



Sistema Predictivo Basado En Aprendizaje Automático Para La Deserción Estudiantil En Instituciones De Educación Superior

Leandro Pájaro Fuentes
Máster Universitario en Ingeniería Informática

Ignasi Lorente Puchades



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivada [3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

Licencias alternativas (elegir alguna de las siguientes y sustituir la de la página anterior)

A) Creative Commons:



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivada [3.0 España de Creative Commons](#)



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-CompartirIgual [3.0 España de Creative Commons](#)



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial [3.0 España de Creative Commons](#)



Esta obra está sujeta a una licencia de Reconocimiento-SinObraDerivada [3.0 España de Creative Commons](#)



Esta obra está sujeta a una licencia de Reconocimiento-CompartirIgual [3.0 España de Creative Commons](#)



Esta obra está sujeta a una licencia de Reconocimiento [3.0 España de Creative Commons](#)

B) GNU Free Documentation License (GNU FDL)

Copyright © AÑO TU-NOMBRE.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free

Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.

A copy of the license is included in the section entitled "GNU Free Documentation License".

C) Copyright

© (el autor/a)

Reservados todos los derechos. Está prohibido la reproducción total o parcial de esta obra por cualquier medio o procedimiento, comprendidos la impresión, la reprografía, el microfilme, el tratamiento informático o cualquier otro sistema, así como la distribución de ejemplares mediante alquiler y préstamo, sin la autorización escrita del autor o de los límites que autorice la Ley de Propiedad Intelectual.

FICHA DEL TRABAJO FINAL

Título del trabajo:	Sistema predictivo basado en algoritmo de inteligencia artificial para la deserción estudiantil en instituciones de educación superior
Nombre del autor:	Leandro Pájaro Fuentes
Nombre del consultor:	Ignasi Lorente Puchades
Fecha de entrega (mm/aaaa):	09/2015
Área del Trabajo Final:	
Titulación:	Máster Universitario en Ingeniería Informática
Resumen del Trabajo (máximo 250 palabras):	

Abstract (in English, 250 words or less):

Palabras clave (entre 4 y 8):

Tabla de contenido

1	Introducción	1
1.1	Contexto y justificación del trabajo	1
2	Descripción	2
3	Objetivos del Trabajo	3
3.1	Objetivo general.....	3
3.2	Objetivos específicos.....	3
3.3	Objetivos secundarios	3
4	Marco Teórico	5
4.1	Deserción estudiantil	5
4.2	Aprendizaje automático	6
4.3	Modelo Vista Controlador	8
5	Metodología	10
6	Arquitectura de la aplicación.....	12
7	Plataforma de desarrollo.....	13
8	Implementación algoritmo Naïve Bayes.....	15
9	Guía de instalación	18
9.1	Instalación Xampp	18
9.2	Instalación base de datos	23
9.3	Instalación aplicación.....	23
10	Planificación.....	24
10.1	Definición del alcance.....	24
10.2	Creación de la EDT	24
10.3	Realización del cronograma	24
10.4	Definición de criterios de aceptación.....	24
10.5	Establecimiento de organigrama del proyecto.....	24
10.6	Realizar un plan de comunicación.....	24
10.7	Cronograma del proyecto.....	25
10.8	Seguimiento proyecto.....	25
11	Requerimientos.....	27
12	Casos de usos	27
13	Historias de usuario	30
14	Interfaces de usuario	35
14.1	Prototipos	35
14.2	Interfaces de usuario	36
15	Diseño de base de datos	39
15.1	Script base de datos.....	40
16	Pruebas	49
17	Conclusiones	50
18	Anexo 1 Entregables del proyecto	51
19	Anexo 2 Código fuente del proyecto.....	51
20	Anexo 3 Manual de usuario	51
20.1	Acceso al sistema.....	51
20.2	Entrenamiento	52
20.3	Clasificación	53
20.4	Estudiantes.....	54
20.5	Seguimiento	56
20.6	Configuración	57

21	Glosario	60
22	Bibliografía	61

Lista de figuras

No se encuentran elementos de tabla de ilustraciones.

1 Introducción

1.1 Contexto y justificación del trabajo

Una preocupación generalizada en instituciones de educación superior es la deserción. Basados en informes realizados por el Ministerio de Educación Nacional, la deserción por cohorte en programas con nivel de formación universitaria, se ubica en 46.83%¹. Las consecuencias de un alto porcentaje de deserción pueden ser varias, pero se destaca principalmente el no cumplimiento de exigencias sociales, económicas, políticas y culturales de la población colombiana.

Actualmente las instituciones de educación superior emplean diferentes estrategias para disminuir la deserción y en mayor o menor medida estas presentan resultados satisfactorios, sin embargo estas estrategias pueden ser más efectivas cuando se dispone de sistemas que puedan predecir potenciales estudiantes desertores y sobre estos poder enfatizar las acciones encaminadas a aumentar la permanencia.

El desarrollo de un sistema predictivo que tenga en cuenta condiciones sociales del estudiante como estrato socioeconómico, ingresos familiares, así como el género, resultados académicos de estudios previos al pregrado y variables específicas de una determinada institución que afecten las condiciones de permanencia del estudiante, podría convertirse en una herramienta que alerte la eventual deserción de un estudiante, permitiendo la implementación de estrategias que eviten que la deserción pueda materializarse

¹ Sistema para la prevención de la deserción de la educación superior SPADIES. <http://spadies.mineducacion.gov.co/spadies/consultas_predefinidas.html?2>

2 Descripción

La deserción estudiantil es un fenómeno que inicia en las instituciones de educación con formación técnica, tecnológica y/o universitaria, pero que tiene un alto impacto en el desarrollo social y económico de un país, al no contar con nuevas generaciones que posean las competencias y habilidades técnicas que permitan su inserción en el mercado laboral formal, así como el desarrollo de nuevas empresas.

El Ministerio de Educación nacional realiza monitoreo de este fenómeno a nivel del país aportando cifras de deserción en instituciones de educación superior cercanas al 47% por cohorte. Adicionalmente a través del sistema para la prevención de la deserción de la educación superior SPADIES², ha analizado una serie de condiciones que impactan en la deserción de estudiantes, dentro de estas condiciones se pueden destacar, los ingresos familiares, los resultados de pruebas ICFES³, sexo, nivel educativo de la madre, propiedad de la vivienda entre otros, con este análisis cada institución de educación puede tener la información del comportamiento de su deserción por cohorte y las variables que más puedan incidir en este fenómeno. Con estos datos la potestad de la implementación de estrategias para el control de la deserción es de la institución de educación superior.

Muchas de las estrategias que implementan las instituciones, están encaminadas a reducir las causas que hacen que un estudiante pueda desertar, sin embargo no se tiene una identificación precisa de cuales estudiantes puedan estar en estas condiciones antes de que la deserción se materialice. Un sistema predictivo de la deserción busca identificar de forma precisa cuales son los estudiantes que están en riesgo de desertar y sobre estos poder aplicar las estrategias de retención.

El presente trabajo pretende la creación de un sistema de alertas de la deserción estudiantil, basado en algoritmo de inteligencia artificial que sea entrenado con las condiciones particulares de cada institución que pueda predecir de forma puntual los estudiantes en alto grado de desertar y la causas que originan este evento.

² <http://www.mineduccion.gov.co/sistemasdeinformacion/1735/w3-propertyname-2895.html>

³ <http://www.icfes.gov.co/examenes/saber-11o/antecedentes>

3 Objetivos del Trabajo

3.1 Objetivo general

- Desarrollar un sistema predictivo basado en aprendizaje automático para la deserción estudiantil en instituciones de educación superior.

3.2 Objetivos específicos

- Identificar las condiciones que inciden con mayor impacto en la deserción estudiantil. Estas condiciones permitirán que el modelo creado, empiece a entrenarse y pueda reconocer nuevos patrones basado en comportamientos previos.
- Crear modelo de aprendizaje automático que permita a partir de datos previos, clasificar grupo de estudiantes en posibles desertores.
- Diseñar interfaz de usuario para la aplicación, teniendo en cuenta aspectos de usabilidad basados en principios heurísticos (Nielsen, Larry Constantine, Deborah Mayhew, entre otros).
- Crear modelo entidad-relación de la base de datos que soportará el aplicativo.
- Implementar modelo entidad-relación en sistema gestor de bases de datos MySQL.
- Realizar diagramas de casos de uso de la aplicación.
- Implementar diseño en herramienta de programación Php utilizando Framework MVC Symfony.
- Diseñar modelo de pruebas unitarias para la aplicación.
- Aplicar modelo de pruebas unitarias y realizar ajustes al desarrollo.

3.3 Objetivos secundarios

- Realizar la planeación, coordinación y seguimiento del proyecto.
- Diseñar y modelar aplicación web.
- Identificar y analizar requerimientos funcionales de usuarios.
- Presentar y sustentar resultados del trabajo investigativo.
- Estimar y controlar costos asociados al proyecto.

4 Marco Teórico

4.1 Deserción estudiantil

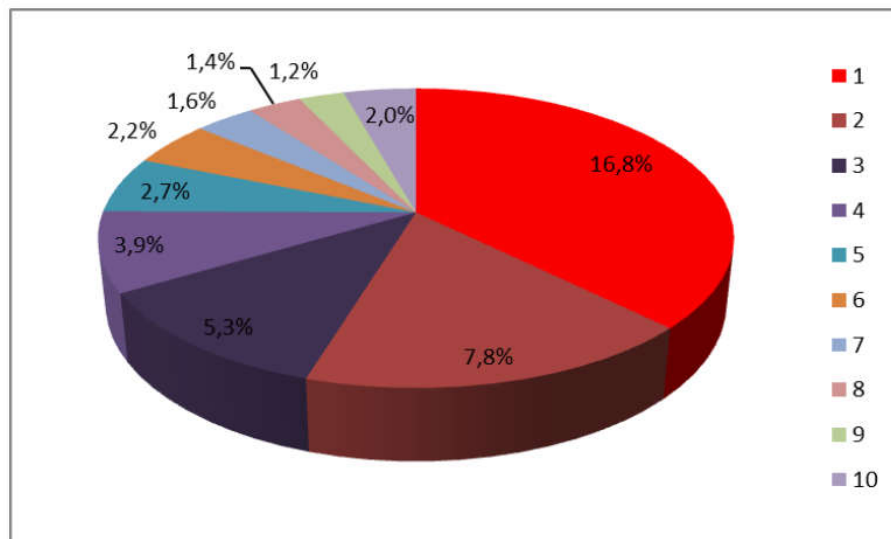
El ministerio de educación nacional de Colombia define la deserción estudiantil como el abandono del sistema académico por parte de los estudiantes, provocado por factores que se generan tanto al interior del sistema como en contextos tipo social, familiar, individual y del entorno.

Deserción en cifras

Para el año 2013, la deserción universitaria en Colombia alcanzó el 44.9% lo que indica que de cada 10 estudiantes 5 no culminan sus estudios. En los niveles técnicos y tecnológicos este porcentaje es aún mayor situándose en 62.4% y 53.8 %⁴

Los semestres en los cuales se presenta con mayor intensidad el fenómeno de la deserción son del primero al cuarto con un porcentaje del 75%.

En el siguiente gráfico se visualiza el momento en el cual porcentualmente se presenta la deserción por semestre.



Fuente: SPADIES. Abril de 2014

El análisis por áreas de conocimiento muestran que la deserción es mayor para las áreas de matemáticas y ciencias de la salud ubicándose en un

⁴ Fuente: Ministerio de Educación Nacional
http://www.mineducacion.gov.co/sistemasdeinformacion/1735/articulos-254702_archivo_pdf_estadisticas_2013.pdf

51.5% mientras que las de menor tasa son las áreas de la salud con un 39.4%

Políticas y estrategias

Teniendo presente que la deserción tiene un impacto en aspectos económicos y sociales de un país, evidenciándose en el no aprovechamiento de los recursos públicos y/o privados que se destinaron a un proyecto no culminado, el gobierno nacional ha emprendido una serie de estrategias para disminuir la deserción en instituciones de educación superior las cuales están enmarcadas en el documento “ACUERDO NACIONAL PARA DISMINUIR LA DESERCIÓN EN EDUCACIÓN SUPERIOR”

Dentro de las estrategias implementadas por el gobierno destaca el aumento de recursos financieros encaminados a fortalecer créditos educativos para estudios en instituciones de educación superior pública y privada, adicionalmente el acompañamiento en procesos y capacitaciones a instituciones que presentan mayor tasa de deserción.

Sin embargo el mayor esfuerzo se debe realizar al interior de cada institución de educación para poder reducir las tasas de deserción. En este sentido es clave poder aplicar estrategias que permitan aumentar la permanencia estudiantil y contar con herramientas de monitoreo que de forma periódica informen el comportamiento de la deserción y enfocar estrategias a los aspectos más condicionantes que impactan en la deserción de un estudiante de abandonar la institución.

Uno de los mecanismos para monitorear el proceso lo constituyen herramientas automatizadas que analicen la información y permitan la toma de decisiones. Un sistema de alertas tempranas que permitan a la institución tomar acciones antes de que se produzca el abandono por parte de estudiantes representa un factor determinante para disminuir la deserción.

4.2 Aprendizaje automático

El aprendizaje automático es una rama del campo de inteligencia artificial enfocada a desarrollar mecanismos que permitan a los sistemas de cómputo a aprender.

Mediante el aprendizaje automático se pueden realizar tareas de reconocimiento de patrones con base en información previa que ha servido de entrenamiento.

El objetivo del aprendizaje automático* para clasificación (o aprendizaje supervisado) consiste en inducir una aproximación (modelo o hipótesis) h de una función desconocida f definida desde un espacio de entrada X

hacia un espacio discreto y desordenado $Y = 1, \dots, K$, dado un conjunto de entrenamiento S .⁵

Podemos clasificar los algoritmos en función del principio de inducción que usan para adquirir los modelos o reglas de clasificación a partir de los ejemplos. Estos métodos están basados en: probabilidades, distancias, reglas o kernels.

4.2.1 Métodos probabilísticos

Los métodos estadísticos suelen estimar un conjunto de parámetros probabilísticos, que expresan la probabilidad condicionada de cada clase dadas las propiedades de un ejemplo (descrito en forma de atributos). A partir de aquí, estos parámetros pueden ser combinados para asignar las clases que maximizan sus probabilidades a nuevos ejemplos.

4.2.1.1 Naïve Bayes

El Naïve Bayes es el representante más simple de los algoritmos basados en probabilidades. Está basado en el teorema de Bayes.

El algoritmo de Naïve Bayes clasifica nuevos ejemplos $x = (x_1, \dots, x_m)$ asignándole la clase k que maximiza la probabilidad condicional de la clase dada la secuencia observada de atributos del ejemplo. Es decir,

$$\operatorname{argmax}_k P(k|x_1, \dots, x_m) = \operatorname{argmax}_k \frac{P(x_1, \dots, x_m|k)P(k)}{P(x_1, \dots, x_m)} \approx \operatorname{argmax}_k P(k) \prod_{l=1}^m P(x_l|k)$$

donde $P(k)$ y $P(x_l|k)$ se estiman a partir del conjunto de entrenamiento, utilizando las frecuencias relativas (estimación de la máxima verosimilitud*).

4.2.1.2 Máxima entropía

El algoritmo de máxima entropía apareció por primera vez en 1957 en dos artículos de E. T. Jaynes. El algoritmo de máxima entropía proporciona una manera flexible de combinar evidencias estadísticas de diferentes

⁵ Inteligencia Artificial Avanzada, Raúl Benítez, Samin Kanaan. UOC

fuentes. La estimación de probabilidades no asume conocimientos previos de los datos y se ha probado que es muy robusto.

El principio de máxima entropía consiste en definir un problema a partir de restricciones, y a partir de aquí definimos la distribución de probabilidades que sea más uniforme. Una medida matemática de la uniformidad de una distribución condicional $p(y|x)$ nos la da la entropía condicional:

$$H(p) = - \sum_{x,y} \tilde{p}(x)p(y|x) \log(p(y|x))$$

donde $\tilde{p}(x)$ corresponde a la distribución de probabilidad empírica.

4.2.2 Métodos basados en distancias

4.2.2.1 kNN

En este algoritmo la clasificación de nuevos ejemplos se realiza buscando el conjunto de los k ejemplos más cercanos de entre un conjunto de ejemplos etiquetados previamente guardados y seleccionando la clase más frecuente de entre sus etiquetas

Una parte muy importante de este método es la definición de la medida de distancia (o similitud) apropiada para el problema a tratar. Esta debería tener en cuenta la importancia relativa de cada atributo y ser eficiente computacionalmente. El tipo de combinación para escoger el resultado de entre los k ejemplos más cercanos y el valor de la propia k también son cuestiones a decidir de entre varias alternativas.

4.3 Modelo Vista Controlador

Es un patrón de arquitectura software que separa los datos, la lógica de negocio y la interfaz de usuario en una aplicación.

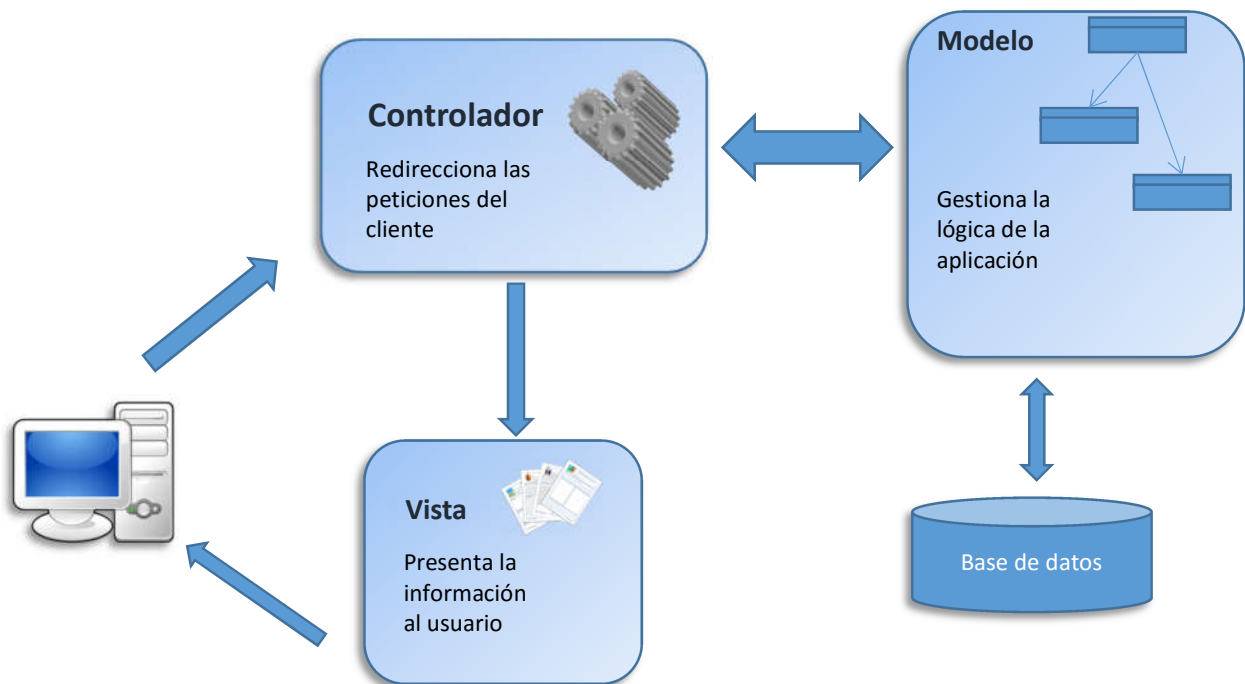
El patrón de diseño MVC organiza el código en base a su función. De hecho, este patrón separa el código en tres capas:

- La capa del modelo define la lógica de negocio (la base de datos pertenece a esta capa). Como ya sabes, Symfony guarda todas las clases y archivos relacionados con el modelo en el directorio `lib/model/`.

- La vista es lo que utilizan los usuarios para interactuar con la aplicación (los gestores de plantillas pertenecen a esta capa). En Symfony la capa de la vista está formada principalmente por plantillas en PHP. Estas plantillas se guardan en varios directorios llamados templates/ repartidos por todo el proyecto, tal y como veremos hoy mismo.

- El controlador es un bloque de código que realiza llamadas al modelo para obtener los datos y se los pasa a la vista para que los muestre al usuario. Cuando instalamos Symfony el primer día, explicamos que todas las peticiones se canalizan a través de los controladores frontales (index.php y frontend_dev.php). Estos controladores frontales realmente delegan todo el trabajo en las acciones.⁶

A continuación se presenta esquema que tendrá la aplicación haciendo uso del modelo vista controlador. El cliente realiza una petición la cual es gestionada por el controlador quien a su vez se encarga de direccionarla hacia el modelo si necesita ejecutar alguna lógica de la aplicación o a la vista si se requiere presentar información al usuario. El modelo es el encargado de realizar la lógica de la aplicación y establece conexión con el gestor de base de datos para persistir y extraer información.



⁶ http://librosweb.es/libro/jobeeet_1_4/capitulo_4/la_arquitectura_mvc.html

5 Metodología

Para la realización del presente trabajo se procederá a identificar un modelo de aprendizaje automático que será usado para la identificación de potenciales desertores.

A partir del modelo creado se iniciará la implementación de una aplicación web que permitirá gestionar todos los procesos que permitan realizar un entrenamiento del modelo y utilización del mismo para realizar la predicción de estudiantes en condiciones de deserción.

El proyecto se divide en las siguientes fases:

- 1 **Identificación modelo de aprendizaje automático a aplicar.** Se realiza un análisis de modelos probabilísticos, geométricos y lógicos que puedan ser aplicados para la clasificación de patrones y se adaptan mejor a las condiciones y variables que se tienen en cuenta para el análisis de la deserción.
- 2 **Desarrollo del modelo.** Una vez identificado al algoritmo a utilizar se procederá a programar el modelo en lenguaje de computacional.
- 3 **Entrenamiento del modelo creado.** El modelo implementado requiere un grado de entrenamiento donde empieza a conocer determinados patrones de comportamiento y son los que sirven para poder identificar patrones similares.
- 4 **Validación del modelo.** En esta última fase el modelo entra en una etapa de validación y ajuste que permitan tener un alto grado de certeza en la predicción realizada.

Para llevar a cabo el proyecto en cada una de las fases anteriormente descritas se hará uso de la metodología ágil SCRUM, la cual permite obtener resultados a en periodos de tiempo cortos y facilita el ajuste de requisitos del proyecto durante el ciclo de vida.

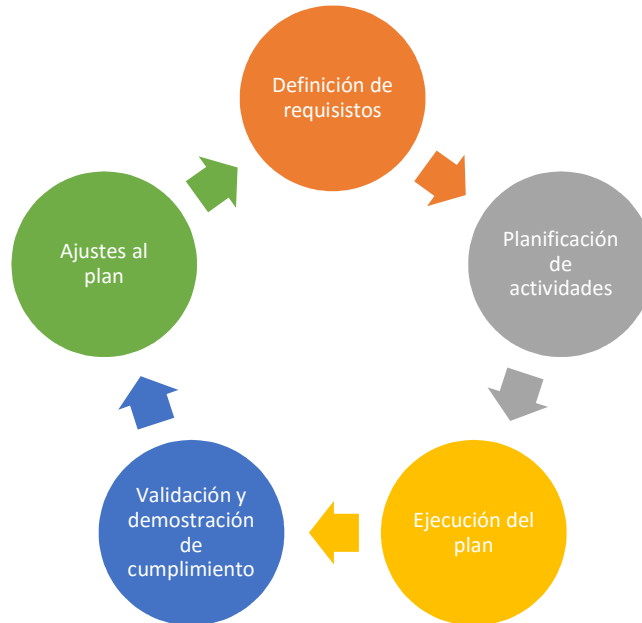
La metodología SCRUM se base en la ejecución del proyecto en bloques de tiempo establecidos. Para el caso aquí planteado estos bloques de tiempo se establecen en 1 semana.

En cada semana se desarrollarán actividades planeadas. Al final de cada semana se realizar una revisión del cumplimiento de los entregables y se realiza una nueva programación de tareas a ejecutar.

El ciclo iterativo que se realizará de acuerdo a la metodología es el siguiente:

- Definición de lista de requisitos de la aplicación.

- Planificación de la ejecución para el cumplimiento los requisitos, estimando recursos y tiempos.
- Ejecución de las actividades programadas.
- Verificación del cumplimiento de requisitos y ajustes a nueva programación



Para lograr el cumplimiento de las fechas de entrega del proyecto que debe culminar a finales del mes de junio, se establecen puntos de corte y validaciones de avances en las fechas pactadas de las PACS, las cuales son 6 de abril, 8 de mayo, 13 de junio y 24 de junio, en total se estima la duración del proyecto entre 14 y 15 semanas.

6 Arquitectura de la aplicación

La aplicación seguirá el patrón de software MVC modelo-vista-controlador, utilizando como lenguaje de programación php, sistema gestor de bases de datos Mysql y servidor web Apache.

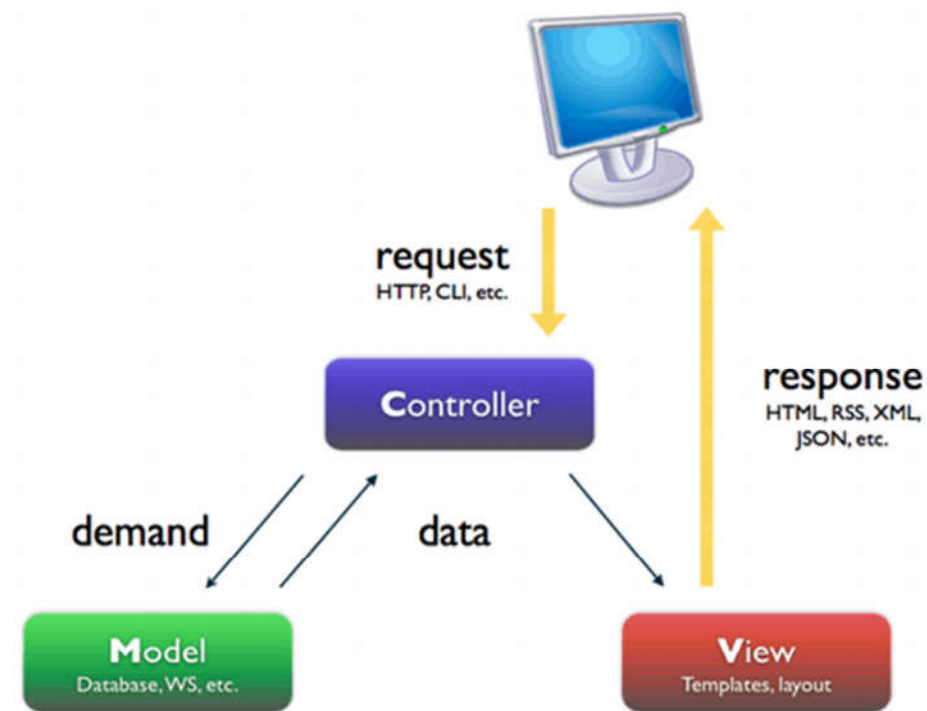


Figura 1⁷

Para facilitar la codificación e implementación se utiliza el framework symfony3⁸

El entorno de desarrollo será instalado utilizando el paquete Xampp⁹ para Windows, el cual realiza la instalación de las herramientas de desarrollo, base de datos y servidor web.

⁷ http://librosweb.es/libro/jobeeet_1_4/capitulo_4/la_arquitectura_mvc.html

⁸ <http://symfony.com/>

⁹ <https://www.apachefriends.org/es/index.html>

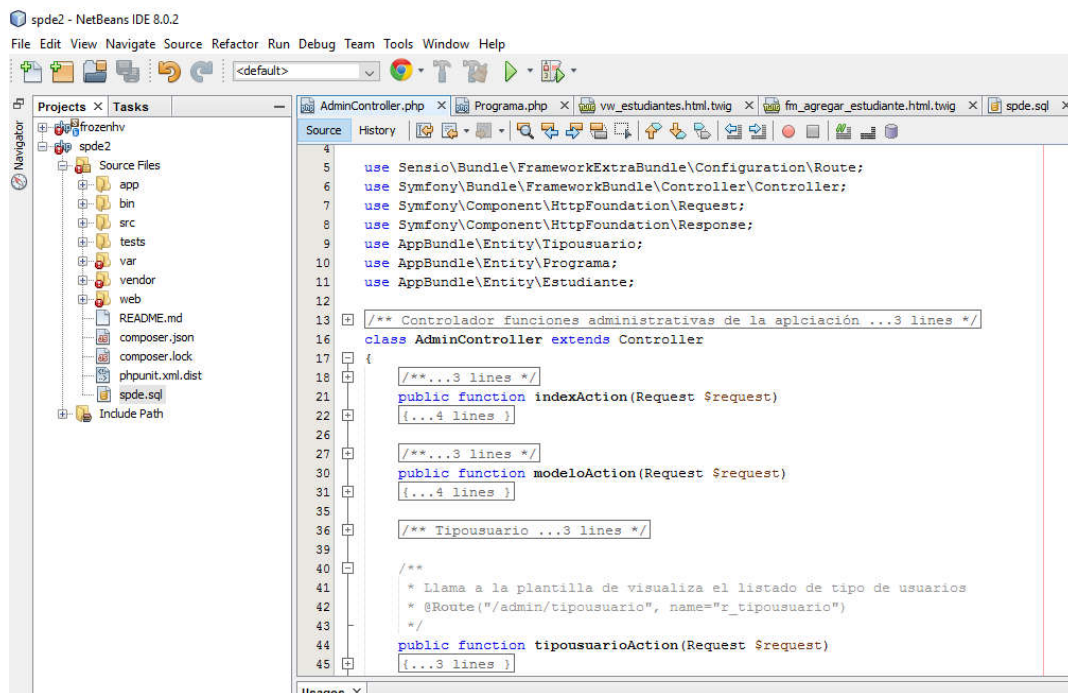
7 Plataforma de desarrollo

El lenguaje seleccionado para realizar la codificación de la aplicación es php PHP (acrónimo recursivo de PHP: Hypertext Preprocessor) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML¹⁰

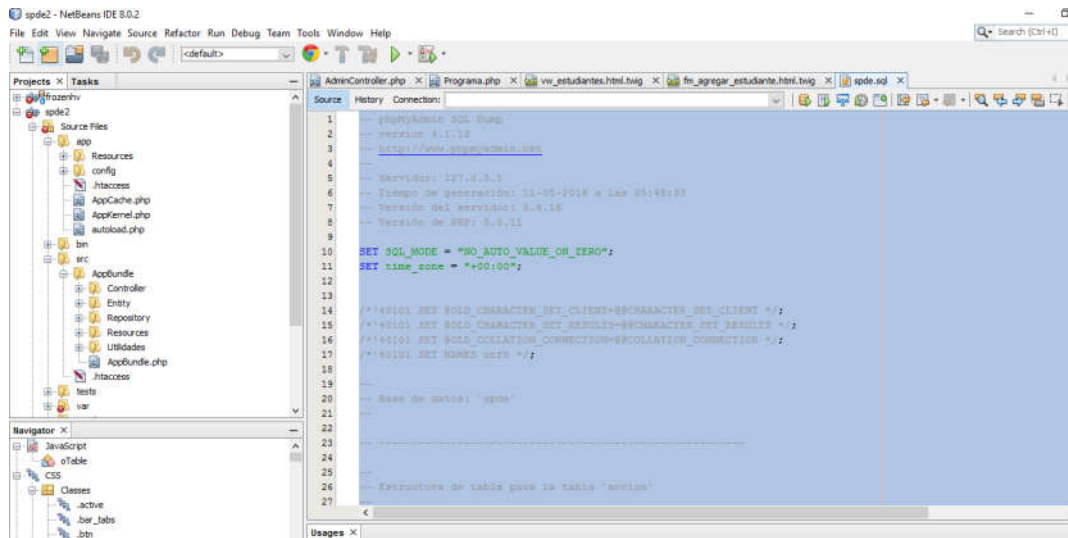
Como servidor de aplicación se utilizará apache, el cual es un servidor web de código abierto para plataformas Wnix, Windows Macintosh.

La base de datos se gestionará utilizando el servidor Mysql versión 5.6

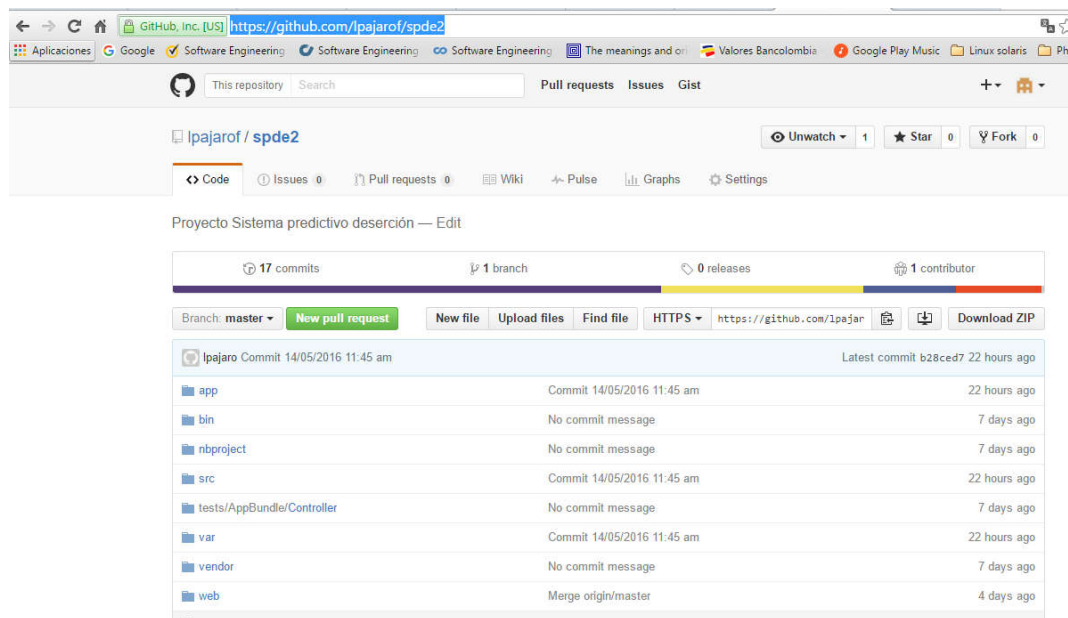
Para el desarrollo de aplicación se ha utilizado el entorno de desarrollo integrado Netbeans 8.0.1. A continuación se muestran imágenes del entorno.



¹⁰ <https://secure.php.net/manual/es/intro-what-is.php>



El control del versionamiento de la aplicación se ha hecho con la herramienta GitHub. El código fuente del proyecto está disponible para consulta en la siguiente url <https://github.com/lpajaro/spde2> donde se ha alojado el proyecto y el acceso es público. A continuación se muestra imágenes del repositorio.



8 Implementación algoritmo Naïve Bayes

El algoritmo Naïve Bayes se enfoca en la clasificación de un individuo teniendo en cuenta determinadas características, dado un conjunto previo de datos usados como entrenamiento. Se basa en un modelo probabilístico para seleccionar el individuo con la probabilidad de ocurrencia más alta.

A continuación se describen los pasos para la implementación en la aplicación.

1. Seleccionar el conjunto de datos de entrenamiento.
Los datos de entrenamiento se cargan en la tabla *entrenamiento*.

identrenamiento	desertor	c1	c2	c3	c4	idestudiante
1	1	6	0.1	0.2	1	202
2	1	5.92	0.05	0.266667	1	203
3	1	5.58	0.15	0.2	1	204
4	1	5.92	0.175	0.333333	1	205
9	0	5	0.5	0.6	1	210
10	0	5.5	0.25	0.466667	1	211
11	0	5.42	0.35	0.533333	1	212
46	0	5.75	0.25	0.4	1	366

2. Realizar cálculos de las probabilidades de cada clase (desertor y no desertor), así como la media μ y la varianza σ para aplicación de la fórmula

$$P(x = v|c) = \frac{1}{\sqrt{2\pi\sigma_c^2}} \exp\left(-\frac{(v - \mu_c)^2}{2\sigma_c^2}\right)$$

Donde x corresponde al atributo ($c1$, $c2$, $c3$ o $c4$), v el valor, c la clase (desertor o no desertor), μ al promedio y σ^2 desviación estándar de cada clase. La anterior fórmula representa la probabilidad de ocurrencia de cada atributo de cada clase.

Los valores de cada variable de la fórmula se realizan en la vista *calmodelo* de la base de datos.

id_desertor	probabilidad	mediaC1	varianzac1	mediac2	varianzac2	mediac3	varianzac3	mediac4	varianzac4
1 Desertor	0.5000	5.855000019073486	0.035033353932703876	0.1187500013038516	0.0030729166263093687	0.2500000074505806	0.004074074581817373	1	0
0 No Desertor	0.5000	5.417500019073486	0.09722500012715807	0.3374999985098839	0.013958333283662805	0.5000000149011612	0.007407409173471122	1	0

```
SELECT 1 AS `ID`, IF(`SPDE`.`ENTRENAMIENTO`.`DESERTOR` =
'1'), 'DESERTOR', `SPDE`.`ENTRENAMIENTO`.`DESERTOR`) AS
`DESERTOR`, (COUNT(`SPDE`.`ENTRENAMIENTO`.`IDENTRENAMIENTO`) /
(SELECT COUNT(`SPDE`.`ENTRENAMIENTO`.`IDENTRENAMIENTO`) FROM
`SPDE`.`ENTRENAMIENTO`)) AS
`PROBABILIDAD`, AVG(`SPDE`.`ENTRENAMIENTO`.`C1`) AS
```



```

`MEDIAC1`,VAR_SAMP(`SPDE`.`ENTRENAMIENTO`.`C1`) AS
`VARIANZAC1`,`AVG(`SPDE`.`ENTRENAMIENTO`.`C2`) AS
`MEDIAC2`,`VAR_SAMP(`SPDE`.`ENTRENAMIENTO`.`C2`) AS
`VARIANZAC2`,`AVG(`SPDE`.`ENTRENAMIENTO`.`C3`) AS
`MEDIAC3`,`VAR_SAMP(`SPDE`.`ENTRENAMIENTO`.`C3`) AS
`VARIANZAC3`,`AVG(`SPDE`.`ENTRENAMIENTO`.`C4`) AS
`MEDIAC4`,`VAR_SAMP(`SPDE`.`ENTRENAMIENTO`.`C4`) AS `VARIANZAC4`
FROM `SPDE`.`ENTRENAMIENTO` WHERE
(`SPDE`.`ENTRENAMIENTO`.`DESERTOR` = 1)
UNION
SELECT 0 AS `ID`,`IF((`SPDE`.`ENTRENAMIENTO`.`DESERTOR` =
'0'),'NO `DESERTOR`,`SPDE`.`ENTRENAMIENTO`.`DESERTOR`) AS
`DESERTOR`,`(COUNT(`SPDE`.`ENTRENAMIENTO`.`IDENTRENAMIENTO`)
/(SELECT COUNT(`SPDE`.`ENTRENAMIENTO`.`IDENTRENAMIENTO`) FROM
`SPDE`.`ENTRENAMIENTO`)) AS
`PROBABILIDAD`,`AVG(`SPDE`.`ENTRENAMIENTO`.`C1`) AS
`MEDIAC1`,`VAR_SAMP(`SPDE`.`ENTRENAMIENTO`.`C1`) AS
`VARIANZAC1`,`AVG(`SPDE`.`ENTRENAMIENTO`.`C2`) AS
`MEDIAC2`,`VAR_SAMP(`SPDE`.`ENTRENAMIENTO`.`C2`) AS
`VARIANZAC2`,`AVG(`SPDE`.`ENTRENAMIENTO`.`C3`) AS
`MEDIAC3`,`VAR_SAMP(`SPDE`.`ENTRENAMIENTO`.`C3`) AS
`VARIANZAC3`,`AVG(`SPDE`.`ENTRENAMIENTO`.`C4`) AS
`MEDIAC4`,`VAR_SAMP(`SPDE`.`ENTRENAMIENTO`.`C4`) AS `VARIANZAC4`
FROM `SPDE`.`ENTRENAMIENTO` WHERE
(`SPDE`.`ENTRENAMIENTO`.`DESERTOR` = 0)

```

Esta vista es la parte central del modelo ya que contiene todos los valores para la realización de los cálculos de las probabilidades

3. Clasificación del individuo. Para este paso se ha creado un controlador el cual recibe como parámetros los valores del individuo que se quiere clasificar y realiza llamado a métodos que realizan los cálculos identificando la probabilidad de la clase con mayor valor. A continuación se muestra fragmento de código y explicación de cada uno.

Controlador *ModeloController*

```

/**
 * Agrega un nuevo registro a la tabla clasificacion
 * @Route("/admin/clasificacion/clasificacionAdd",name="clasificacionAdd")
 */
public function clasificacionAddAction(Request $request)
{
    try{
        $em = $this->getDoctrine()->getManager();
        $v_clasificacion = new Clasificacion();

        $v_estudiante = $em->getRepository('AppBundle:Estudiante')->
        >findOneBy(array('idestudiante'=>$request->request->get('idestudiante')));

        $v_clasificacion->setC1($request->request->get('c1'));
        $v_clasificacion->setC2($request->request->get('c2'));
        $v_clasificacion->setC3($request->request->get('c3'));
        $v_clasificacion->setC4($request->request->get('c4'));
        $v_clasificacion->setIdestudiante($v_estudiante);

        $c1=$request->request->get('c1');
        $c2=$request->request->get('c2');
        $c3=$request->request->get('c3');
        $c4=$request->request->get('c4');
        /**
         * Toma los datos del estudiante y determina si es posible desertor o
         * no.
         * Selecciona los datos para clasificar como desertor
         */
        $v_calmodeloS = $this->getDoctrine()

```

```

        ->getRepository("AppBundle:Calmodelo")
        ->find(1);
    /**
     * Selecciona los datos para clasificar como no desertor
     */
    $v_calmodeloN = $this->getDoctrine()
        ->getRepository("AppBundle:Calmodelo")
        ->find(0);
    /**
     * Compara las probabilidades de ser un desertor y no desertor.
     * Se elige la mayor probabilidad para la clasificacion
     */
    if ($v_calmodeloS->priori($c1,$c2,$c3,$c4) > $v_calmodeloN-
>priori($c1,$c2,$c3,$c4))
    {
        $v_respuesta=' DESERTOR ';
        $v_clasificacion->setDesertor(1);
    }
    else{
        $v_respuesta=' NO ES UN DESERTOR ';
        $v_clasificacion->setDesertor(0);
    }
    $em->persist($v_clasificacion);
    $em->flush();

    return New Response('<div class="alert alert-success alert-
dismissible fade in" role="alert">'
        . $v_respuesta
        . 'Los datos han sido agregados
satisfactoriamente'
        . '</div>');
    } catch (\Exception $e) {
        switch (get_class($e)) {
            case
'Doctrine\DBAL\Exception\UniqueConstraintViolationException':
                return new Response('Registro Existente<br>DBAL
Exception<br>');

            case 'Doctrine\DBA\DBAException':
                return new Response('DBA Exception<br/>');

            case 'PDOException':
                return new Response('DBA Exception<br/>' . $e->getMessage());

            default:
                return new Response($e->getMessage() . '--' . get_class($e));
        }
    }
}
}

```

Entidad *Calmodelo*. Se muestra método principal que realiza el cálculo de probabilidad del individuo que se desea clasificar

```

/**
 * Función que devuelve la probabilidad de un atributo del cual se quiere
 * conocer su clasificación
 */
public function priori($atributo1,$atributo2,$atributo3,$atributo4)
{
    $probabilidad1 = (1/(sqrt(2*pi()*$this->varianzaC1)))*exp(-
1*pow(($atributo1 - $this->mediaC1),2)/(2*$this->varianzaC1));
    $probabilidad2 = (1/(sqrt(2*pi()*$this->varianzaC2)))*exp(-
1*pow(($atributo2 - $this->mediaC2),2)/(2*$this->varianzaC2));
    $probabilidad3 = (1/(sqrt(2*pi()*$this->varianzaC3)))*exp(-
1*pow(($atributo3 - $this->mediaC3),2)/(2*$this->varianzaC3));
    $probabilidad4 = (1/(sqrt(2*pi()*$this->varianzaC4)))*exp(-
1*pow(($atributo4 - $this->mediaC4),2)/(2*$this->varianzaC4));
    return ($probabilidad1*$probabilidad2*$probabilidad3*$probabilidad4*$this-
>probabilidad);
}

```

9 Guía de instalación

Los requisitos para el despliegue de la aplicación son servidor web con soporte php y gestor de base de datos mysql. A continuación se describen los pasos para realizar la instalación en ambiente Windows.

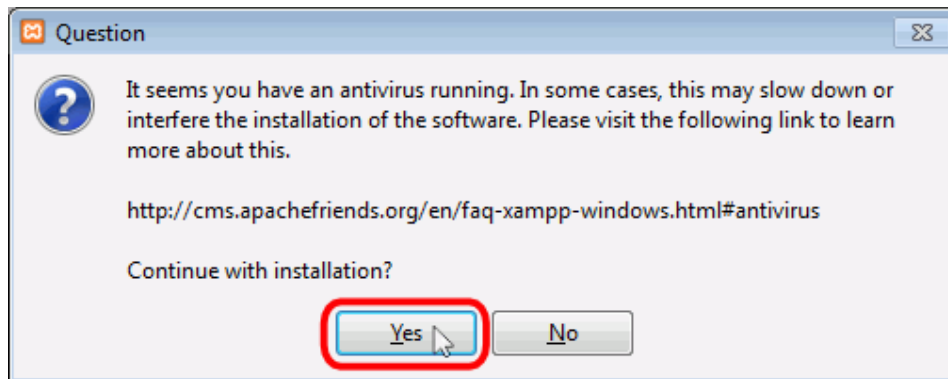
9.1 Instalación Xampp

La instalación del servidor apache con soporte para php y el gestor de base de datos Mysql se hacen con el instalador de la utilidad Xampp. A continuación se describen los pasos (las imágenes de la instalación son tomadas de la siguiente url

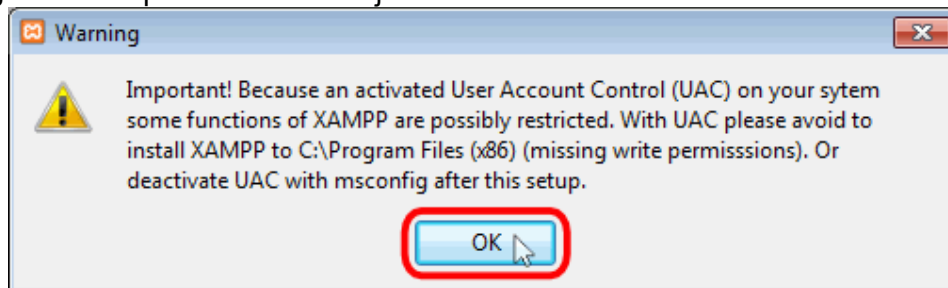
http://www.mclibre.org/consultar/php/otros/in_php_instalacion.html).

La descarga del instalador Xampp se realiza desde la siguiente ubicación <https://www.apachefriends.org/es/index.html>

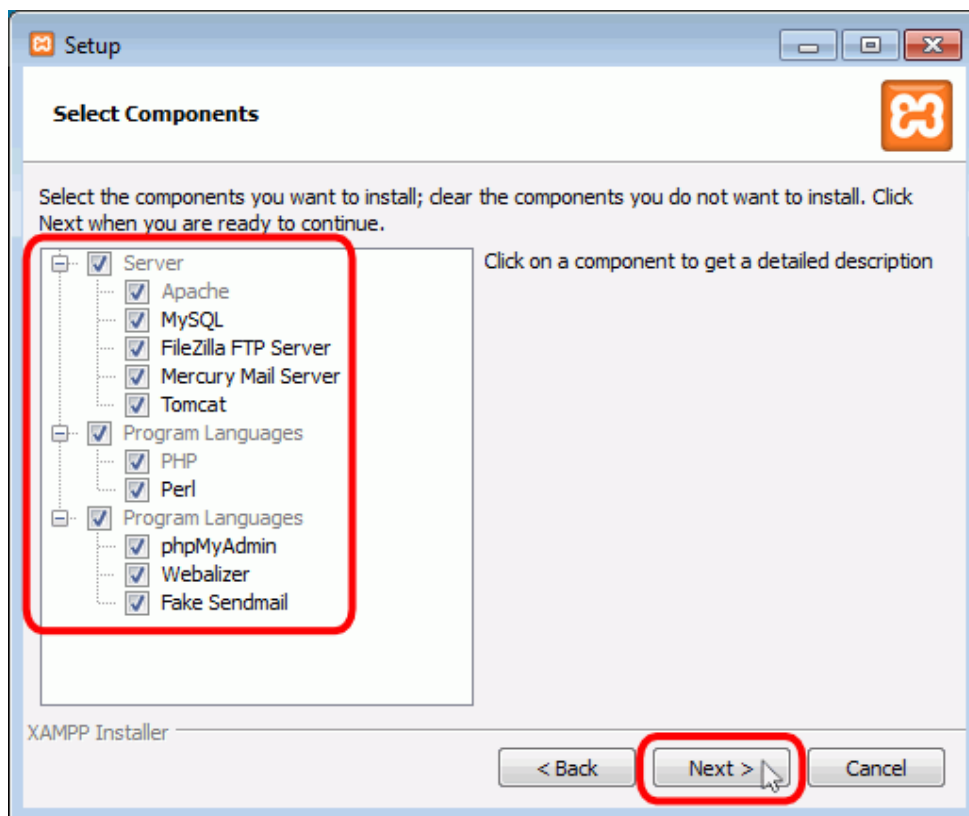
Una vez descargado el instalador para ambiente Windows procedemos a ejecutarlo con lo cual nos muestra la siguiente advertencia en caso de tener un antivirus:

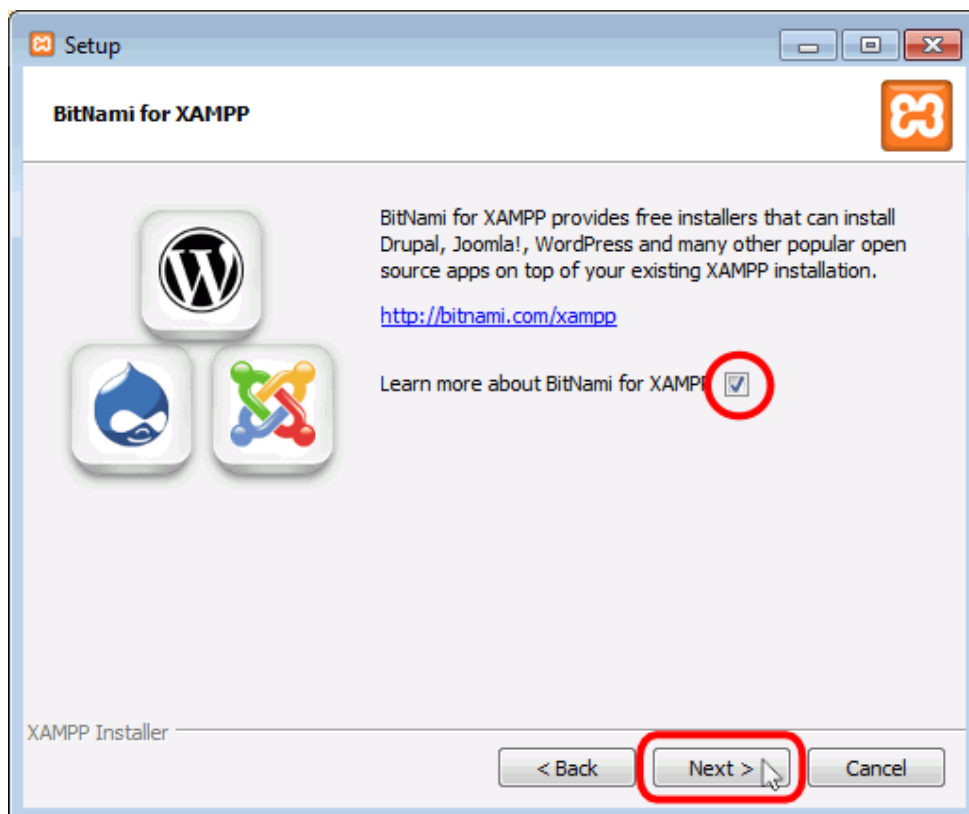
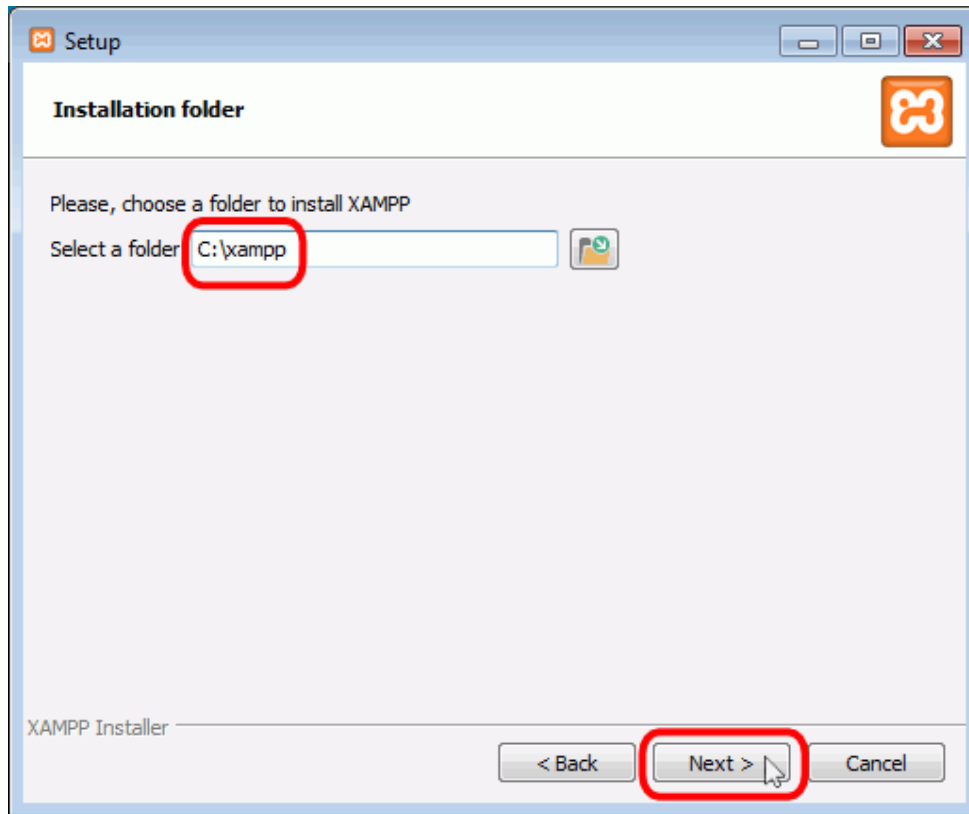


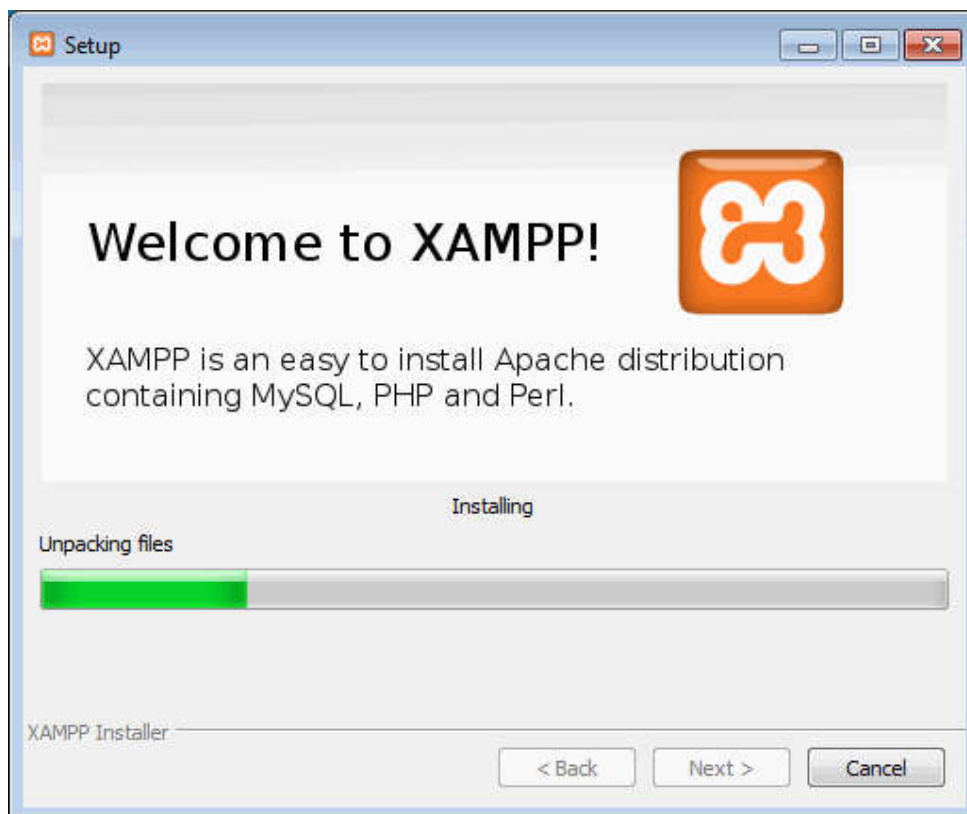
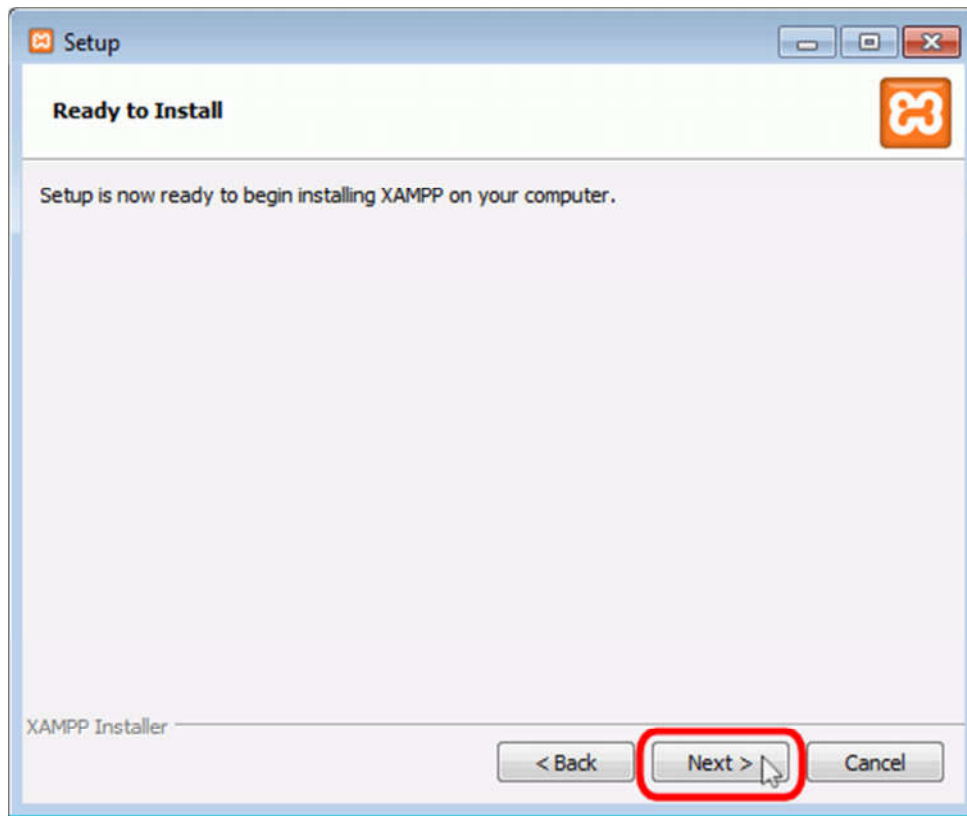
Seguido nos aparece un mensaje relacionado con el control de cuentas

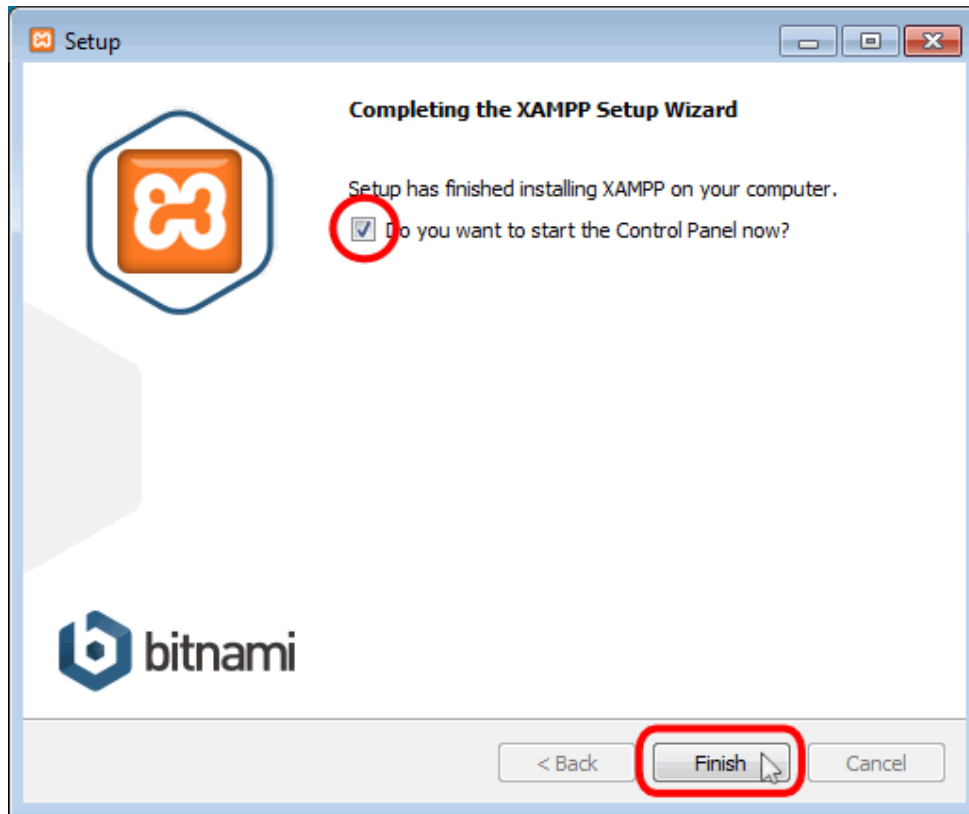


Confirmado estos dos primeros mensajes inicia el proceso de instalación de Xampp

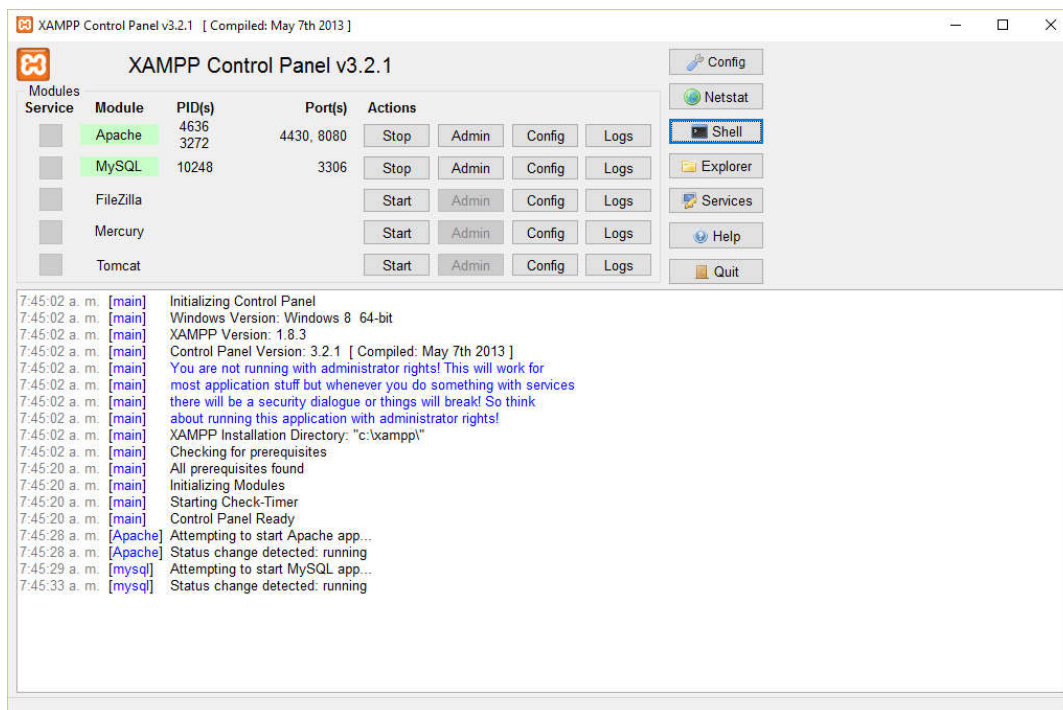








Una vez completada la instalación se procede a iniciar a través del panel de control los servidores Apache y MySQL con los cuales tendremos el entorno necesario para el despliegue de la aplicación.



9.2 Instalación base de datos

Para la creación de la base de datos de la aplicación se debe ejecutar el script `spde.sql`, el cual se encuentra en el repositorio del proyecto.

En la línea de comandos del gestor de base de datos Mysql se crea la base de datos `spde`, mediante el siguiente comando:

```
create database spde;
```

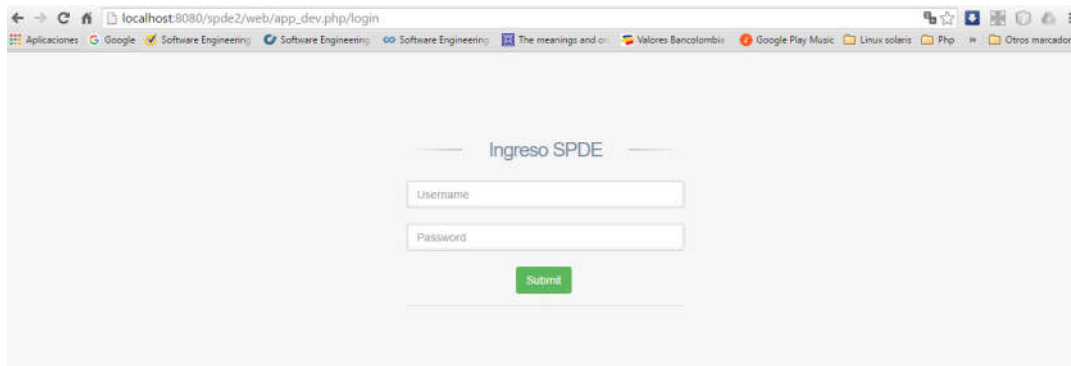
Con la base de datos creada se carga el archivo `spde.sql` encargado de la creación de objetos:

```
source spde.sql;
```

9.3 Instalación aplicación

Una vez completada la etapa de requerimientos básicos se procede a la instalación de la aplicación, para lo cual se debe copiar la carpeta `spde2`, la cual contiene los archivos fuentes, en directorio de despliegue de `xampp htdocs`.

Realizado el paso anterior se procede a ejecutar la aplicación a través de navegador web. A continuación se muestra imagen de despliegue de la aplicación.



10 Planificación

A continuación se describen los procesos a desarrollar para la planeación del proyecto.

10.1 Definición del alcance

Consiste en describir de forma general los requisitos funcionales y técnicos que deberá satisfacer la aplicación. Como resultado de esta fase se deberá contar con una especificación clara del producto que se espera obtener, así como los objetivos del proyecto, los entregables, criterios de aceptación y restricciones del proyecto.

10.2 Creación de la EDT

La estructura detallada de trabajo EDT consiste en la identificación de cada una de las actividades a desarrollar durante el proyecto. Cada una de estas actividades deberá tener una descripción, duración, recursos a utilizar, costos y entregables de cada actividad.

10.3 Realización del cronograma

Con la lista de actividades se procede a la realización de un cronograma teniendo en cuenta la secuenciación de actividades, recursos y limitaciones del proyecto.

10.4 Definición de criterios de aceptación

Esta etapa consiste en la definición de los criterios mínimos de aceptación para cada uno de los entregables.

10.5 Establecimiento de organigrama del proyecto

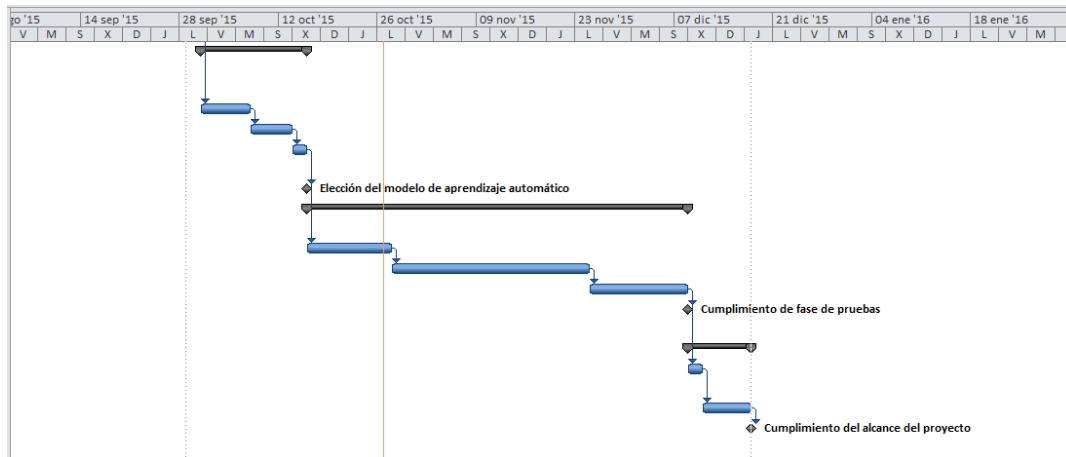
Se realizará la definición de roles del proyecto, asignación de responsabilidades a personal involucrado en el desarrollo de cada una de las actividades.

10.6 Realizar un plan de comunicación

Se desarrollará un plan para informar del avance del proyecto así como el tipo de información que se compartirá entre los involucrados en el proyecto y los responsables de la generación divulgación almacenamiento y custodia.

10.7 Cronograma del proyecto

Nombre de tarea	Duración	Comienzo	Fin
Sistema predictivo de la deserción	58 días	mar 29/09/15	jue 17/12/15
Planificación del proyecto	2 días	mar 29/09/15	mié 30/09/15
Aprobación del alcance y plan del proyecto	0 días	mié 30/09/15	mié 30/09/15
Identificación de algoritmo de IA a utilizar	11 días	jue 01/10/15	jue 15/10/15
Análisis de los algoritmos más comunes de inteligencia artificial	5 días	jue 01/10/15	mié 07/10/15
Comparación de los algoritmos seleccionados	4 días	jue 08/10/15	mar 13/10/15
Elección de algoritmo a utilizar	2 días	mié 14/10/15	jue 15/10/15
Elección del modelo de aprendizaje automático	0 días	jue 15/10/15	jue 15/10/15
Implementación de modelo	38 días	vie 16/10/15	mar 08/12/15
Diseño de modelo	8 días	vie 16/10/15	mar 27/10/15
Codificación en lenguaje de programación	20 días	mié 28/10/15	mar 24/11/15
Pruebas	10 días	mié 25/11/15	mar 08/12/15
Cumplimiento de fase de pruebas	0 días	mar 08/12/15	mar 08/12/15
Entrenamiento y validación del modelo	7 días	mié 09/12/15	jue 17/12/15
Entrenamiento del modelo	2 días	mié 09/12/15	jue 10/12/15
Validación y ajustes	5 días	vie 11/12/15	jue 17/12/15
Cumplimiento del alcance del proyecto	0 días	jue 17/12/15	jue 17/12/15



10.8 Seguimiento proyecto

Para el seguimiento de actividades del proyecto se ha utilizado la herramienta de gestión de proyectos basada en Scrum Target Process¹¹, en la cual se han

¹¹ <https://www.targetprocess.com/>

creado las distintas historias de usuario y se realiza el seguimiento y cumplimiento de las mismas a continuación se muestra el plan de iteración de las historias de usuario creadas

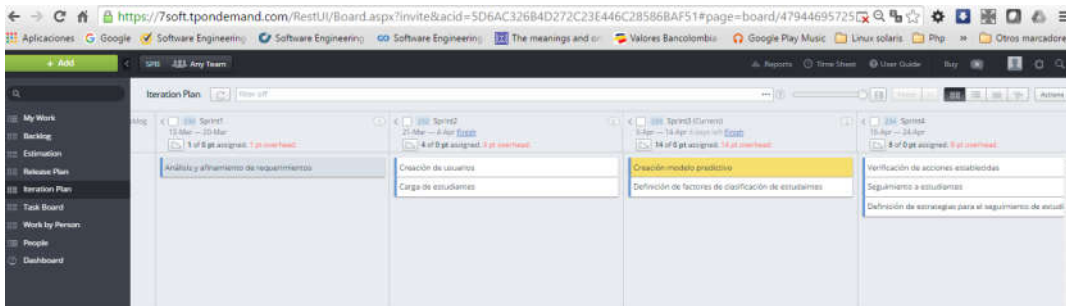


Figura 2 Seguimiento proyecto

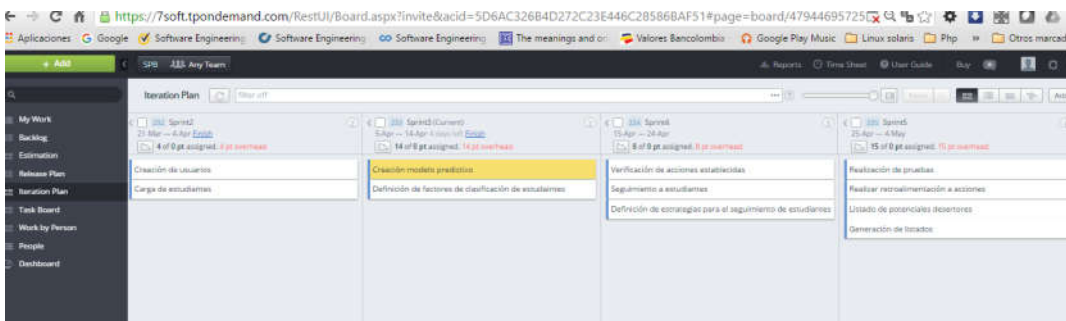


Figura 3 Seguimiento proyecto

11 Requerimientos

Los requerimientos se plantean a través de historias de usuarios de acuerdo a la metodología SCRUM

Id	Rol	Descripción	Razón/Resultado
H1	Administrador	Se requiere poder cargar información de estudiantes que han desertado de la institución y estudiantes activos con los factores de deserción característicos de cada grupo. Estos factores pueden ser promedio académico, semestre actual, forma de pago de semestre	Este cargue de datos es con el objetivo de poder ir entrenando al sistema
H2	Administrador	Crear usuarios que puedan acceder al sistema y asignarles un rol	Esto se realiza con el fin de que otros usuarios puedan ingresar a la aplicación
H3	Administrador	Se deberá poder definir factores para clasificar a estudiantes	Esto permitirá que el modelo se pueda ir adaptando a nuevas condiciones
H4	Coordinador	Podrá generar listado de estudiantes cargados, diferenciados por desertores y no desertores	Validar la información cargada en el sistema
H5	Coordinador	Establecer estrategias de seguimiento de estudiantes que han sido catalogados como desertores	Permite que se le asignen al estudiante diferentes acciones y su respectivo seguimiento para asegurar su permanencia
H6	Coordinador	Realizar seguimiento a acciones que se establezcan para mantener a un estudiante dentro del sistema académico.	Monitorear el cumplimiento de las acciones establecidas
H7	Estudiante	Podrá ingresar al sistema para verificar las acciones establecidas	Verificar las acciones que le han sido asignadas
H8	Estudiante	Podrá realizar una retroalimentación a las acciones que le han sido asignadas	Dar cumplimiento a las acciones que le ha sido asignadas
H9	Coordinador	Generar listado consolidado de posibles estudiantes desertores	Necesario para poder implementar acciones

12 Casos de usos

Los siguientes son los casos de uso relacionados a los procesos que deberá realizar la aplicación.

Creación de usuario	
ID	1
Nombre	Crear usuario sistema
Descripción	La aplicación deberá permitir la creación de usuario al sistema con un rol específico
Autor	Leandro Pájaro Fuentes
Actores	Administrador del sistema
Precondiciones	El usuario a crear no debe existir, el sistema validará el documento de identidad. El administrador del sistema deberá estar autenticado en la aplicación
Flujo normal	El administrador del sistema ingresa al módulo de creación de usuario y registra la siguiente información: <ul style="list-style-type: none"> • Documento de identidad • Nombres • Apellidos • Email • Rol • Estado • Usuario • Contraseña
Post condición	El usuario queda activo en el sistema
Excepciones	En caso de que el documento de identidad o el usuario existan el sistema validará y el usuario podrá cancelar la operación terminando el caso de uso. En caso de no disponer de algunos valores requerido el caso de uso termina.

Carga información de estudiantes	
ID	2
Nombre	Cargar estudiantes al sistema
Descripción	Permite la carga masiva de estudiantes
Autor	Leandro Pájaro Fuentes
Actores	Administrador del sistema
Precondiciones	El administrador del sistema deberá estar autenticado en la aplicación Archivo de texto con el formato establecido para la realización de la carga de datos.
Flujo normal	Se selecciona archivo a cargar y se procede a ejecutar validación de los datos
Post condición	Lo estudiantes y factores asociados quedan cargados en el sistema
Excepciones	En caso de que el archivo a cargar no contenga la estructura y datos correctos, el sistema no procederá a realizar la carga de los datos. El usuario podrá cancelar el proceso de cargue

Creación de acciones	
ID	3
Nombre	Creación de acciones para mitigar la deserción
Descripción	Este proceso consiste en crear acciones a establecer a estudiantes para minimizar los riesgos asociados a la deserción
Autor	Leandro Pájaro Fuentes
Actores	Coordinador
Precondiciones	El coordinador deberá estar autenticado en el sistema
Flujo normal	El coordinador define el tipo de acción a implementar, establece las condiciones o criterios de cumplimiento y los tiempos en los cuales se valida el cumplimiento
Post condición	Las acciones a implementar quedan registradas en el sistema
Excepciones	El usuario podrá desistir de la creación de las acciones

Creación de acciones	
ID	4
Nombre	Realizar seguimiento de estrategias
Descripción	El coordinador podrá ver el estado en el cual se encuentran las estrategias implementadas, el cumplimiento y resultado de las mismas
Autor	Leandro Pájaro Fuentes
Actores	Coordinador
Precondiciones	El coordinador deberá estar autenticado en el sistema
Flujo normal	El coordinador selecciona estrategias para visualizar el estado en el cual se encuentra.
Post condición	
Excepciones	

13 Historias de usuario

Las historias de usuario han sido registradas en la herramienta de gestión de proyectos online TargetProcess y sobre la cual se realiza toda la gestión de proyecto a continuación se detallan las historias de usuario creadas:

The screenshot displays a user story card in TargetProcess. The card is titled '#231 Análisis y afinamiento de requerimientos'. The description reads: 'Analizar y especificar en detalles los requerimientos del proyecto, determinando herramientas y metodología a seguir'. The card is assigned to the 'Desarrollo' team. The progress bar shows 100% completion. The state is 'Done'. The effort is 1 pt, assigned to 'Leandro2 Pajaro2' and 'Leandro Pájaro'. The QA Engineer role is also listed. The left sidebar shows a backlog of other user stories, with 'Análisis y afinamiento de requerimientos' at the top.

The screenshot displays a user story card in TargetProcess. The card is titled '#228 Realización de pruebas'. The description reads: 'Realizar pruebas al modelo y realizar ajustes'. The card is assigned to the 'Desarrollo' team. The progress bar shows 0% completion. The state is 'Open'. The effort is 5 pt, assigned to 'Leandro Pájaro' and 'QA Engineer'. The QA Engineer role is also listed. The left sidebar shows a backlog of other user stories, with 'Realización de pruebas' at the top.

Backlog filter off

USER STORY #227 Creación modelo predictivo Follow Share link

Description Tasks 0 Bugs 0 Tests Relations 0 More Actions +

Funcional Add tag Tags Cloud Edit description

Creación e modelo basado en algoritmo inteligencia artificial que clasifique a los estudiantes en desertores y no desertores

Drag and drop files here or click to select files

Comments Add comment

Spent 0h Remain 0h

State Open Planned In Progress In Testing Done

Teams Desarrollo Cycle time

Assignments Effort

Developer	13 pt
Leandro2 Pajaro2	
QA Engineer	
Leandro2 Pajaro2	
Total	13 pt

Info

Backlog filter off

USER STORY #226 Realizar retroalimentación a acciones Follow Share link

Description Tasks 0 Bugs 0 Tests Relations 0 More Actions +

Estudiantes Add tag Tags Cloud Edit description

Podrá realizar una retroalimentación a las acciones que le han sido asignadas.
Dar cumplimiento a las acciones que le ha sido asignadas

Drag and drop files here or click to select files

Comments Add comment

Spent 0h Remain 0h

State Open Planned In Progress In Testing Done

Teams Desarrollo Cycle time

Assignments Effort

Developer	8 pt
Leandro Pájaro	
QA Engineer	
Total	8 pt

Info Sistema Predictivo Basado En Aprendizaje

Backlog filter off

USER STORY #225 Verificación de acciones establecidas Follow Share link

Description Tasks 0 Bugs 0 Tests Relations 0 More Actions +

Estudiantes Add tag Tags Cloud Edit description

Estudiante podrá ingresar al sistema para verificar las acciones establecidas.
Verificar las acciones que le han sido asignadas

Drag and drop files here or click to select files

Comments Add comment

Spent 0h Remain 0h

State Open Planned In Progress In Testing Done

Teams Desarrollo Cycle time

Assignments Effort

Developer	2 pt
Leandro2 Pajaro2	
QA Engineer	
Total	2 pt

Info Sistema Predictivo

Backlog filter off

USER STORY #224 Listado de potenciales desertores

Description Tasks Bugs Tests Relations More Actions

Coordinación Add tag Tags Cloud

Generar listado consolidado de posibles estudiantes desertores.
Necesario para poder implementar acciones

Drag and drop files here or click to select files

Comments
+ Add comment

Spent 0h Remain 0h

State
Open
Planned
In Progress
In Testing
Done

Teams Desarrollo Cycle time

Assignments
Effort
Developer 1 pt
Leandro2 Pajaro2
QA Engineer
Total 1 pt

Info
Sistema Predictivo
Basado En Aprendizaje

Backlog filter off

USER STORY #223 Seguimiento a estudiantes

Description Tasks Bugs Tests Relations More Actions

Coordinación Add tag Tags Cloud

Realizar seguimiento a acciones que se establezcan para mantener a un estudiante dentro del sistema académico.
Monitorear el cumplimiento de las acciones establecidas

Drag and drop files here or click to select files

Comments
+ Add comment

Spent 0h Remain 0h

State
Open
Planned
In Progress
In Testing
Done

Teams Desarrollo Cycle time

Assignments
Effort
Developer 3 pt
Leandro Pájaro
QA Engineer
Total 3 pt

Info
Sistema Predictivo
Basado En Aprendizaje

Backlog filter off

USER STORY #222 Definición de estrategias para el seguimiento de estudiantes

Description Tasks Bugs Tests Relations More Actions

Coordinación Add tag Tags Cloud

Establecer estrategias de seguimiento de estudiantes que han sido catalogados como desertores.
Permite que se le asignen al estudiante diferentes acciones y su respectivo seguimiento para asegurar su permanencia

Drag and drop files here or click to select files

Comments
+ Add comment

Spent 0h Remain 0h

State
Open
Planned
In Progress
In Testing
Done

Teams Desarrollo Cycle time

Assignments
Effort
Developer 3 pt
Leandro2 Pajaro2
Leandro Pájaro
QA Engineer
Total 3 pt

Info

Backlog filter off

USER STORY #221 Generación de listados Follow Share link

Description Tasks 0 Bugs 0 Tests Relations 0 More Actions +

Coordinación Add tag Tags Cloud

Podrá generar listado de estudiantes cargados, diferenciados por desertores y no desertores.

Validar la información cargada en el sistema

Drag and drop files here or click to select files

Comments + Add comment

Spent 0h Remain 0h

State: Open

Planned
In Progress
In Testing
Done

Teams Desarrollo Cycle time

Assignments Effort

Developer	1 pt
Leandro2 Pajaro2	
QA Engineer	
Total	1 pt

Info Sistema Predictivo Basado En Aprendizaje

Backlog filter off

USER STORY #220 Definición de factores de clasificación de estudiantes Follow Share link

Description Tasks 0 Bugs 0 Tests Relations 0 More Actions +

Administración Add tag Tags Cloud

Se deberá poder definir factores para clasificar a estudiantes.

Esto permitirá que el modelo se pueda ir adaptando a nuevas condiciones

Drag and drop files here or click to select files

Comments + Add comment

Spent 0h Remain 0h

State: Open

Planned
In Progress
In Testing
Done

Teams Desarrollo Cycle time

Assignments Effort

Developer	1 pt
Leandro2 Pajaro2	
QA Engineer	
Total	1 pt

Info Sistema Predictivo

Backlog filter off

USER STORY #219 Creación de usuarios Follow Share link

Description Tasks 0 Bugs 0 Tests Relations 0 More Actions +

Administración Add tag Tags Cloud

Crear usuarios que puedan acceder al sistema y asignarles un rol.

Esto se realiza con el fin de que otros usuarios puedan ingresar a la aplicación

Drag and drop files here or click to select files

Comments + Add comment

Spent 5h Remain 10h 34%

State: Planned

Open
Planned
In Progress
In Testing
Done

Teams Desarrollo Cycle time

Assignments Effort

Developer	2 pt
Leandro Pájaro	
QA Engineer	
Total	2 pt

Info Sistema Predictivo Basado En Aprendizaje

Backlog filter off

USER STORY #218 Carga de estudiantes Follow Share link

Description Tasks 0 Bugs 0 Tests Relations 0 More Actions +

Administración Add tag Tags Cloud Edit description

Se requiere poder cargar información de estudiantes que han desertado de la institución y estudiantes activos con los factores de deserción característicos de cada grupo. Estos factores pueden ser promedio académico, semestre actual, forma de pago de semestre.

Este cargue de datos es con el objetivo de poder ir entrenando al sistema

Drag and drop files here or click to select files

Comments + Add comment

Spent 0h Remain 20h

State

- Open
- Planned
- In Progress**
- In Testing
- Done

Teams Desarrollo Cycle time

Assignments Effort

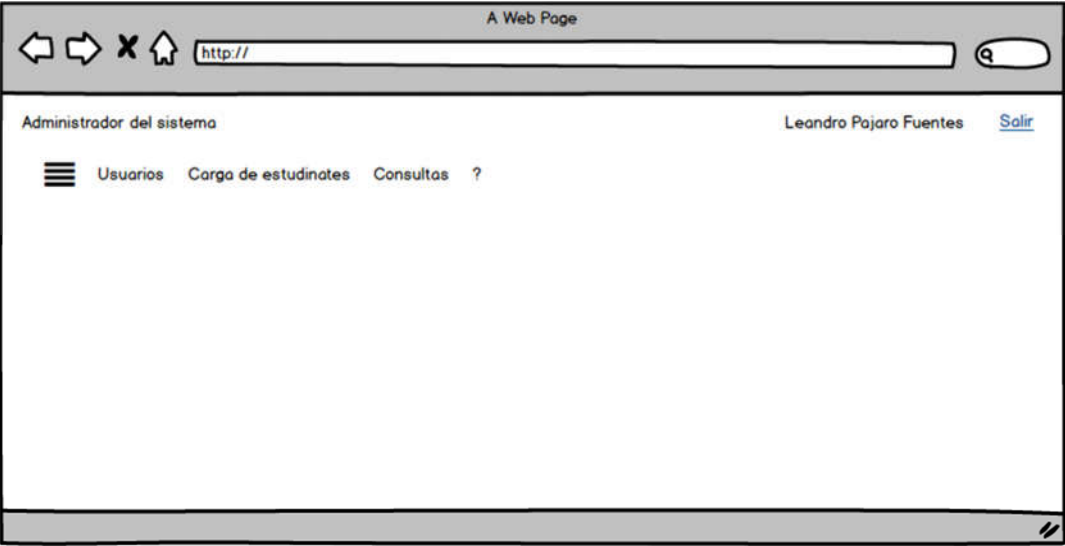
- Developer 2 pt
- Leandro Pájaro
- QA Engineer
- Total 2 pt**

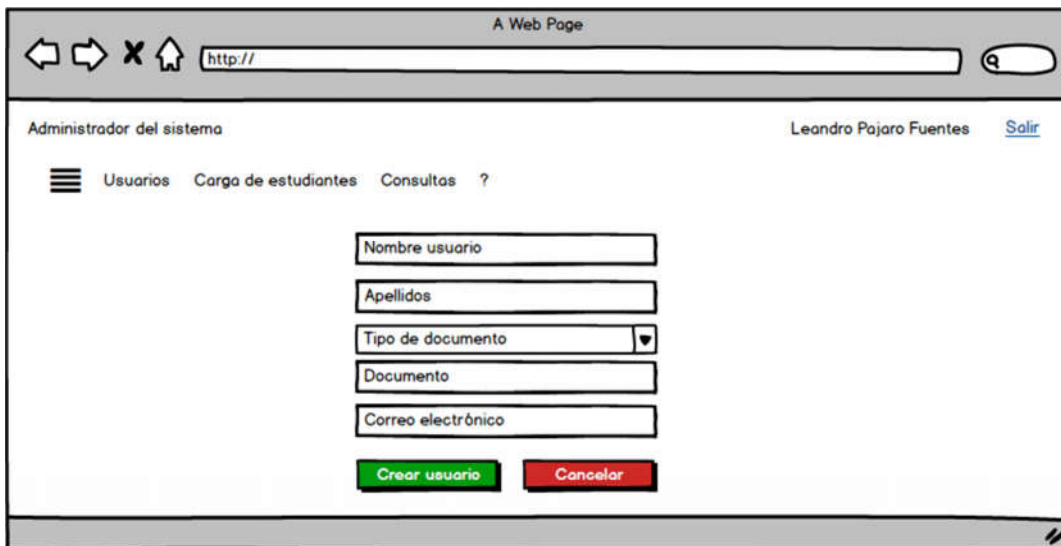
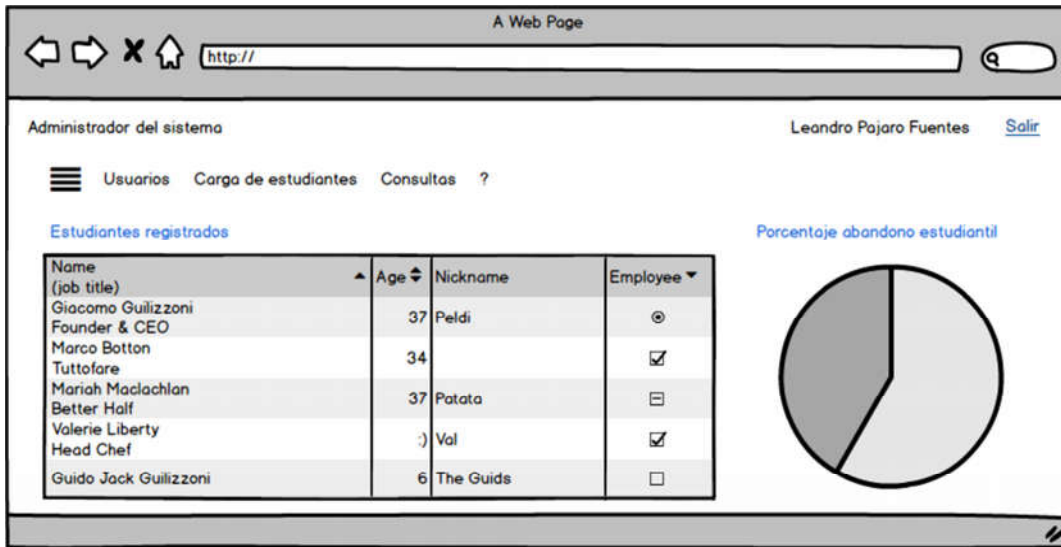
Info Sistema Predictivo Basado En Aprendizaje

14 Interfaces de usuario

14.1 Prototipos

A continuación se muestran algunos diseños de interfaces que tendría la aplicación:





14.2 Interfaces de usuario

Una vez realizado un diseño preliminar de algunas de las interfaces que tendrá la aplicación se ha utilizado una plantilla predefinida llamada Gentelella¹² basada en la librería Bootstrap, la cual facilita el desarrollo del entorno visual que tendrá la aplicación. A continuación se detallan algunas de las interfaces que tiene la aplicación

¹² <https://colorlib.com/polygon/gentelella/index.html>

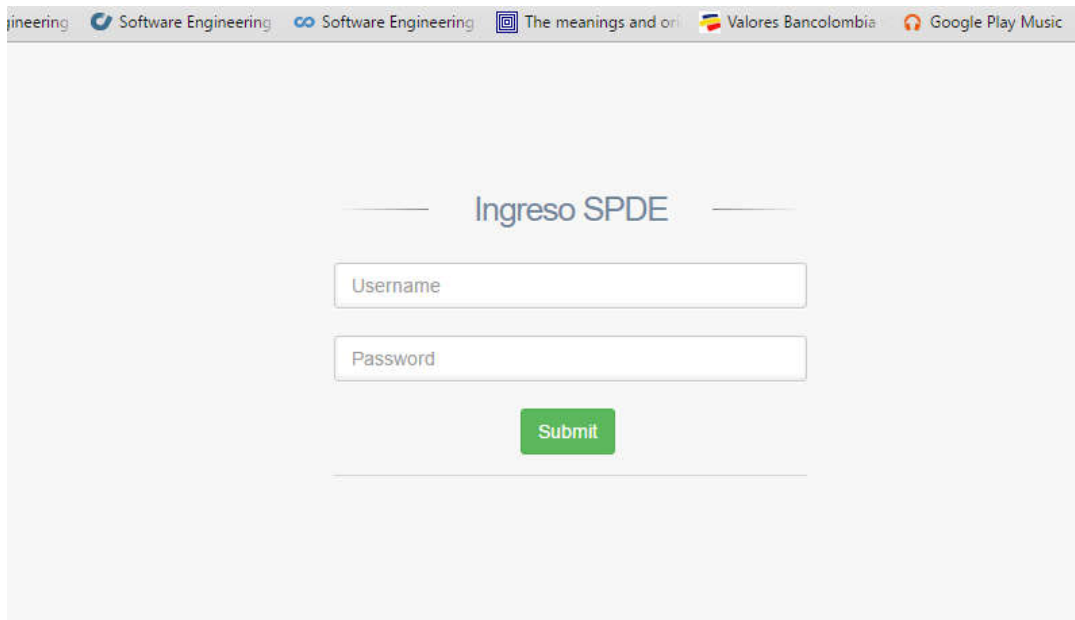


Figura 4 Página de ingreso a la aplicación

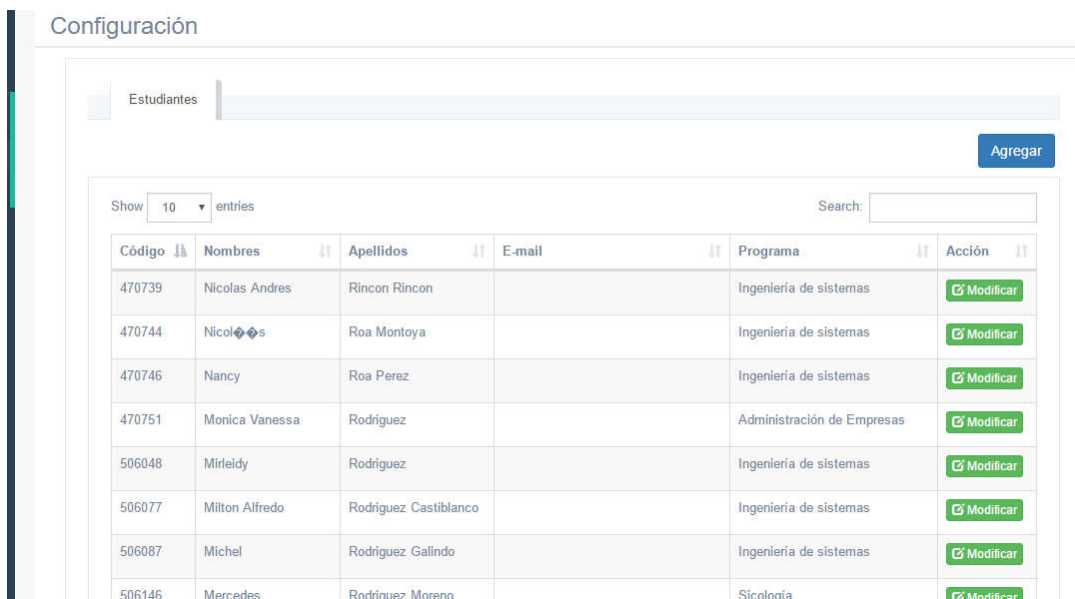


Figura 5 Listado de estudiantes

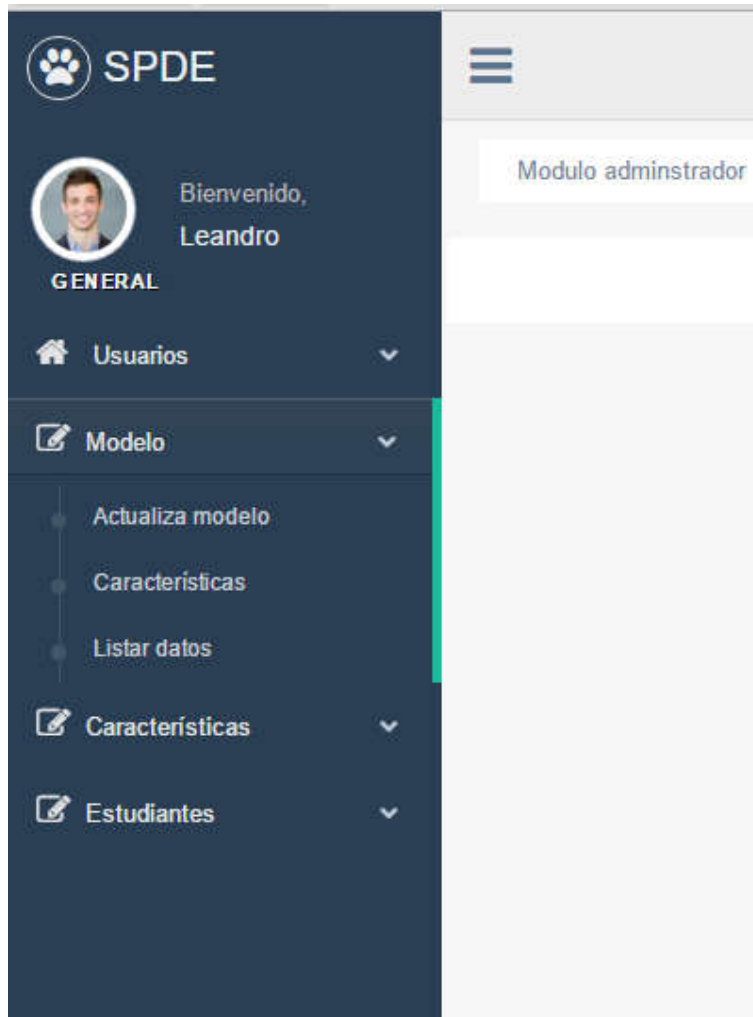


Figura 6 Menú aplicación

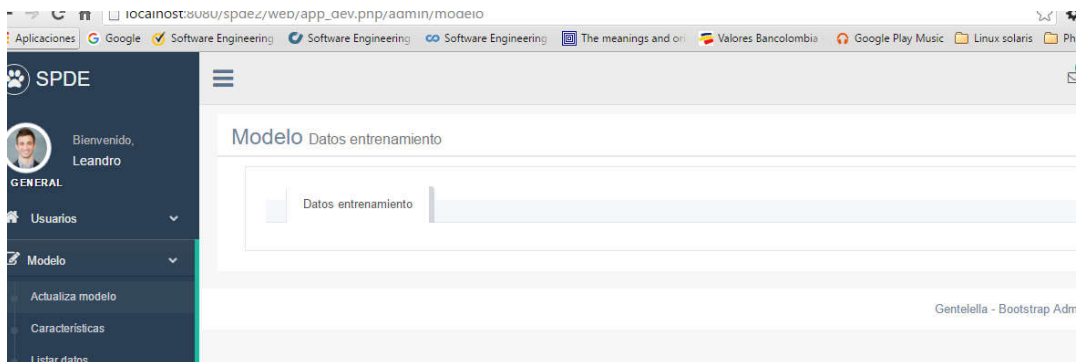
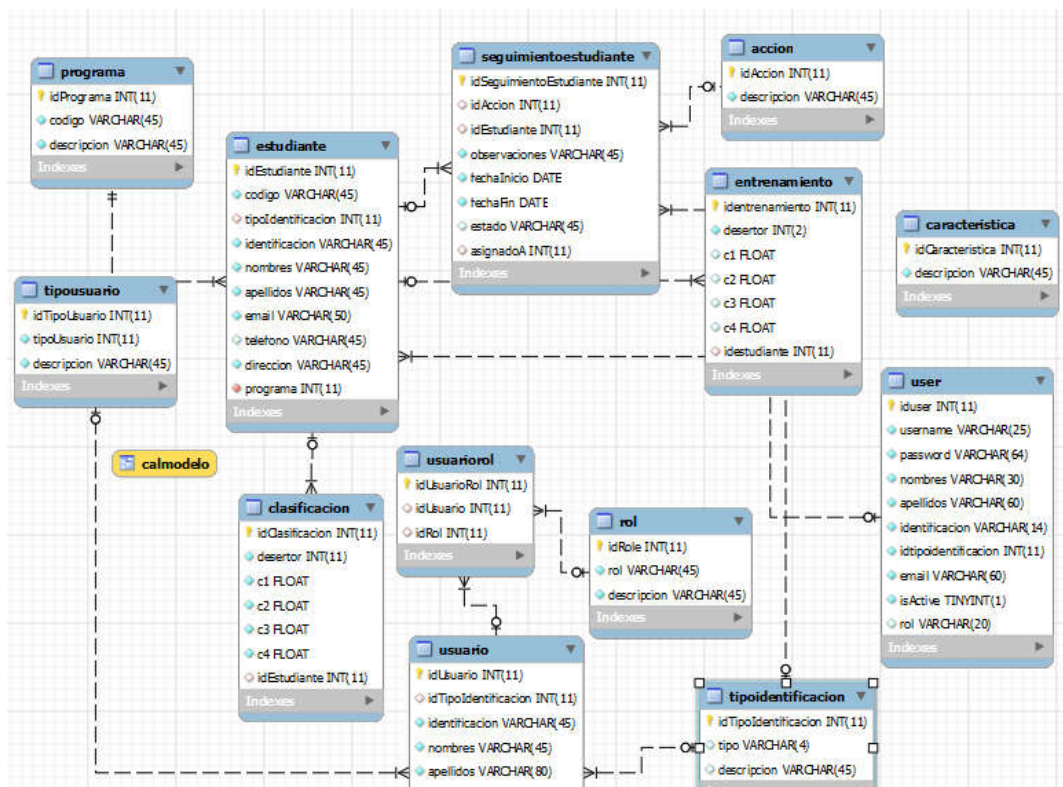


Figura 7 Plantilla de contenido

15 Diseño de base de datos

El gestor de base de datos utilizado es Mysql 5.6.16 y sobre el cual se realiza la persistencia de los datos. Como herramienta para realizar la modelación y diseño preliminar de la base de datos se ha usado Mysql WorkBench.

El sistema está compuesto por 10 tablas las cuales se describen a continuación y se anexa el diagrama entidad relación.



Descripción de tablas

Nombre tabla	Descripción
tiposUsuarios	Contiene información de los distintos tipos de usuarios que harán parte del sistema, dentro de los cuales están, estudiantes, docentes, empleados, entre otros
usuarios	Almacena los datos de los usuarios que interactuarán con la aplicación, comprende información básica, como nombres, direcciones teléfonos
roles	Define los diferentes roles que tendrá el sistema
usuarioRol	Mantiene una relación entre los usuarios y los diferentes roles que puede tener asignado
tipoidentificación	Define los tipos de documentos de identificación que puede tener un estudiante y/o usuario

estudiantes	Información básica de los estudiantes de la institución
entrenamiento	Tabla con la información que el sistema utiliza para realizar el aprendizaje y posterior clasificación de estudiantes. Contiene los valores de 5 características que pueden categorizar a un estudiante en desertor o no.
caracteristicas	Describe cada una de las características que definen un estudiante como desertor o no.
seguimientoEstudiante	Almacena información de las acciones tomadas para que un estudiante no deserte de la institución.
acciones	Describe cada una de las acciones que se pueden llevar a cabo sobre un estudiante para asegurar su permanencia

15.1 Script base de datos

A continuación se detallan los scripts de creación de tablas de la base de datos

```
-- phpMyAdmin SQL Dump
-- version 4.1.12
-- http://www.phpmyadmin.net
--
-- Servidor: 127.0.0.1
-- Tiempo de generación: 10-06-2016 a las 14:33:02
-- Versión del servidor: 5.6.16
-- Versión de PHP: 5.5.11

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
SET time_zone = "+00:00";

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8 */;

--
-- Base de datos: `spde`
--
-----

--
-- Estructura de tabla para la tabla `accion`
--

CREATE TABLE IF NOT EXISTS `accion` (
  `idAccion` int(11) NOT NULL AUTO_INCREMENT,
  `descripcion` varchar(45) COLLATE latin1_spanish_ci NOT NULL,
  PRIMARY KEY (`idAccion`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci
AUTO_INCREMENT=4 ;

--
```

```

-- Volcado de datos para la tabla `accion`
--

INSERT INTO `accion` (`idAccion`, `descripcion`) VALUES
(2, 'Tutorías'),
(3, 'Opciones de financiamiento');

-----

--
-- Estructura Stand-in para la vista `calmodelo`
--
CREATE TABLE IF NOT EXISTS `calmodelo` (
`id` bigint(20)
,`desertor` varchar(11)
,`probabilidad` decimal(24,4)
,`mediaC1` double
,`varianzac1` double
,`mediac2` double
,`varianzac2` double
,`mediac3` double
,`varianzac3` double
,`mediac4` double
,`varianzac4` double
);

-----

--
-- Estructura de tabla para la tabla `caracteristica`
--

CREATE TABLE IF NOT EXISTS `caracteristica` (
`idCaracteristica` int(11) NOT NULL AUTO_INCREMENT,
`descripcion` varchar(45) COLLATE latin1_spanish_ci NOT NULL,
PRIMARY KEY (`idCaracteristica`),
UNIQUE KEY `descripcion` (`descripcion`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci
AUTO_INCREMENT=9 ;

--
-- Volcado de datos para la tabla `caracteristica`
--

INSERT INTO `caracteristica` (`idCaracteristica`, `descripcion`)
VALUES
(3, 'Número de asignaturas'),
(1, 'Promedio'),
(2, 'Semestre'),
(4, 'Sexo');

-----

--
-- Estructura de tabla para la tabla `clasificacion`
--

CREATE TABLE IF NOT EXISTS `clasificacion` (
`idClasificacion` int(11) NOT NULL AUTO_INCREMENT,
`desertor` int(11) NOT NULL,
`c1` float NOT NULL,
`c2` float NOT NULL,

```

```

`c3` float NOT NULL,
`c4` float NOT NULL,
`idEstudiante` int(11) DEFAULT NULL,
PRIMARY KEY (`idClasificacion`),
KEY `fk_idEstudiante` (`idEstudiante`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci
AUTO_INCREMENT=149 ;

```

```

--
-- Volcado de datos para la tabla `clasificacion`
--

```

```

INSERT INTO `clasificacion` (`idClasificacion`, `desertor`, `c1`,
`c2`, `c3`, `c4`, `idEstudiante`) VALUES
(146, 0, 6, 130, 8, 1, 365),
(147, 1, 6, 180, 11, 1, 366),
(148, 0, 5, 160, 10, 1, 370);

```

```

-----
--
-- Estructura de tabla para la tabla `entrenamiento`
--

```

```

CREATE TABLE IF NOT EXISTS `entrenamiento` (
  `identrenamiento` int(11) NOT NULL AUTO_INCREMENT,
  `desertor` int(2) NOT NULL,
  `c1` float DEFAULT NULL,
  `c2` float DEFAULT NULL,
  `c3` float DEFAULT NULL,
  `c4` float DEFAULT NULL,
  `idestudiante` int(11) DEFAULT NULL,
  PRIMARY KEY (`identrenamiento`),
  KEY `fk_idestudiante2` (`idestudiante`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=47 ;

```

```

--
-- Volcado de datos para la tabla `entrenamiento`
--

```

```

INSERT INTO `entrenamiento` (`identrenamiento`, `desertor`, `c1`,
`c2`, `c3`, `c4`, `idestudiante`) VALUES
(1, 1, 6, 180, 12, 1, 202),
(2, 1, 5.92, 190, 11, 1, 203),
(3, 1, 5.58, 170, 12, 1, 204),
(4, 1, 5.92, 165, 10, 1, 205),
(9, 0, 5, 100, 6, 1, 210),
(10, 0, 5.5, 150, 8, 1, 211),
(11, 0, 5.42, 130, 7, 1, 212),
(46, 0, 5.75, 150, 9, 1, 366);

```

```

-----
--
-- Estructura de tabla para la tabla `estudiante`
--

```

```

CREATE TABLE IF NOT EXISTS `estudiante` (
  `idEstudiante` int(11) NOT NULL AUTO_INCREMENT,
  `codigo` varchar(45) COLLATE latin1_spanish_ci NOT NULL,
  `tipoIdentificacion` int(11) DEFAULT NULL,

```

```

`identificacion` varchar(45) COLLATE latin1_spanish_ci NOT NULL,
`nombres` varchar(45) COLLATE latin1_spanish_ci NOT NULL,
`apellidos` varchar(45) COLLATE latin1_spanish_ci NOT NULL,
`email` varchar(50) COLLATE latin1_spanish_ci NOT NULL,
`telefono` varchar(45) COLLATE latin1_spanish_ci DEFAULT NULL,
`direccion` varchar(45) COLLATE latin1_spanish_ci NOT NULL,
`programa` int(11) NOT NULL,
PRIMARY KEY (`idEstudiante`),
UNIQUE KEY `codigo` (`codigo`),
UNIQUE KEY `identificacion` (`identificacion`),
KEY `fk_tipoIdentificacionEst` (`tipoIdentificacion`),
KEY `fk_programa` (`programa`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci
AUTO_INCREMENT=401 ;

--
-- Volcado de datos para la tabla `estudiante`
--

INSERT INTO `estudiante` (`idEstudiante`, `codigo`,
`tipoIdentificacion`, `identificacion`, `nombres`, `apellidos`,
`email`, `telefono`, `direccion`, `programa`) VALUES
(202, '124488', 1, '5825971', 'LUIS FELIPE ANDRES', 'ZAMBRANO NOVOA',
'', '7887100', 'CR 45', 3),
(203, '124491', 1, '7546740', 'LUIS EDUARDO', 'ZAMBRANO CALVACHE', '',
'5438200', 'CL 70 BIS 119 24', 4),
(204, '124506', 1, '7547556', 'LUIS CARLOS', 'ZABALA GUZMAN', '',
'3656300', 'CR 11 C 65 59 SUR', 5),
(205, '124511', 1, '7706521', 'LIZETH LORENA', 'WALTEROS PACAVITA',
'', '8293400', 'CR 54 2 A 54', 5),
(206, '124520', 1, '13702156', 'LIRIS YANETH', 'VILLAR COHECHA', '',
'3723500', 'TV 5 F BIS 44 38', 5),
(207, '125969', 1, '14396683', 'LINA MARIA', 'VILLAMIL CASALLAS', '',
'5678600', 'CR 6 D ES 113 55 SUR', 5),
(208, '125975', 1, '16188763', 'LILIANA FABIOLA', 'VILLALBA AMAYA',
'', '7161700', 'CR 42 82 SUR 83', 5),

--
-- Estructura de tabla para la tabla `programa`
--

CREATE TABLE IF NOT EXISTS `programa` (
`idPrograma` int(11) NOT NULL AUTO_INCREMENT,
`codigo` varchar(45) COLLATE latin1_spanish_ci NOT NULL,
`descripcion` varchar(45) COLLATE latin1_spanish_ci NOT NULL,
PRIMARY KEY (`idPrograma`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci
AUTO_INCREMENT=6 ;

--
-- Volcado de datos para la tabla `programa`
--

INSERT INTO `programa` (`idPrograma`, `codigo`, `descripcion`) VALUES
(3, '1020', 'Administración de Empresas'),
(4, '1022', 'Sicología'),
(5, '1023', 'Ingeniería de sistemas');
-----

```

```

--
-- Estructura de tabla para la tabla `rol`
--

CREATE TABLE IF NOT EXISTS `rol` (
  `idRole` int(11) NOT NULL AUTO_INCREMENT,
  `rol` varchar(45) COLLATE latin1_spanish_ci NOT NULL,
  `descripcion` varchar(45) COLLATE latin1_spanish_ci NOT NULL,
  PRIMARY KEY (`idRole`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci
AUTO_INCREMENT=1 ;

-----

--
-- Estructura de tabla para la tabla `seguimientoestudiante`
--

CREATE TABLE IF NOT EXISTS `seguimientoestudiante` (
  `idSeguimientoEstudiante` int(11) NOT NULL AUTO_INCREMENT,
  `idAccion` int(11) DEFAULT NULL,
  `idEstudiante` int(11) DEFAULT NULL,
  `observaciones` varchar(45) COLLATE latin1_spanish_ci NOT NULL,
  `fechaInicio` date NOT NULL,
  `fechaFin` date NOT NULL,
  `estado` varchar(45) COLLATE latin1_spanish_ci DEFAULT NULL,
  `asignadoA` int(11) DEFAULT NULL,
  PRIMARY KEY (`idSeguimientoEstudiante`),
  KEY `fk_idEstudianteSeg` (`idEstudiante`),
  KEY `fk_idAccion` (`idAccion`),
  KEY `fk_asignadoA` (`asignadoA`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci
AUTO_INCREMENT=7 ;

--
-- Volcado de datos para la tabla `seguimientoestudiante`
--

INSERT INTO `seguimientoestudiante` (`idSeguimientoEstudiante`,
`idAccion`, `idEstudiante`, `observaciones`, `fechaInicio`,
`fechaFin`, `estado`, `asignadoA`) VALUES
(5, 2, 365, 'Fortalecer matemáticas', '2016-06-08', '2016-06-15', '1',
5),
(6, 2, 366, 'Créditos', '2016-06-09', '2016-06-16', '1', 4);

-----

--
-- Estructura de tabla para la tabla `tipoidentificacion`
--

CREATE TABLE IF NOT EXISTS `tipoidentificacion` (
  `idTipoIdentificacion` int(11) NOT NULL AUTO_INCREMENT,
  `tipo` varchar(4) COLLATE latin1_spanish_ci DEFAULT NULL,
  `descripcion` varchar(45) COLLATE latin1_spanish_ci DEFAULT NULL,
  PRIMARY KEY (`idTipoIdentificacion`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci
AUTO_INCREMENT=3 ;

--
-- Volcado de datos para la tabla `tipoidentificacion`

```

```

--

INSERT INTO `tipoidentificacion` (`idTipoIdentificacion`, `tipo`,
`descripcion`) VALUES
(1, '1', 'Cédula de ciudadanía'),
(2, '2', 'Tarjeta de identidad');

-----

--
-- Estructura de tabla para la tabla `tipousuario`
--

CREATE TABLE IF NOT EXISTS `tipousuario` (
  `idTipoUsuario` int(11) NOT NULL AUTO_INCREMENT,
  `tipoUsuario` int(11) NOT NULL,
  `descripcion` varchar(45) COLLATE latin1_spanish_ci NOT NULL,
  PRIMARY KEY (`idTipoUsuario`),
  UNIQUE KEY `tipoUsuario` (`tipoUsuario`),
  UNIQUE KEY `descripcion` (`descripcion`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci
AUTO_INCREMENT=18 ;

--
-- Volcado de datos para la tabla `tipousuario`
--

INSERT INTO `tipousuario` (`idTipoUsuario`, `tipoUsuario`,
`descripcion`) VALUES
(1, 1, 'Estudiante'),
(2, 2, 'Docente'),
(3, 4, 'Administrativo');

-----

--
-- Estructura de tabla para la tabla `user`
--

CREATE TABLE IF NOT EXISTS `user` (
  `iduser` int(11) NOT NULL AUTO_INCREMENT,
  `username` varchar(25) NOT NULL,
  `password` varchar(64) NOT NULL,
  `nombres` varchar(30) NOT NULL,
  `apellidos` varchar(60) NOT NULL,
  `identificacion` varchar(14) NOT NULL,
  `idtipoidentificacion` int(11) NOT NULL,
  `email` varchar(60) NOT NULL,
  `isActive` tinyint(1) NOT NULL,
  `rol` varchar(20) DEFAULT NULL,
  PRIMARY KEY (`iduser`),
  UNIQUE KEY `username` (`username`,`email`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=7 ;

--
-- Volcado de datos para la tabla `user`
--

INSERT INTO `user` (`iduser`, `username`, `password`, `nombres`,
`apellidos`, `identificacion`, `idtipoidentificacion`, `email`,
`isActive`, `rol`) VALUES

```

```

(4, 'admin',
'$2y$13$D4BpJBkOVjrOd8gyjOmXq.lvtOD4jbPVwhv9olbeD8quUyirgmo4G',
'Leandro', 'Pájaro Fuentes', '73574448', 1, 'lpajarof@gmail.com', 1,
'ROLE_ADMIN'),
(5, 'vmartinez',
'$2y$13$hsm7W0o2IqQCD.IB6pb0tumsFikB6RtenjDlM5PzenuyrHNWW8Npa',
'Valentina', 'Martinez', '454534', 1, 'valentina@gmail.com', 1,
'ROLE_USER'),
(6, 'llima',
'$2y$13$cctmtOVh3VeafiHudHdAquBRz/yKVvXMryH4k7hNXFYj.e3PrsZdC',
'Lucas', 'Lima', '6789010', 1, 'lucasilima@gmail.com', 1, 'ROLE_USER');

```

```

--
-- Estructura de tabla para la tabla `usuario`
--

```

```

CREATE TABLE IF NOT EXISTS `usuario` (
  `idUsuario` int(11) NOT NULL AUTO_INCREMENT,
  `idTipoIdentificacion` int(11) DEFAULT NULL,
  `identificacion` varchar(45) COLLATE latin1_spanish_ci NOT NULL,
  `nombres` varchar(45) COLLATE latin1_spanish_ci NOT NULL,
  `apellidos` varchar(80) COLLATE latin1_spanish_ci NOT NULL,
  `email` varchar(80) COLLATE latin1_spanish_ci NOT NULL,
  `contrasena` varchar(45) COLLATE latin1_spanish_ci DEFAULT NULL,
  `telefono` varchar(45) COLLATE latin1_spanish_ci DEFAULT NULL,
  `idTipoUsuario` int(11) DEFAULT NULL,
  PRIMARY KEY (`idUsuario`),
  KEY `fk_tipoIdentificacion` (`idTipoIdentificacion`),
  KEY `fk_tipousuario` (`idTipoUsuario`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci
AUTO_INCREMENT=2 ;

```

```

--
-- Volcado de datos para la tabla `usuario`
--

```

```

INSERT INTO `usuario` (`idUsuario`, `idTipoIdentificacion`,
`identificacion`, `nombres`, `apellidos`, `email`, `contrasena`,
`telefono`, `idTipoUsuario`) VALUES
(1, 1, '73574448', 'Leandro', 'Pájaro', 'lpajarof@gmail.com', NULL,
'3004041950', 3);

```

```

--
-- Estructura de tabla para la tabla `usuariorol`
--

```

```

CREATE TABLE IF NOT EXISTS `usuariorol` (
  `idUsuarioRol` int(11) NOT NULL AUTO_INCREMENT,
  `idUsuario` int(11) DEFAULT NULL,
  `idRol` int(11) DEFAULT NULL,
  PRIMARY KEY (`idUsuarioRol`),
  KEY `fk_idUsuario` (`idUsuario`),
  KEY `fk_idRol` (`idRol`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci
AUTO_INCREMENT=1 ;

```

```

--
-- Estructura para la vista `calmodelo`
--
DROP TABLE IF EXISTS `calmodelo`;

CREATE ALGORITHM=UNDEFINED DEFINER=`root`@`localhost` SQL SECURITY
DEFINER VIEW `calmodelo` AS select 1 AS
`id`,if((`entrenamiento`.`desertor` =
'1'),'Desertor',`entrenamiento`.`desertor`) AS
`desertor`,(count(`entrenamiento`.`identrenamiento`) / (select
count(`entrenamiento`.`identrenamiento`) from `entrenamiento`)) AS
`probabilidad`,`avg`(`entrenamiento`.`c1`) AS
`mediaC1`,`var_samp`(`entrenamiento`.`c1`) AS
`varianzac1`,`avg`(`entrenamiento`.`c2`) AS
`mediac2`,`var_samp`(`entrenamiento`.`c2`) AS
`varianzac2`,`avg`(`entrenamiento`.`c3`) AS
`mediac3`,`var_samp`(`entrenamiento`.`c3`) AS
`varianzac3`,`avg`(`entrenamiento`.`c4`) AS
`mediac4`,`var_samp`(`entrenamiento`.`c4`) AS `varianzac4` from
`entrenamiento` where (`entrenamiento`.`desertor` = 1) union select 0
AS `id`,if((`entrenamiento`.`desertor` = '0'),'No
Desertor',`entrenamiento`.`desertor`) AS
`desertor`,(count(`entrenamiento`.`identrenamiento`) / (select
count(`entrenamiento`.`identrenamiento`) from `entrenamiento`)) AS
`probabilidad`,`avg`(`entrenamiento`.`c1`) AS
`mediaC1`,`var_samp`(`entrenamiento`.`c1`) AS
`varianzac1`,`avg`(`entrenamiento`.`c2`) AS
`mediac2`,`var_samp`(`entrenamiento`.`c2`) AS
`varianzac2`,`avg`(`entrenamiento`.`c3`) AS
`mediac3`,`var_samp`(`entrenamiento`.`c3`) AS
`varianzac3`,`avg`(`entrenamiento`.`c4`) AS
`mediac4`,`var_samp`(`entrenamiento`.`c4`) AS `varianzac4` from
`entrenamiento` where (`entrenamiento`.`desertor` = 0);

--
-- Restricciones para tablas volcadas
--

--
-- Filtros para la tabla `clasificacion`
--
ALTER TABLE `clasificacion`
  ADD CONSTRAINT `fk_idEstudiante` FOREIGN KEY (`idEstudiante`)
REFERENCES `estudiante` (`idEstudiante`);

--
-- Filtros para la tabla `entrenamiento`
--
ALTER TABLE `entrenamiento`
  ADD CONSTRAINT `fk_idestudiante2` FOREIGN KEY (`idestudiante`)
REFERENCES `estudiante` (`idEstudiante`);

--
-- Filtros para la tabla `estudiante`
--
ALTER TABLE `estudiante`
  ADD CONSTRAINT `fk_programa` FOREIGN KEY (`programa`) REFERENCES
`programa` (`idPrograma`),

```



```

    ADD CONSTRAINT `fk_tipoIdentificacionEst` FOREIGN KEY
(`tipoIdentificacion`) REFERENCES `tipoidentificacion`
(`idTipoIdentificacion`);

--
-- Filtros para la tabla `seguimientoestudiante`
--
ALTER TABLE `seguimientoestudiante`
    ADD CONSTRAINT `fk_asignadoA` FOREIGN KEY (`asignadoA`) REFERENCES
`user` (`idUser`),
    ADD CONSTRAINT `fk_idAccion` FOREIGN KEY (`idAccion`) REFERENCES
`accion` (`idAccion`),
    ADD CONSTRAINT `fk_idEstudianteSeg` FOREIGN KEY (`idEstudiante`)
REFERENCES `estudiante` (`idEstudiante`);

--
-- Filtros para la tabla `usuario`
--
ALTER TABLE `usuario`
    ADD CONSTRAINT `fk_tipoIdentificacion` FOREIGN KEY
(`idTipoIdentificacion`) REFERENCES `tipoidentificacion`
(`idTipoIdentificacion`),
    ADD CONSTRAINT `fk_tipousuario` FOREIGN KEY (`idTipoUsuario`)
REFERENCES `tipousuario` (`idTipoUsuario`);

--
-- Filtros para la tabla `usuariorol`
--
ALTER TABLE `usuariorol`
    ADD CONSTRAINT `fk_idRol` FOREIGN KEY (`idRol`) REFERENCES `rol`
(`idRole`),
    ADD CONSTRAINT `fk_idUsuario` FOREIGN KEY (`idUsuario`) REFERENCES
`usuario` (`idUsuario`);

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;

```

16 Pruebas

Para la validación de las funcionalidades de la aplicación y verificación de los objetivos planteados se realizaron una serie de casos de pruebas donde se valida el correcto funcionamiento de la solución.

Id caso	Funcionalidad a probar	Prerrequisitos	Resultado esperado	Resultado obtenido
CPM01	Creación de datos de entrenamiento del modelo	Datos de estudiantes previamente ingresados	Datos de entrenamiento registrados	Se puede realizar el registro de datos de entrenamiento de manera satisfactoria
CPM02	Modificación de datos de entrenamiento	Registro de datos de entrenamiento	Modificar cualquiera de las variables que constituyen el entrenamiento a modificar	El sistema permite la modificación de cualquiera de las siguientes variables: Estudiante, características e indicador de desertor
CPM03	Eliminación de datos de entrenamiento	Registro de datos de entrenamiento	Eliminación de registro de entrenamientos	Eliminación de registros de forma correcta
CC01	Clasificar a un estudiante en desertor o no desertor	Datos de entrenamiento previamente ingresados	Clasificación de un estudiante en desertor o no desertor	El sistema de acuerdo a los datos previamente ingresados de entrenamiento logra clasificar a un estudiante.
CC02	Modificación de datos de clasificación	Existencia de registros de estudiantes clasificados	Modificación de cualquier de las variables de clasificación	Se puede realizar la modificación de los datos de manera satisfactoria.
CC03	Eliminación de registros de clasificación	Existencia de registros de estudiantes clasificados	Eliminación de registro de clasificación	Se puede eliminar sin inconveniente registros de estudiantes clasificados
CL01	Listado de estudiantes	Datos de estudiantes cargados en el sistema	Listado de estudiantes	Al seleccionar del menú estudiante, la opción listar se visualizan todos los estudiantes cargados.
CL02	Modificar datos de estudiantes	Existencia de registros de estudiantes clasificados	Cambio de alguna de las siguientes variables código, nombres, apellidos, email, programa	La aplicación permite la modificación de manera satisfactoria de las variables de registro de los estudiantes.
CL03	Registro de estudiantes	Datos de programas previamente ingresados	Registro de datos de estudiante	Se pueden ingresar nuevos registros de estudiantes a la aplicación
CS01	Creación de seguimiento a un estudiante	Usuarios registrados. Acciones parametrizadas Estudiante clasificado como posible desertor	Creación de un seguimiento a un estudiante clasificado como desertor	Se logra crear seguimiento a estudiante clasificado como desertor, asignando una acción de seguimiento y un responsable
CS02	Modificación de datos de un seguimiento	Previo registro de un seguimiento	Poder modificar alguno de los datos del seguimiento	La modificación de datos se realiza sin inconvenientes
CS03	Eliminación de seguimiento	Previo registro de un seguimiento	Eliminación de seguimiento	Se logra eliminar seguimiento de manera satisfactoria.
CO01	Realizar operaciones de consulta, adición, modificación y/o eliminación de tipos de usuarios, programas, características y usuarios		Realizar operaciones de consulta, adición, modificación y/o eliminación de registros sobre las tablas de tipo de usuarios, usuarios, acciones y características	Las operaciones de consulta, adición, modificación y eliminación de logran hacer sin evidenciar ningún inconveniente

17 Conclusiones

El objetivo de diseñar y construir una aplicación web que permita la predicción con base en métodos probabilísticos de la deserción estudiantil se ha cumplido con lo cual se convierte en una herramienta útil para instituciones de educación superior que deseen establecer estrategias encaminadas a la disminución del porcentaje de deserción estudiantil.

A través del desarrollo del proyecto se identificaron algunos factores que pueden ser catalizadores de la deserción estudiantil, como lo son el estrato económico, situación laboral, métodos de financiación de los estudios, entre otros, los cuales fueron utilizados para simulaciones en la identificación de desertores dentro de una muestra estudiantil. Los resultados aunque muy acertados evidenciaron que cada institución puede tener variables muy específicas que puedan explicar el fenómeno de la deserción, razón por la cual en el desarrollo de la aplicación se decidió por no utilizar variables estáticas para la clasificación de la deserción en cambio se diseñó la aplicación para que estas variables fueran parametrizadas de acuerdo a las necesidades de cada institución.

Teniendo ya identificadas las variables propias de cada institución con las cuales se realizará la identificación de desertores, se eligió el clasificador probabilístico Naïve Bayes por ser uno de los más rápidos y simple de utilizar. Este clasificador requiere datos de entrenamiento previo para poder clasificar nuevos datos, para lo cual se realizó carga de datos en gestor de bases de datos y se realizó una vista que contiene los cálculos necesarios para la aplicación de la fórmula:

$$P(x = v|c) = \frac{1}{\sqrt{2\pi\sigma_c^2}} e^{-\frac{(v-\mu_c)^2}{2\sigma_c^2}}$$

De esta manera el sistema responderá en menor tiempo al momento de realizar la clasificación de individuos.

Con las variables identificadas, el modelo de datos creado, se realizó la codificación de la aplicación web. Una parte fundamental del desarrollo de la aplicación es la interfaz gráfica, la cual se realizó en su totalidad con una plantilla de bootstrap llamada Gentelella y es posible descargarla de la url <https://colorlib.com/polygon/gentelella/>. Esta plantilla permitió agilizar mucho la construcción de la interfaz de usuario al poseer hojas de estilos y funciones en Javascript para la interacción entre la aplicación y el usuario final.

Teniendo en conjunto las variables, modelo y aplicación web la construcción de una herramienta que con datos históricos puede predecir con una alta probabilidad si un estudiante puede ser o no un desertor y establecer seguimientos a través de estrategias que permitan eliminar o disminuir los factores que pueden incidir en el abandono de la institución.

18 Anexo 1 Entregables del proyecto

A continuación se listan los entregables del proyecto:

- Documento del proyecto
- Código fuente de la aplicación
- Manual de usuario
- Manual de instalación
- Presentación
- Video

19 Anexo 2 Código fuente del proyecto

Todo el código fuente del proyecto se ha publicado en la herramienta github y es de acceso libre a través de la url <https://github.com/lpajarof/spde2>


20 Anexo 3 Manual de usuario

20.1 Acceso al sistema

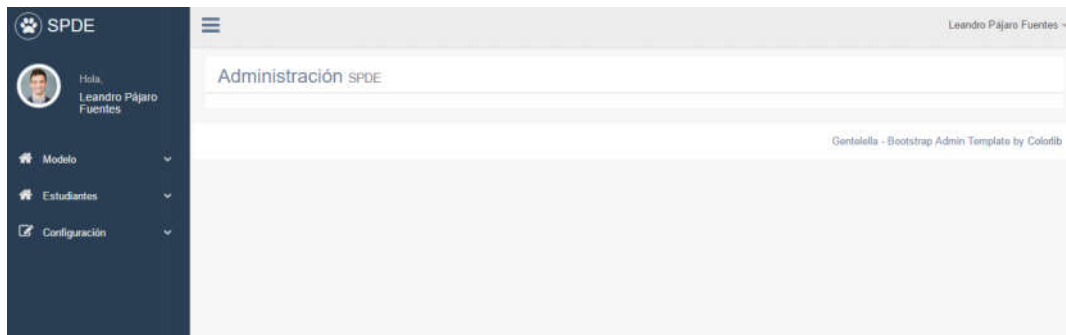
El sistema cuenta con mecanismo de autenticación para lo cual se requiere tener un usuario y contraseña. A continuación se presenta pantalla principal de acceso al sistema a través de la url:

http://localhost:8080/spde2/web/app_dev.php/login

:

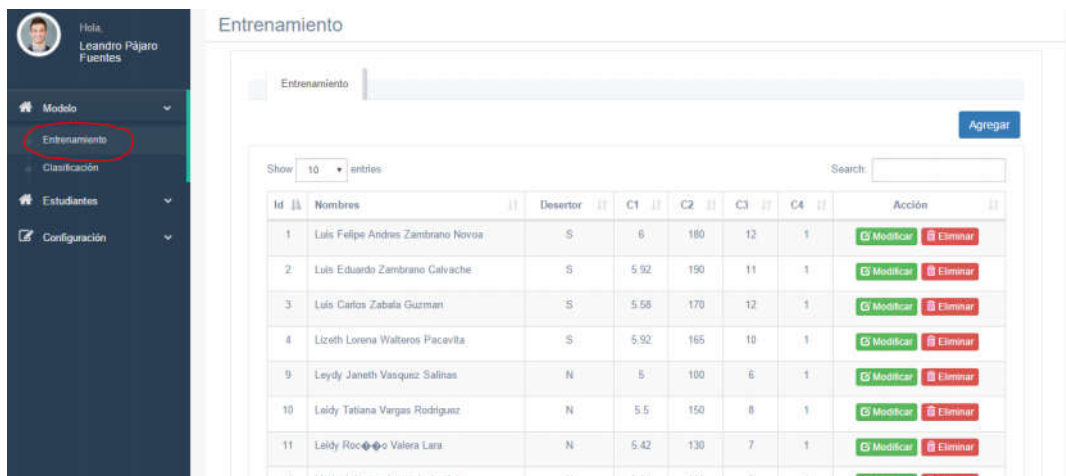


Al ingresar con las credenciales usuario *admin* y contraseña *admin* el sistema permitirá el acceso con el rol de administrador y mostrará las siguientes opciones:




20.2 Entrenamiento

Una vez se ha ingresado a la aplicación con el rol de administrador, lo primero que se debe realizar es el entrenamiento del modelo, para lo cual se debe ingresar a la opción entrenamiento del menú lateral. En esta opción se podrá realizar el registro de los datos con los cuales se identifica un desertor y lo que no es un desertor.



En esta opción se muestran los datos que son utilizados para el entrenamiento del modelo. También se podrán agregar datos, modificar los existentes o eliminarlos.

El botón agregar  permite abrir formulario para agregar nuevos datos.

Cuando se hace clic sobre el botón agregar se visualiza el formulario para el registro de nuevos datos

En este formulario se ingresan los valores correspondientes a las características definidas con las cuales se realizará la clasificación de individuos.

En este mismo módulo se pueden eliminar registros y/o modificarlos.

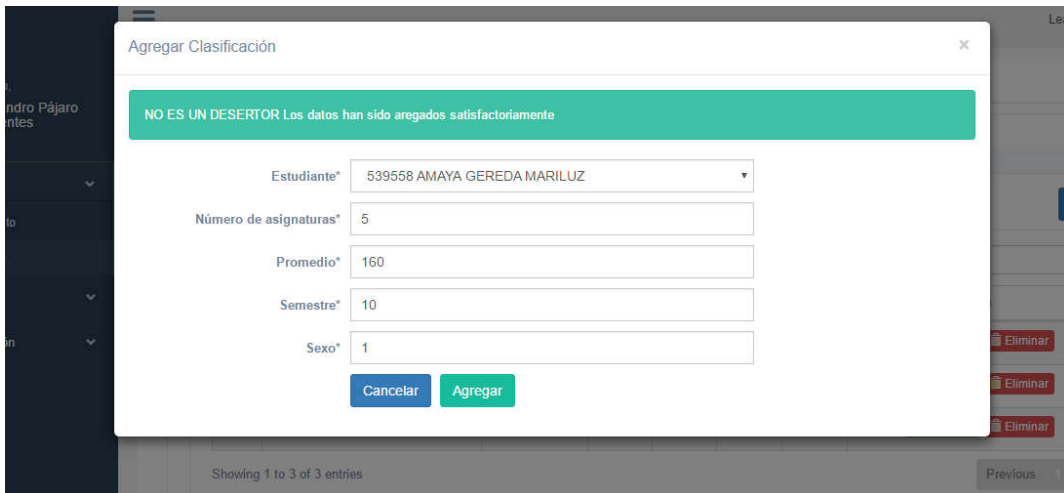
20.3 Clasificación

La clasificación es el proceso mediante el cual un nuevo individuo es clasificado como desertor o no desertor.

Id	Nombres	Desertor	C1	C2	C3	C4	Acción
146	Martha Alejandra Acero Rivera	N	6	130	8	1	Modificar Eliminar
147	Marly Johanna Acevedo Aguirre	S	6	180	11	1	Modificar Eliminar

Al ingresar a la opción de clasificación se visualizarán los registros de los estudiantes que están clasificados.

Para clasificar un nuevo estudiante se procede a ingresar los valores de las características.

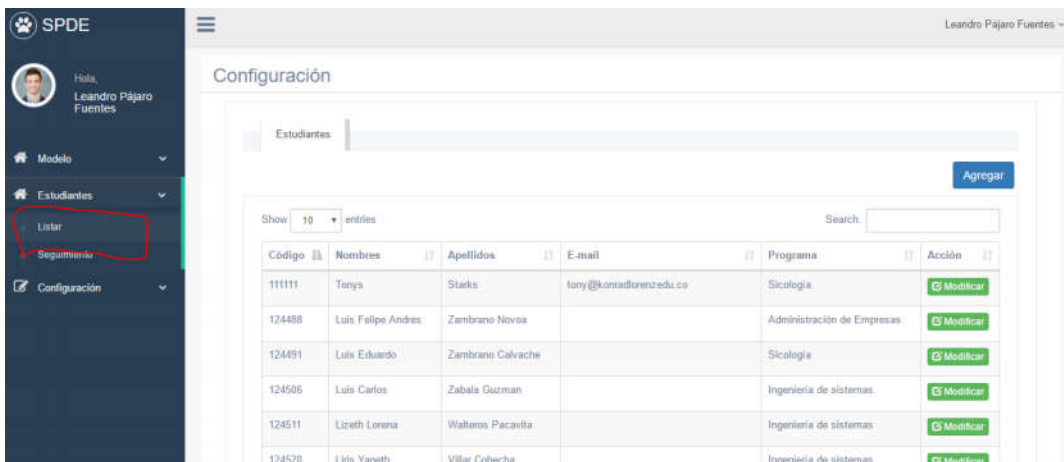


Posteriormente al hacer clic en el botón agregar el sistema informará si este nuevo individuo es un posible desertor o no.

Esta clasificación puede modificarse o eliminarse. En caso de que el estudiante pueda ser un desertor se podrá asignar un seguimiento al estudiante.

20.4 Estudiantes

En esta opción se ingresan todos los estudiantes a los cuales se les podrá clasificar y/o se puedan registrar como parte del entrenamiento.



La forma de creación de nuevo estudiante permite ingresar la siguiente información:

- Código del estudiante
- Programa académico al que pertenece
- Tipo de documento de identificación

- Nombres
- Apellidos
- Email
- Teléfono
- Dirección

The image shows a screenshot of a web application interface. A modal window titled "Agregar Estudiante" is open, displaying a form for adding a student. The form contains the following fields and controls:

- Código ***: A text input field.
- Programa***: A dropdown menu with "Choose.." selected.
- Tipo de Identificación ***: A dropdown menu with "Choose.." selected.
- Identificación ***: A text input field.
- Nombres ***: A text input field.
- Apellidos***: A text input field.
- email ***: A text input field.
- Teléfono***: A text input field.
- Dirección ***: A text input field.

At the bottom of the form, there are two buttons: a blue "Close" button and a green "Agregar" button. The background shows a dark sidebar with the word "ajaro" and several dropdown menus, and a table with columns for "124520", "Luis Venoth", "Villar Cabanba", and "Ingeniería de sistemas".

20.5 Seguimiento

El seguimiento hace referencia a las acciones que se le asignan a un estudiante para que disminuya el riesgo de desertar.

Todo estudiante que la aplicación haya clasificado como posible desertor será candidato para realizar un seguimiento.

Al ingresar a la opción de seguimiento la aplicación mostrará el listado de estudiantes a los cuales se les está o ha hecho algún tipo de acompañamiento.

Estudiante	Seguimiento	Observaciones	Fecha Inicio	Fecha Fin	Estado	Asignado a	Acción
Marly Johanna Acevedo Aguirre	Tutorías	Créditos	05/09/2016	06/16/2016	Abierta	Leandro Pájaro Fuentes	Modificar Eliminar
Martha Alejandra Acero Rivera	Tutorías	Fortalecer matemáticas	05/08/2016	06/15/2016	Abierta	Valentina Martínez	Modificar Eliminar

Dentro de la información de seguimiento está:

- Nombre del estudiante
- Tipo de seguimiento
- Observaciones
- Fecha de inicio del seguimiento
- Fecha final del seguimiento
- Estado del seguimiento, si se encuentra en proceso o está finalizada
- Personal al cual fue asignado el seguimiento.

A continuación se muestra forma para el ingreso de datos de seguimiento

Agregar Seguimiento

Estudiante* 539558 MARILUZ AMAYA GEREDA

Acción* Opciones de financiamiento

Observaciones Estudiante presenta problemas para el pago de su matrícula

Fecha inicio 06/10/2016

Fecha fin 06/15/2016

Estado* Abierta

Asignado a* Leandro Pájaro Fuentes

Cancelar Agregar

Showing 1 to 2 of 2 entries

20.6 Configuración

En este módulo se podrán parametrizar las diferentes variables del sistema, dentro de las cuales se encuentra:

- Tipos de usuario: Permite definir tipos de usuario que accederán al aplicativo (Estudiantes, docentes, administrativos)
- Usuarios: Gestiona la administración de los usuarios que ingresan al sistema, permite la creación, modificación o eliminación de usuarios.
- Programas: Se registran los diferentes programas académicos con los que cuenta la institución
- Características: Gestiona las diferentes características que tendrá el modelo para la clasificación de los estudiantes
- Acciones: Define las distintas acciones que se podrán asignar a un estudiante para minimizar el riesgo de deserción.

A continuación se muestran cada una de las opciones anteriormente descritas

- Tipos de usuarios

Configuración

Tipo de Usuarios

Agregar

Show 10 entries

Tipo de usuario	Descripción	Acción
1	Estudiante	Modificar
2	Docente	Modificar
4	Administrativo	Modificar

Showing 1 to 3 of 3 entries

Previous Next

- Usuarios

Configuración

Usuarios

Agregar

Show 10 entries Search:

Nombres	Apellidos	Identificación	E-mail	Usuario	Rol	Acción
Leandro	Pájaro Fuentes	73574448	lpajarof@gmail.com	admin	ROLE_ADMIN	Modificar Eliminar
Lucas	Lima	6789010	lucaslima@gmail.com	lima	ROLE_USER	Modificar Eliminar
Valentina	Martínez	454534	valentina@gmail.com	vmartinez	ROLE_USER	Modificar Eliminar

Showing 1 to 3 of 3 entries Previous Next

- Programas

Configuración

Programas

Agregar

Show 10 entries Search:

Código	Descripción	Acción
1020	Administración de Empresas	Modificar
1022	Sicología	Modificar
1023	Ingeniería de sistemas	Modificar

Showing 1 to 3 of 3 entries Previous Next

- Características

Configuración

Características

Agregar

Show 10 entries Search:

Id	Característica	Acción
1	Promedio	Modificar
2	Semestre	Modificar
3	Número de asignaturas	Modificar
4	Sexo	Modificar

Showing 1 to 4 of 4 entries Previous Next

- Acciones

The screenshot shows a web application interface with a dark sidebar on the left and a main content area on the right. The sidebar contains a user profile at the top with the name 'Leandro Pájaro Fuentes' and a list of menu items: 'Modelo', 'Estudiantes', 'Configuración', 'Tipos de usuarios', 'Programas', 'Características', 'Acciones', and 'Usuarios'. The 'Configuración' menu item is highlighted. The main content area is titled 'Configuración' and features a sub-header 'Acciones' with an 'Agregar' button. Below this is a table with columns 'Id', 'Acción', and 'Acción'. The table contains two rows: one for 'Tutorías' (Id 2) and one for 'Opciones de financiamiento' (Id 3). Each row has 'Modificar' and 'Eliminar' buttons. At the bottom of the table, it says 'Showing 1 to 2 of 2 entries' and has 'Previous' and 'Next' navigation buttons.

Hola,
Leandro Pájaro
Fuentes

Modelo
Estudiantes
Configuración
Tipos de usuarios
Programas
Características
Acciones
Usuarios

Configuración

Acciones Agregar

Show 10 entries Search:

Id	Acción	Acción
2	Tutorías	Modificar Eliminar
3	Opciones de financiamiento	Modificar Eliminar

Showing 1 to 2 of 2 entries Previous Next

21 Glosario

Bootstrap: Librería originalmente creada por Twitter, para la creación de interfaces de usuario mediante hojas de estilo y Javascript.

Casos de pruebas: Conjunto de condiciones que se validan para determinar si una aplicación cumple con las funcionalidades requeridas.

MVC Modelo vista controlador: Patrón de arquitectura de software que divide la lógica del negocio, los datos de una aplicación de la interfaz de usuario.

Naïve Bayes: Algoritmo basado en probabilidades para la clasificación de datos

Scrum: “Es un proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente, en equipo y obtener el mejor resultado posible de un proyecto”¹³

¹³ Tomado de <https://proyectosagiles.org/que-es-scrum>

22 Bibliografía

Raul Benitez, Gerardo Escudero, Samir Kanaan. Inteligencia artificial avanzada

Clasificador Naïve Bayes, Mayo de 2016. Disponible en la web:
<http://naivebayes.blogspot.com.co/>

Sistema para la prevención de la deserción de la educación superior.
Disponible en la web:
spadies.mineducacion.gov.co/spadies/consultas_predefinidas.html