometría 4 3.4 13.6

Geometría analítica 3 2.7 8.1 Introducción a la ingeniería de sistemas 1 3.5 3.5

Fundamentos de programación I 4 3.4 13.6 Competencia comunicativa I 2 2.8 5.6

Crecimiento intragrupal I 1 3.8 3.8 Cátedra bolivariana 1 3.2 3.2 Sumatorias scr = 16 sn = 52.4 Pp: promedio ponderado = 63.7 / 19 = 3.3

Según el reglamento estudiantil, un estudiante aprueba el semestre si su promedio ponderado es igual o superior a 3.0. De lo contrario, lo reprueba. Si aprueba el semestre y no pierde créditos, puede cursar todos los créditos programados en el nivel siguiente; si aprueba el semestre y pierde créditos, puede cursar los créditos que reprobó más los créditos del siguiente nivel que no tengan como requisito los que reprobó. Si reprueba el semestre, repite solamente los créditos que reprobó.

Usted, como ingeniero, debe suministrar la siguiente información a las directivas de la facultad, mediante el análisis, el diseño, la codificación y la puesta en funcionamiento de la solución (sistema de información), teniendo en cuenta todos los estudiantes de la universidad:

1. Cuántos estudiantes aprobaron el semestre.
2. Cuántos estudiantes reprobaron el semestre.
3. Cuántas asignaturas se aprobaron.
4. Cuántas asignaturas se reprobaron.
5. Cual es el promedio notas de las asignaturas aprobadas y las reprobadas.
6. Cuantos creditos aprobaron los estudiantes que aprobaron el semestre.
7. Cuantos creditos reprobaron los estudiantes que aprobaron el semestre.
8. Cuantos creditos aprobaron los estudiantes que reprobaron el semestre.
9. Cuantos creditos reprobaron los estudiantes que reprobaron el semestre.
10. Cuál es el nombre del estudiante que obtuvo el mejor promedio ponderado del curso y cuál fue ese promedio.
11. Cual es el nombre del estudiante que obtuvo el peor promedio ponderado del y cuál fue ese promedio.

12. Cual es el promedio de los promedios ponderados.
13. Cuántas asignaturas tipo a hay y cual es la suma de sus créditos.
14. Cuántas asignaturas tipo b hay y cual es la suma de sus créditos.
15. Cuántos estudiantes obtuvieron promedio ponderado inferior a 2.0.
16. Cuántos estudiantes obtuvieron promedio ponderado entre 2.0 Y menor a 3.0.
17. Cuántos estudiantes obtuvieron promedio ponderado entre 3.0 Y menor a 4.0.
18. Cuántos estudiantes obtuvieron promedio ponderado entre 4.0 Y 5.0.
19. Con base en los resultados 15. A 18., Cuál es el rango de promedios ponderados que mas se presentó.
20. Con base en los resultados 15. A 18., Cuál es el promedio de cada rango de promedios ponderado.
21. Cual es el promedio de notas por estudiante y del curso.
22. Cual es el promedio de créditos tipo a y tipo b.
23. Cual es la proporción entre los créditos tipo a y b con relación al total de créditos cursados por cada estudiante.
24. Para cada estudiante, cuántas asignaturas tienen menos de tres créditos.
25. Para cada estudiante, cuántas asignaturas tienen tres o más créditos.
26. Para cada estudiante, cuántas asignaturas tienen nota definitiva menor que 2.0.
27. Para cada estudiante, cuántas asignaturas tienen nota definitiva entre 2.0 Y menor que 3.0.
28. Para cada estudiante, cuántas asignaturas tienen nota definitiva entre 3.0 Y menor que 4.0
29. Para cada estudiante, cuántas asignaturas tienen nota definitiva entre 4.0 Y 5.0.
30. Cual es la asignatura con mayor cantidad de créditos.
31. Cual es la asignatura con menor cantidad de créditos.
32. Cuantas horas teóricas y cuantas horas prácticas tiene en total cada estudiante.
33. Cuantos parciales aprobados tiene cada estudiante.
34. Cuantos parciales reprobados tiene cada estudiante.

## 1.2. Descripcion del Problema

Resolver las 34 incógnitas presentadas por la decanatura a través del procesamiento de los datos de los estudiantes de la Carrera.

## 2. Objetivos

Presentar la solución al problema en un programa elaborado en Java.

### 2.1. Objetivo General

Presentar los datos solucionados y las respuestas a las incógnitas presentadas en un programa elaborado en el lenguaje de programación Java.

### 2.2. Objetivos Especificos

Elaborar el modelo de datos, identificar los alcances y limitaciones del proyecto, identificar variables, métodos y clases para la construcción de la solución, construir el programa y su entorno gráfico, prueba y presentación del proyecto.

### 3. Marco de Referencia

El software se elabora dentro del proceso de aprendizaje de algoritmos y programación, para la consolidación de los conocimientos de la materia, con los conocimientos entregados por el profesor Dr. Ing. Juan Rua, y el proceso investigativo en el lenguaje Java con entorno de aplicación Web.



## 4. Recursos

Se realizó un procedimiento previo de investigación en varias áreas para identificar el entorno de desarrollo para el programa así.

### Framework

Java, JSP y HTML5 en el código, Marco de trabajo en CSS y JS de Bootstrap,FontAwesome y servidor Tomcat con proyección a BD MySQL en un futuro.

### Centrado en Soluciones

La solución se desarrolló para cumplir los requerimientos del proyecto final de aula de 34 puntos propuesto por el profesor. Encontrará valores agregados en la documentación.

### Muti-Plataforma

El uso de un marco de trabajo Mobile-Friendly hace la página flexible a todas las interfaces y viewports. Más información sobre la versión móvil en la documentación.

### Facilmente Modificable

Extensiva documentación del proyecto y del código construido para facilitar el reuso del mismo, el aprendizaje continuo y la lectura. Documentación autogenerada por Javadoc.

### Open Source

Proyecto definido como Open Source de licencia GNU-GPL para el trabajo y desarrollo continuo del mismo sin limitantes de copyright.

## 5. Requerimientos Funcionales

Para la implementación del proyecto se utilizó el servicio web de **OpenShift**, a través de llaves ssh y de Ruby, con el montaje de un servidor Tomcat 7 y un repositorio Git para el control de cambios y el desarrollo del software, así como su versionamiento.

## 6. Modelado del Sistema Propuesto

Se identificó como características principales, las siguientes propiedades en el proyecto:

- Web Responsiva
- HTML5 + CSS3 como UX
- Framework en Bootstrap y FontAwesome
- Java + JavaScript + JSP para Desarrollo Lógico y de Procesamiento de Datos
- Facilmente Editable
- Buena documentación y facilidad de lectura
- Open Source

## 7. Diseño de Prototipos

### Clases Principales

- Fc\_core : control principal de la aplicación y control del servlet así como el acceso a datos y direccionamiento de navegación
- Fc\_pensum : clase estática para la inicialización del pensum académico completo y la creación de variables estáticas en el proyecto, así como sus métodos de búsqueda de datos
- Fc\_estudiante: clase de creación y almacenamiento de información referente a cada uno de los estudiantes, de acceso privativo y métodos específicos de búsqueda de variables y construcción de objetos.
- Fc\_calificacion: clase de almacenamiento de datos de calificaciones de los estudiante de acceso heredado y con métodos de lectura de variables.
- Fc\_layout: clase de construcción de código html de carácter repetitivo dentro del sitio para realizar construcción de páginas dinámicas dependiendo del contenido y de variables específicas dentro de la aplicación, de carácter público.
- Fc\_summary: clase de procesamiento lógico de variables, es la encargada de realizar toda la lógica de la aplicación y se alimenta de los objetos creados por las demás clases para entregar la información de manera precisa pero sin formato de datos, de esto se encargará Fc\_core.

## 8. Conclusiones

### CODIGO

Separamos en tópicos este software, el sitio web, el servidor, la lógica y los estilos gráficos.

### JAVA

Java es usado como lenguaje lógico para la construcción del software, todas las clases, variables, arreglos, ciclos y operaciones están hechas en Java y por tanto incluidas en clases específicas para tal propósito siguiendo el MER.

### HTML5+JSP

HTML5 + JSP tendrá la construcción del sitio web a través de la conexión con la sesión Java del Servlet y interpretada por JSP para la construcción dinámica de las páginas.

### CSS3

CSS3 como marco de trabajo con los plugins de Bootstrap y FontAwesome y sus complementos en JavaScript para el entorno gráfico y Frontend así como para la compatibilidad de múltiples viewports.

### SERVIDOR

TOMCAT + Apache es el servidor para el montaje e implementación de la solución, a través del stack de XAMPP que realiza una configuración minimalista y rápida.

### DOCUMENTACION

La documentación completa y textos están en el repositorio web del proyecto alojado gratuitamente por GitHub.

<https://github.com/FavioGalvis/decAid>

### Licencia

Este software está hecho en IDE-NetBeans por Favio Galvis como proyecto final de clase, 100% Libre bajo la licencia de GNU General Public License, version 2 (GPL 2.0)