

Servicio web para la consulta de la tendencia de aprobación de materias



**Diego Armando Rey Vanegas
cód.: 20162495013
email: diego.rey.vaneg@gmail.com
cel.: 3124903640**

**Universidad Distrital Francisco José de Caldas
Maestría en Ciencias de la Infomación y las Comunicaciones
Facultad de Ingeniería
Bogotá D.C.
Noviembre de 2017**

Resumen

Diferentes herramientas estadísticas están siendo usadas en áreas sociales y comerciales. La precisión en los resultados ha llevado a su amplia aceptación e implementación. El área de la educación, aunque es el lugar donde emerge la teoría, no ha sido abordado a fondo mediante aplicaciones.

Un estudio de aprobación de materias de estudiantes basado en observaciones e hipótesis de períodos previos se realiza en este documento. El método LDA(Linear Discriminant Analysis) [1] ha sido escogido para establecer la regla de decisión dada la naturaleza de las observaciones. Posteriormente, se aplica el método a datos de cuatro materias diferentes con buenos resultados.

Resultados de simulaciones y porcentajes extraídos de la aplicación del método permiten contemplar alguna acciones académicas respecto a cada materia, para lo cual se busca implementar una aplicación web que le permita a los estudiantes consultar la probabilidad de aprobar o reprobar una materia dadas sus notas del primer y segundo corte.

Tabla de Contenido

Resumen	I
Introducción	1
1. Planeación	2
1.1. Planteamiento del Problema	2
1.1.1. Información	2
1.2. Justificación	3
1.3. Objetivos	3
1.3.1. Objetivo General	3
1.3.2. Objetivos Específicos	3
1.4. Alcances	3
1.5. Limitaciones	4
1.5.1. Marco Metodológico	4
1.6. Factibilidad	6
1.6.1. Factibilidad Técnica	6
1.6.2. Factibilidad Operativa	6
1.6.3. Factibilidad Económica	7
1.6.4. Factibilidad Legal	7
1.7. Cronograma	7
2. Análisis	9
2.1. Análisis del Algoritmo de Clasificación	9
2.1.1. Regla de decisión	10
2.1.2. Simulación y Resultados	11
2.2. Análisis de Requerimientos	26
2.2.1. Diagrama de Contexto	27
2.2.2. Diagrama de Procesos	27
2.2.3. Requerimientos Funcionales	28
2.2.4. Requerimientos No Funcionales	29
2.3. Definición de Actores	30
3. Diseño	32
3.1. Casos de Uso	32
3.1.1. Diagramas de Casos de Uso	32

3.1.2. Documentación de Casos de Uso	33
3.2. Diagramas de Clases	34
3.3. Diagrama de Diseño Lógico de Persistencia	35
3.4. Diagramas de Actividades	35
3.5. Diagramas de Estados	36
3.6. Diagramas de Secuencia	36
3.7. Diagramas de Componentes	37
3.8. Diagramas de Despliegue	37
3.9. Diagramas de Arquitectura	37
4. Implementación y Pruebas	42
4.1. Servicio web rest y base de datos en Microsoft Azure®Cloud	42
4.2. Programación de aplicación en R para analítica de datos de materias	47
4.3. Despliegue de aplicación en Shinyapps.io Cloud	48
5. Conclusiones	62
Bibliography	64

Listas de Figuras

1.1. Cronograma	8
2.1. <i>Materia₁</i> : LDA, identificación de parámetros de guassiana de H_1	10
2.2. <i>Materia₁</i> : LDA para datos observados de periodos anteriores. En rojo, regla de decisión	11
2.3. <i>Materia₁</i> : datos actuales de prueba y regla de decisión	12
2.4. <i>Materia₁</i> : regla de decisión aplicada a datos actuales de prueba	12
2.5. <i>Materia₂</i> : LDA para datos observados de periodos anteriores. En rojo, regla de decisión	13
2.6. <i>Materia₂</i> : regla de decisión aplicada a datos actuales	13
2.7. <i>Materia₃</i> : LDA para datos observados de periodos anteriores. En rojo, regla de decisión	14
2.8. <i>Materia₃</i> : regla de decisión aplicada a datos de prueba actuales	14
2.9. <i>Materia₄</i> : LDA para datos observados de periodos anteriores. En rojo, regla de decisión	15
2.10. <i>Materia₄</i> : regla de decisión aplicada a datos actuales	15
2.11. Diagrama de Contexto	28
2.12. Diagrama de Procesos	31
3.1. Diagrama de Casos de Uso del Sistema	32
3.2. Diagrama de Clases	35
3.3. Diagrama Lógico de Persistencia	36
3.4. Diagrama de Actividades LDA	38
3.5. Diagrama de Estados	39
3.6. Diagrama de Secuencia Estudiante	39
3.7. Diagrama de Secuencia Administrador	40
3.8. Diagrama de Componentes	40
3.9. Diagrama de Despliegue	40
3.10. Diagrama de Arquitectura Negocio	41
3.11. Diagrama-de-Arquitectura-Aplicación	41
3.12. Diagrama-de-Arquitectura-tecnología	41
4.1. Opciones de suscripción en Azure	43
4.2. Panel de opciones Azure	44
4.3. Crear App Services	45
4.4. Menú de opciones completo	46
4.5. Selección de aplicación web y MySQL	47
4.6. Crear la aplicación	48
4.7. Parámetros de configuración	49

4.8. Opciones avanzadas	50
4.9. plan de pago base de datos	51
4.10. Configuración final	52
4.11. Credenciales de implementación	52
4.12. Datos de conexión base de datos Azure	53
4.13. Carque de base de datos por consola	53
4.14. FileZilla®, cargue de credenciales	54
4.15. FileZilla®, entorno de archivos	54
4.16. Prueba del web service, consumo por cliente POSTMAN®	55
4.17. Datos consumidos desde Web Service en Azure®	55
4.18. Prueba de la aplicación localmente	56
4.19. Cuenta en Shinyapps	56
4.20. Token en Shinyapps	57
4.21. Despliegue ShinyApp	58
4.22. Estado y consumo de la aplicación en <i>Shinyapps.io</i>	59
4.23. Inicio aplicación en ShinyApps.io	59
4.24. Histórico de tendencia de aprobación de materia 1	60
4.25. Clasificación de datos de prueba para materia 4	60
4.26. Clasificación individual de estudiante, pronóstico de tendencia <i>reprobado</i>	61
4.27. Clasificación individual de estudiante, pronóstico de tendencia <i>aprobado</i>	61

Lista de Tablas

1.1.	Actividades en Fases de RUP	4
1.2.	Costo de Integrantes del Proyecto	7
1.3.	Costo de Recursos	7
2.1.	Resultados basados en prueba de hipótesis LDA: estudiantes en periodo actual	16
2.2.	Requerimientos Funcionales	28
2.3.	Requerimientos No Funcionales	29
2.4.	Actores	30
3.1.	Documentación de Caso de Uso	33

Introducción

Si un estudiante supiera que tan probable es aprobar una materia en el examen final antes de ingresar a presentarlo, tal vez cambiaría la cantidad de tiempo invertido en la preparación del examen. Si antes de inscribir una materia se conociera una regla que indique una nota mínima en cada evaluación para tener una *buenas probabilidad* de aprobar la materia, tal vez cambiaría la cantidad de horas invertidas fuera del salón por el estudiante a lo largo del semestre.

Tener una buena aproximación de la cantidad de estudiantes que van a aprobar o reprobar una materia ayudaría a dimensionar la carga docente del siguiente periodo académico o modificar la distribución de planta física. También, podrían programarse actividades correctivas como cursos vacacionales, tutorías intersemestrales o cursos cortos de refuerzo.

Capítulo 1

Planeación

1.1. Planteamiento del Problema

Ciertos estudiantes de una IES¹ son evaluados en sus materias mediante tres notas. Las dos primeras notas, N_1 y N_2 , son valoradas cada una con un 30 % y una nota final o examen final N_3 correspondiente al 40 % de la nota definitiva N_d de la materia. La nota definitiva se calcula como se muestra en (1.1)

$$N_d = \frac{3}{10}N_1 + \frac{3}{10}N_2 + \frac{4}{10}N_3 \quad (1.1)$$

La escala de evaluación es [0, 50] y se define así:

- Si $N_d \geq 30$ se considera que el estudiante aprobó la materia.
- Si $N_d < 30$ se considera que el estudiante reprobó la materia.

Se desea averiguar antes de ingresar al examen final, si con *buenas probabilidades*² los estudiantes de un semestre (periodo actual de prueba) aprueban o reaprueban una materia.

1.1.1. Información

Se dispone de información sobre varias materias en tres períodos académicos previos³ y del *periodo académico de prueba* 2017 – 1. En los cuatro períodos se conocen las notas N_1 y N_2 y en los tres períodos académicos previos se conoce N_d de cada estudiante que toma o tomó cada materia.

¹Institución de Educación Superior

²Resultado basado en prueba de hipótesis

³Períodos académicos: 2015 – 2, 2016 – 1 y 2016 – 2

1.2. Justificación

Es válido resolver el problema de pronóstico de estudiantes aprobados y reprobados por materia con un aplicativo web que realice la clasificación, debido a la necesidad de esta herramienta en la Institución de Educación Superior que sirve como caso de estudio para este proyecto.

Con base en datos reales de notas de materias cursadas por estudiantes en varios semestres se pretende realizar un modelamiento matemático para definir un clasificador sencillo, donde se realicen simulaciones en MATLAB®, para validar el análisis.

1.3. Objetivos

1.3.1. Objetivo General

Diseñar y desarrollar un prototipo de software basado en herramientas *Open Source* y arquitectura orientada a servicios para ofrecer un servicio web de consulta de la tendencia y probabilidad de aprobación de una materia a estudiantes de una Institución de Educación Superior conociendo sus dos notas de exámenes parciales.

1.3.2. Objetivos Específicos

- Desarrollar una aplicación web utilizando tecnología de código abierto para implementar el módulo de evaluación de tendencia de aprobación por materia, para una Institución de Educación Superior.
- Realizar pruebas de funcionamiento para validar el modelo y el desempeño del software de aplicación.

1.4. Alcances

El proyecto podrá ofrecer a los estudiantes de una Institución de Educación Superior una herramienta de consulta web, con la cual puedan obtener una estimación de la tendencia y probabilidad con que pueden aprobar una materia que estén cursando o quieran cursar, y así tener idea acerca del esfuerzo académico necesario a destinar para aprobar la materia de interés.

Además la herramienta web puede ser utilizada por el personal encargado de la planeación de recursos académicos para realizar una estimación de la necesidad de cupos en salones para estudiantes repitentes para el próximo periodo académico.

1.5. Limitaciones

El proyecto no ofrecerá un resultado exacto, ya que se realiza una estimación con base en las calificaciones de materias obtenidas en semestres anteriores y almacenados en la base de datos de la Institución de Educación Superior, lo cual hace al proceso estocástico[2], al haber aleatoriedad en las notas de exámenes y variación en el tiempo.

1.5.1. Marco Metodológico

La metodología RUP *Rational Unified Process* [3] es un proceso de desarrollo de software basado en UML[4]. Este proceso es una metodología para el análisis, implementación y documentación de aplicaciones de software basadas en el paradigma de orientación a objetos.

RUP se divide en cuatro fases que se llevan a cabo mediante iteraciones.

1. Fase de Inicio. Se establece las características del producto así como sus alcances mediante el análisis de los requerimientos.
2. Fase de Elaboración. Se planifican las actividades necesarias. Se lleva a cabo el diseño del producto mediante la creación de diferentes diagramas estructurales y de comportamiento.
3. Fase de Construcción. Se construye el prototipo de la aplicación totalmente operativo. Se elabora manuales de instalación y usuario. Se realizan pruebas unitarias del prototipo.
4. Fase de Transición. Se realiza la instalación del producto en el cliente así como la capacitación de los usuarios finales. La transición del producto a los usuarios también incluye el soporte y mantenimiento del producto.

La Tabla 1.1 presenta las diferentes actividades realizadas con base en las fases de la metodología RUP.

Tabla 1.1: Actividades en Fases de RUP

Etapa	Fase	Actividad	Descripción
Análisis	Inicio	Modelamiento del negocio	Permite definir el flujo de eventos en los procesos de negocio que la aplicación debe resolver
		Modelo de procesos	Presenta mediante lenguaje BPMN los procesos de negocio
	Elaboración	Requerimientos funcionales	Establece los diferentes requerimientos del sistema que determinan los servicios que la aplicación debe ofrecer
		Requerimientos no funcionales	Especifica las características adicionales en el sistema para que los servicios determinados por los requerimientos funcionales, puedan llevarse a cabo

	Definición de Actores	Determina la lista de actores del sistema
Construcción		
Transición		
Diseño	Inicio	Lista inicial de casos de uso Permite definir los casos de uso básicos que tiene el sistema
		Documentación de casos de uso Presenta todas las características de los casos de uso definidos para el sistema
	Elaboración	Diagramas casos de uso Presenta en notación UML los casos de uso del sistema permitiendo la agrupación de casos de uso mediante relaciones de generalización y especialización
		Diagrama de clases Presenta la estructura en la capa de negocio del proyecto
		Diagrama de secuencia Muestra el flujo de eventos de cada caso de uso documentado
		Diagrama de componentes Presenta la estructura general del proyecto
		Diagrama de despliegue Presenta la interacción de los diferentes dispositivos del proyecto y la forma de ser usado por los actores del sistema
Construcción	Configuración de equipos de desarrollo	Se configuran servidores de aplicaciones, motores de bases de datos entre otros
	Instalación de frameworks	Se instala en los equipos de desarrollo los frameworks que serán utilizados en el proyecto
	Transición	
Implementación	Inicio	Diccionario de datos Establece las tablas, relaciones y tipos de datos de la persistencia del proyecto
	Elaboración	Modelo relacional Presenta la estructura de la persistencia del proyecto
	Construcción	Creación de la base de datos Se configuran servidores de aplicaciones, motores de bases de datos entre otros
		Creación de clases de negocio Se crean las clases del proyecto en la capa de lógica de negocio
		Creación de interfaz gráfica de usuario Se crea el diseño gráfico y las clases para conectar la interfaz gráfica con la lógica de negocio
		Creación de persistencia Se crean las clases para llevar a cabo conexiones con bases de datos
	Transición	Despliegue de la versión inicial de la aplicación Se publican en el servidor de aplicaciones los componentes desarrollados
Pruebas	Inicio	

	Elaboración	
Construcción	Pruebas unitarias	Se realizan pruebas de cada uno de los modulas de la aplicación
Transición	Pruebas globales	Se realizan pruebas de toda la aplicación considerando todos los requerimientos funcionales y no funcionales del sistema

1.6. Factibilidad

1.6.1. Factibilidad Técnica

Para el desarrollo e implantación del proyecto es necesario contar con los siguientes dispositivos:

- Equipo de computo para desarrollo de aplicaciones web.
- Servicio de hosting (opcional sujeto a requerimiento del cliente) con soporte para servidor de aplicaciones Apache, bases de datos PostgreSQL y servidor RStudio Server.

Así mismo, es necesario contar con los siguientes recursos de software

- Editor de código Notepad++, Sublime Text 3, Visual Studio Code
- Servidor Apache para aplicaciones
- Servidor PostgreSQL para almacenamiento de datos
- Servidor RStudio para ejecución de los algoritmos de analítica a utilizar

1.6.2. Factibilidad Operativa

El proyecto será elaborado por el estudiante de maestría en Ciencias de la Información y las Comunicaciones Diego Armando Rey Vanegas, el cual cuenta con el conocimiento tecnológico para desarrollar el proyecto. Adicionalmente, el ingeniero Jose Nelson Pérez Castillo será el encargado de brindar asesoría a lo largo del proyecto.

1.6.3. Factibilidad Económica

El costo del proyecto se presenta con base en: 1) el costo de los equipos requeridos para la elaboración del diseño y desarrollo del prototipo que requiere el proyecto y 2) el costo de mano de obra de los integrantes. La Tabla 1.2 presenta el costo total de los integrantes del proyecto. La Tabla 1.3 presenta el costo total de los demás recursos requeridos para elaborar el proyecto.

Tabla 1.2: Costo de Integrantes del Proyecto

Integrante	Valor hora	Horas	Valor total
Estudiante 1	\$20.000	192	\$3.840.000
Jose Nelson Pérez	\$40.000	32	\$1.280.000
Total			\$5.120.000

Tabla 1.3: Costo de Recursos

Recurso	Cantidad	Valor Unitario	Valor total
Equipos de Computo	1	\$2.500.000	\$2.500.000
Internet	1	\$50.000	\$50.000
Hosting	1	\$100.000	\$100.000
Total			\$2.650.000

1.6.4. Factibilidad Legal

Se utiliza licencias de código abierto para el software utilizado en el desarrollo de la aplicación.

1.7. Cronograma

La Figura 1.1 presenta el cronograma con las fechas de inicio y fin establecidas con el profesor en la materia.

Figura 1.1: Cronograma



Capítulo 2

Análisis

2.1. Análisis del Algoritmo de Clasificación

Con base en datos reales de notas de materias cursadas por estudiantes en varios semestres se realizó un modelamiento matemático para definir un clasificador por regla de decisión y se realizaron simulaciones en MATLAB® y en R®, para validar el análisis, posteriormente en la etapa de implementación se realizará la implementación en lenguaje R, el cual es usado por el servidor de RStudio®, para realizar el procesamiento de los algoritmos.

Para determinar la regla de decisión se asume que los datos x_1 y x_2 se distribuyen normalmente (2.1)

$$\begin{aligned} \mathbf{X} &\sim \mathcal{N}(\mu_1, Q_T), \text{ para } H_1 \\ \mathbf{X} &\sim \mathcal{N}(\mu_2, Q_T), \text{ para } H_2 \end{aligned} \tag{2.1}$$

Se hizo uso de la técnica LDA¹. Buscando encontrar una buena regla de decisión [1], por tanto, se realizó la prueba con los datos de la *materia₁* como se observa en la Fig. 2.1.

Como se observa en la Fig. 2.1 (nóte las escalas de los ejes), LDA es un método suficiente para establecer una buena regla de decisión dada la naturaleza del proceso (es matemáticamente imposible lograr aprobar la materia con ciertos valores de N_1 y N_2). LDA es el método utilizado para la prueba de hipótesis de cada materia.

¹Linear Discriminant Analysis

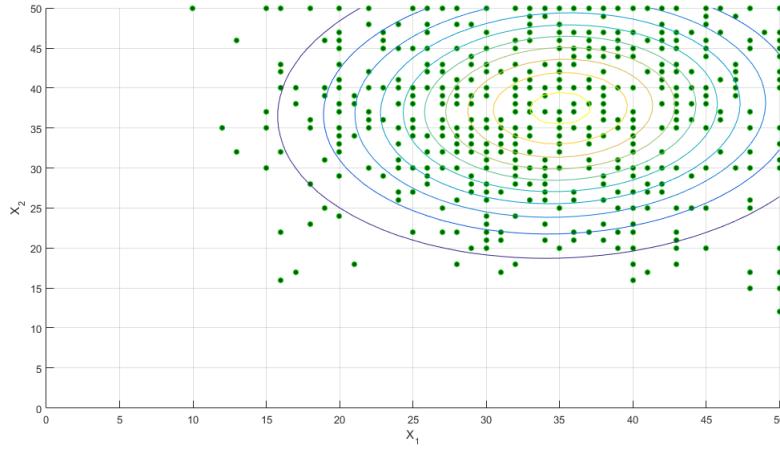


Figura 2.1: *Materia₁*: LDA, identificación de parámetros de guassiana de H_1

2.1.1. Regla de decisión

Para obtener la regla de decisión mediante el método LDA, se obtienen los valores de μ_1 y Q_1 para la hipótesis H_1 y μ_2 y Q_2 para la hipótesis H_2 por materia a partir de los datos de los períodos previos.

Se establece una matriz de covarianza promedio en (2.2) para facilitar la obtención de la regla de decisión[2].

$$Q_T = \frac{Q_1 + Q_2}{2} \quad (2.2)$$

La recta que define la regla de decisión está dada por (2.3)

$$w^\top x = c \quad (2.3)$$

Con w dado por (2.4) y c dado por (2.5)

$$w = (\mu_1 - \mu_2)^\top Q_T^{-1} \quad (2.4)$$

$$c = \frac{1}{2} (\mu_1^\top Q_T^{-1} \mu_1 - \mu_2^\top Q_T^{-1} \mu_2) \quad (2.5)$$

Una vez hallada la regla de decisión, se aplicó a cada uno de los estudiantes que en el periodo de prueba actual estaban cursando la materia. Se extraen los resultados para cada materia, generando los porcentajes de aprobación y reprobación en base a la prueba de hipótesis.

2.1.2. Simulación y Resultados

Se evaluaron cuatro materias en tres períodos académicos previos y el periodo actual como se describió anteriormente. Las materias presentan distintos programas académicos y niveles de dificultad, el cual es indicado por la cantidad de créditos académicos.²

Simulación MATLAB®

La Fig. 2.2 muestra para la *materia₁* los datos de cada hipótesis, las curvas de nivel de las distribuciones que se asumen en LDA para cada hipótesis, y la recta que define la regla de decisión.

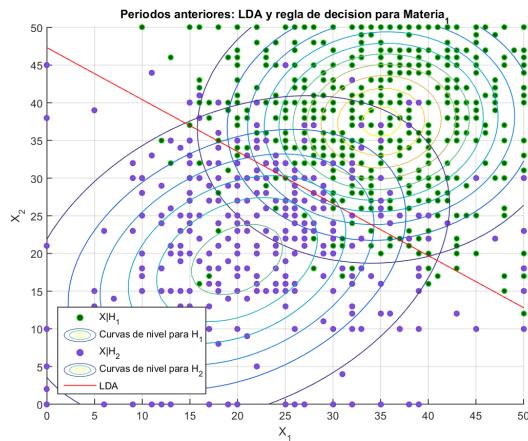


Figura 2.2: *Materia₁*: LDA para datos observados de períodos anteriores. En rojo, regla de decisión

La Fig. 2.3 muestra la regla de decisión para las notas N_1 y N_2 de estudiantes en el periodo actual de prueba para la *materia₁*.

En la Fig. 2.4 se observa la regla de decisión aplicada a las notas N_1 y N_2 de estudiantes en el periodo actual de prueba para la *materia₁*.

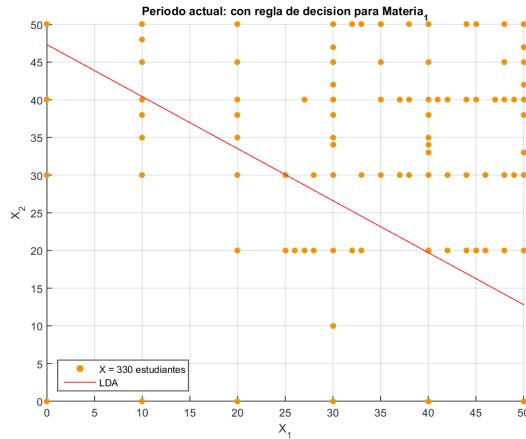
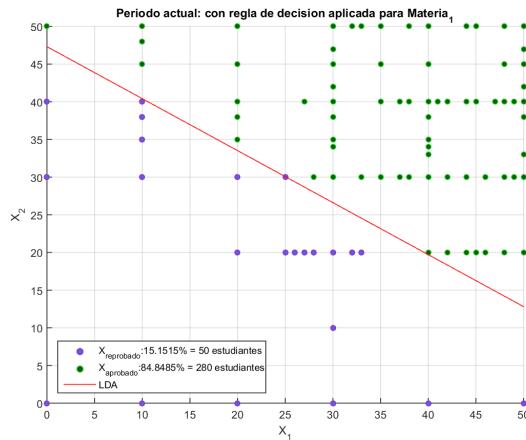
Las Fig. 2.5 y 2.6 muestran el método LDA para la *materia₂* y su aplicación a los datos del periodo actual de prueba.

Las Fig. 2.7 y 2.8 muestran el método LDA para la *materia₃* y su aplicación a los datos del periodo actual de prueba.

Las Fig. 2.9 y 2.10 muestran el método LDA para la *materia₄* y su aplicación a los datos del periodo actual de prueba.

Tabla 2.1 resumen de los resultados de las simulaciones.

²El crédito académico indica la cantidad de horas semanales que debe dedicar un estudiante fuera del aula por cada hora tomada en el aula semanalmente.

Figura 2.3: *Materia₁*: datos actuales de prueba y regla de decisiónFigura 2.4: *Materia₁*: regla de decisión aplicada a datos actuales de prueba

Según la Tabla 2.1 se pueden contemplar las siguientes acciones:

- Los estudiantes en cada materia que se encuentren bajo la regla de decisión, marcados en azul, es altamente probable que pierdan la materia. Por tanto, deberían invertir gran cantidad de tiempo en el estudio de la materia para *intentar* revertir el resultado final. Tener en cuenta que algunos, matemáticamente ya han reprobado la materia aún sin conocer la nota del examen final N_3 .
- Para las materias 1, 2 y 4 es posible dar apertura a cursos vacacionales o intersemestrales.
- Para las materias 3 y 4 se pueden plantear tutorías o refuerzos a lo largo del semestre dado el porcentaje elevado de estudiantes que reprueban la materia.
- Para la materia 3 se deben dimensionar adecuadamente el salón para el siguiente periodo, ya que se requiere un 24,07 % extra en capacidad.

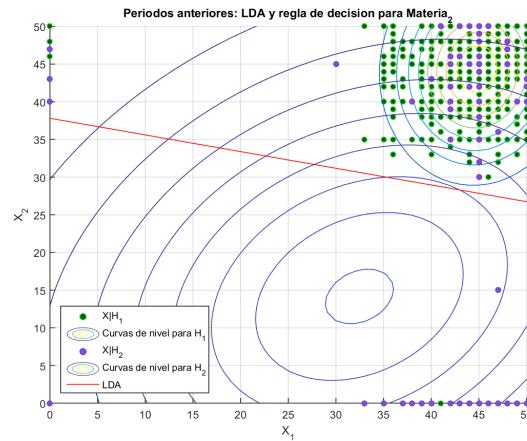


Figura 2.5: *Materia₂*: LDA para datos observados de periodos anteriores. En rojo, regla de decisión

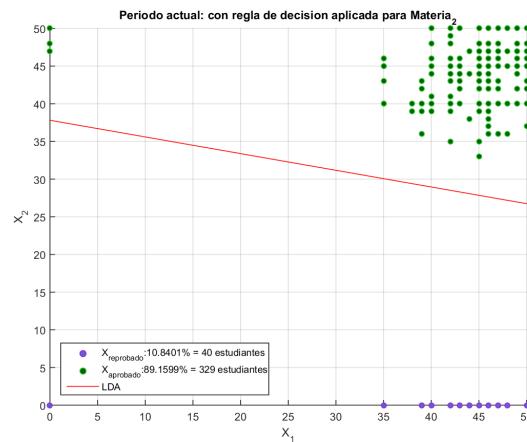


Figura 2.6: *Materia₂*: regla de decisión aplicada a datos actuales

Simulación R(R)

A continuación se codifica y se prueban los cálculos en lenguaje R(R):

```
#####
#####PREÁMBULO#####
rm(list = ls()) #instrucción para limpiar la memoria de trabajo
library(utils) #cargue de la librería "utils" para utilizar la función de importación de datos
library(data.table) #cargue de la librería "data.table" para manejo de datos tipo data.frame
#####
#####CARGUE DE DATOS LOCAL#####
#1. Carga de datos: X_[materia]_[periodo]_h[hipótesis]
ruta="~/Escritorio/DataMateriasCsv/" #ruta de las notas de las materias en equipo local
#ruta="DataMateriasCsv/" #Ruta para ejecución con Knitr, carpeta en mismo directorio
```

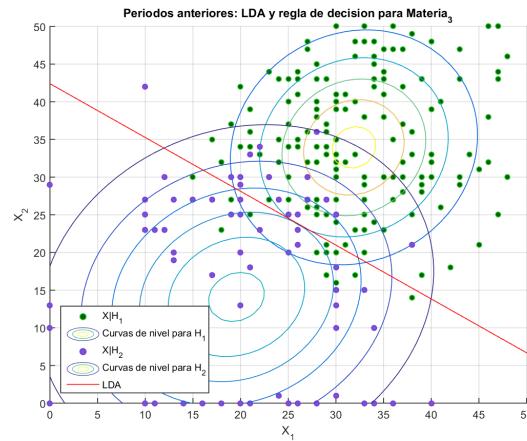


Figura 2.7: *Materia₃*: LDA para datos observados de periodos anteriores. En rojo, regla de decisión

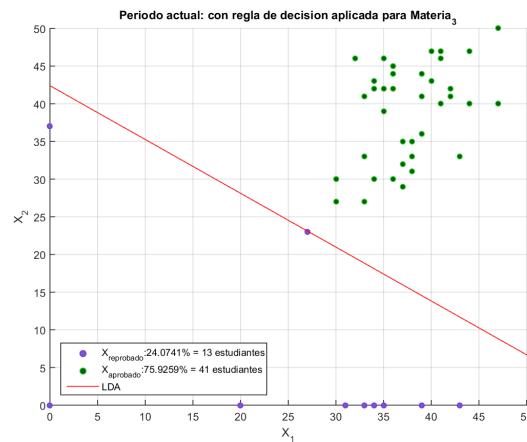


Figura 2.8: *Materia₃*: regla de decisión aplicada a datos de prueba actuales

```
#Materia 1 --> se importan los archivos .csv de esta materia y se cargan en memoria de Rstuds
#H1 --> Hipótesis 1 = estudiantes aprobados
X_1_20151_h1=read.csv(paste0(ruta, "X_1_20151_h1.csv"), header = FALSE)
X_1_20152_h1=read.csv(paste0(ruta, "X_1_20152_h1.csv"), header = FALSE)
X_1_20161_h1=read.csv(paste0(ruta, "X_1_20161_h1.csv"), header = FALSE)
#H2 --> Hipótesis 2 = estudiantes reprobados
X_1_20151_h2=read.csv(paste0(ruta, "X_1_20151_h2.csv"), header = FALSE)
X_1_20152_h2=read.csv(paste0(ruta, "X_1_20152_h2.csv"), header = FALSE)
X_1_20161_h2=read.csv(paste0(ruta, "X_1_20161_h2.csv"), header = FALSE)
#Hx --> Datos de prueba para clasificar
X_1_20162=read.csv(paste0(ruta,"X_1_20162.csv"), header = FALSE)
#Materia 2 --> se importan los archivos .csv de esta materia y se cargan en memoria de Rstuds
#H1 --> Hipótesis 1 = estudiantes aprobados
X_2_20151_h1=read.csv(paste0(ruta, "X_2_20151_h1.csv"), header = FALSE)
```

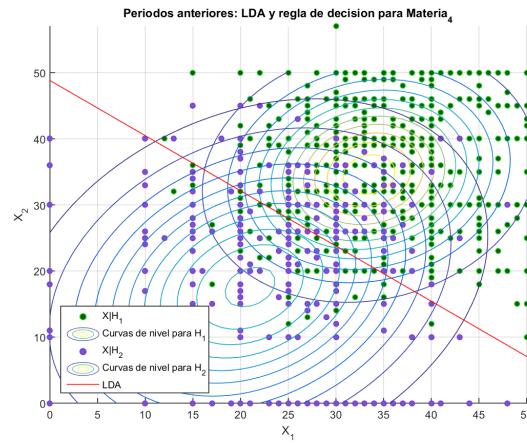


Figura 2.9: *Materia₄*: LDA para datos observados de periodos anteriores. En rojo, regla de decisión

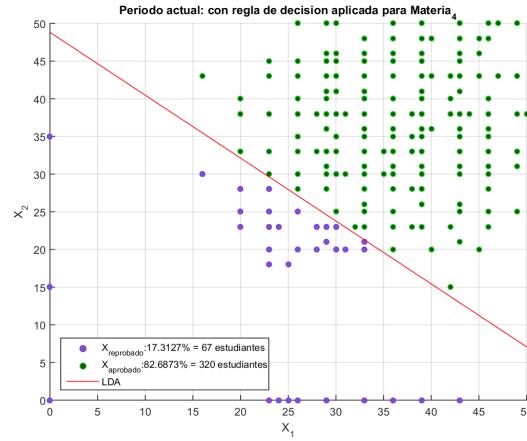


Figura 2.10: *Materia₄*: regla de decisión aplicada a datos actuales

```
X_2_20152_h1=read.csv(paste0(ruta, "X_2_20152_h1.csv"), header = FALSE)
X_2_20161_h1=read.csv(paste0(ruta, "X_2_20161_h1.csv"), header = FALSE)
#H2 --> Hipótesis 2 = estudiantes reprobados
X_2_20151_h2=read.csv(paste0(ruta, "X_2_20151_h2.csv"), header = FALSE)
X_2_20152_h2=read.csv(paste0(ruta, "X_2_20152_h2.csv"), header = FALSE)
X_2_20161_h2=read.csv(paste0(ruta, "X_2_20161_h2.csv"), header = FALSE)
#Hx --> Datos de prueba para clasificar
X_2_20162=read.csv(paste0(ruta, "X_2_20162.csv"), header = FALSE)
#Materia 3 --> se importan los archivos .csv de esta materia y se cargan en memoria de Rstu
#H1 --> Hipótesis 1 = estudiantes aprobados
X_3_20151_h1=read.csv(paste0(ruta, "X_3_20151_h1.csv"), header = FALSE)
X_3_20152_h1=read.csv(paste0(ruta, "X_3_20152_h1.csv"), header = FALSE)
X_3_20161_h1=read.csv(paste0(ruta, "X_3_20161_h1.csv"), header = FALSE)
#H2 --> Hipótesis 2 = estudiantes reprobados
```

Materia	Estudiantes	Aprobados	Reprobados
1	330	280 (84,85 %)	50 (15,15 %)
2	369	329 (89,16 %)	40 (10,84 %)
3	54	41 (75,93 %)	13 (24,07 %)
4	387	320 (82,69 %)	67 (17,31 %)

Tabla 2.1: Resultados basados en prueba de hipótesis LDA: estudiantes en periodo actual

```

X_3_20151_h2=read.csv(paste0(ruta, "X_3_20151_h2.csv"), header = FALSE)
X_3_20152_h2=read.csv(paste0(ruta, "X_3_20152_h2.csv"), header = FALSE)
X_3_20161_h2=read.csv(paste0(ruta, "X_3_20161_h2.csv"), header = FALSE)
#Hx --> Datos de prueba para clasificar
X_3_20162=read.csv(paste0(ruta, "X_3_20162.csv"), header = FALSE)
#Materia 4 --> se importan los archivos .csv de esta materia y se cargan en memoria de Rstuds
#H1 --> Hipótesis 1 = estudiantes aprobados
X_4_20151_h1=read.csv(paste0(ruta, "X_4_20151_h1.csv"), header = FALSE)
X_4_20152_h1=read.csv(paste0(ruta, "X_4_20152_h1.csv"), header = FALSE)
X_4_20161_h1=read.csv(paste0(ruta, "X_4_20161_h1.csv"), header = FALSE)
#H2 --> Hipótesis 2 = estudiantes reprobados
X_4_20151_h2=read.csv(paste0(ruta, "X_4_20151_h2.csv"), header = FALSE)
X_4_20152_h2=read.csv(paste0(ruta, "X_4_20152_h2.csv"), header = FALSE)
X_4_20161_h2=read.csv(paste0(ruta, "X_4_20161_h2.csv"), header = FALSE)
#Hx --> Datos de prueba para clasificar
X_4_20162=read.csv(paste0(ruta, "X_4_20162.csv"), header = FALSE)
##
#####
#####VARIABLES GLOBALES#####
min_data = 0 #nota mínima
max_data = 50 #nota máxima
step_data = 1 #paso de incremento para las notas
##
#####
#####ANÁLISIS DE DISCRIMINANTE LINEAL#####
#####se programa la cantidad de iteraciones según el número de materias disponibles#####
for(materia_i in 1:4){
  if(materia_i==1){ # Selección de datos según materia para cargar matrices X_h1, X_h2 y X_x
    X_h1 = rbind(X_1_20151_h1, X_1_20152_h1, X_1_20161_h1) #concatenar los vectores hipótesis 1
    X_h2 = rbind(X_1_20151_h2, X_1_20152_h2, X_1_20161_h2) #concatenar los vectores hipótesis 2
    X_x=X_1_20162
  }else{
    if(materia_i==2){ # Selección de datos según materia para cargar matrices X_h1, X_h2 y X_x
      X_h1 = rbind(X_2_20151_h1, X_2_20152_h1, X_2_20161_h1) #concatenar los vectores hipótesis 1
      X_h2 = rbind(X_2_20151_h2, X_2_20152_h2, X_2_20161_h2) #concatenar los vectores hipótesis 2
      X_x=X_2_20162
    }else{

```

```

if(materia_i==3){ # Selección de datos según materia para cargar matrices X_h1, X_h2 y X_x
X_h1 = rbind(X_3_20151_h1, X_3_20152_h1, X_3_20161_h1) #concatenar los vectores hipótesis 1
X_h2 = rbind(X_3_20151_h2, X_3_20152_h2, X_3_20161_h2) #concatenar los vectores hipótesis 2
X_x=X_3_20162
}else{
X_h1 = rbind(X_4_20151_h1, X_4_20152_h1, X_4_20161_h1) #concatenar los vectores hipótesis 1
X_h2 = rbind(X_4_20151_h2, X_4_20152_h2, X_4_20161_h2) #concatenar los vectores hipótesis 2
X_x=X_4_20162
}
}
}
}

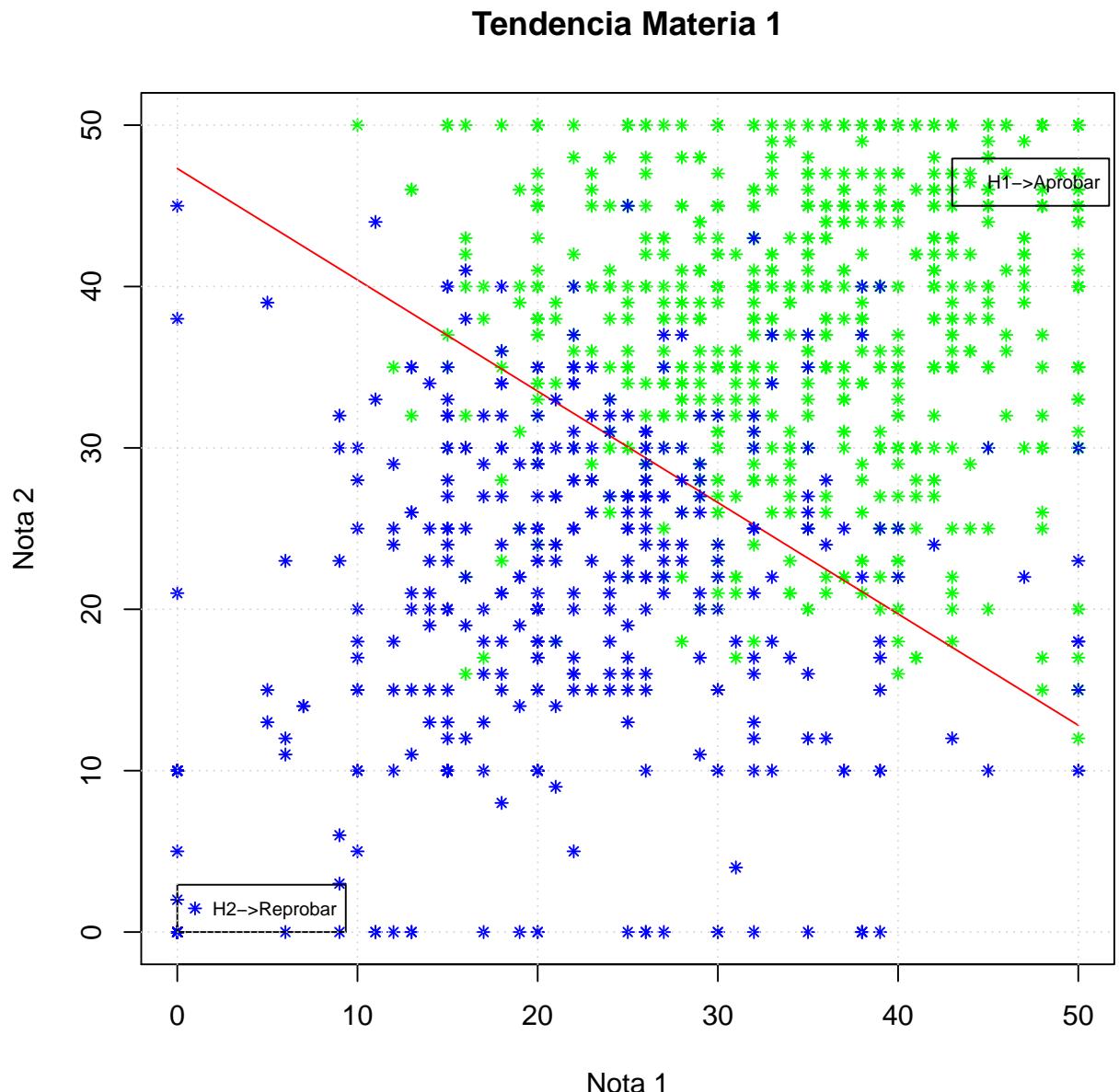
#####
#####Cálculo de las esperanzas de las variables aleatorias#####
mu_X_h1 =c(mean(X_h1$V1), mean(X_h1$V2)) #cálculo de la media por cada vector de Nota1 y Nota2
mu_X_h1=as.data.frame(mu_X_h1) #conversión de la matriz a tipo de dato data.frame
mu_X_h1=t(mu_X_h1) #cálculo de la matriz transpuesta
mu_X_h2 =c(mean(X_h2$V1), mean(X_h2$V2)) #cálculo de la media por cada vector de Nota1 y Nota2
mu_X_h2=as.data.frame(mu_X_h2) #conversión de la matriz a tipo data.frame
mu_X_h2=t(mu_X_h2) #cálculo de la matriz transpuesta
#####
#####Cálculo de las matrices de covarianza#####
Q_X_h1 =cov(X_h1) #covarianza H1
Q_X_h2 =cov(X_h2) #covarianza H2
#Matriz de covarianza promedio total
Q_X_T = (Q_X_h1+Q_X_h2)/2
#####
#####Cálculo de coeficientes regla de decisión#####
w = (mu_X_h1-mu_X_h2)%*%solve(Q_X_T) #cálculos para encontrar matriz de coeficientes W
w=as.vector(w) #conversión a tipo de dato vector
c = (mu_X_h1%*%solve(Q_X_T)%*%t(mu_X_h1) - mu_X_h2%*%solve(Q_X_T)%*%t(mu_X_h2))/2 #cálculos
c=as.vector(c) #conversión a tipo de dato vector
x1_LDA = c(min_data:max_data) #conjunto de puntos eje X
x2_LDA = ((-w[1]/w[2])*x1_LDA)+(c/w[2]) #conjunto de puntos eje Y
#####
#####Grafica de tendencia de materia histórico y regla de decisión#####
plot(X_h1$V1, X_h1$V2, xlab = "Nota 1", ylab = "Nota 2", pch=8, col="green", cex=0.7 , xlim=x1_LDA, ylim=x2_LDA)
lines(x1_LDA, x2_LDA ,col="red") #dibuja la línea del discriminante lineal
points(X_h2$V1, X_h2$V2, pch=8, col="blue", cex=0.7) #dibuja los puntos de las hipótesis h2
title(main=paste0('Tendencia Materia ',as.character(materia_i))) #coloca el título a la gráfica
legend(43, 45, #establecer los parámetros de la leyenda de la hipótesis 1
       legend = c(paste0("H1->Aprobar")), #mensaje de aprobar
       cex=0.7, #anchura del punto
       pch=8, col="green", #tipo y color de punto
       xjust = 0, yjust = 0) #ajuste de coordenadas de visualización
legend(0, 0, #establecer los parámetros de la leyenda de la hipótesis 2
       legend = c(paste0("H2->Reprobar")), #mensaje de reprobar
       cex=0.7, #anchura del punto
       pch=8, col="blue", #tipo y color de punto
       xjust = 0, yjust = 0) #ajuste de coordenadas de visualización

```

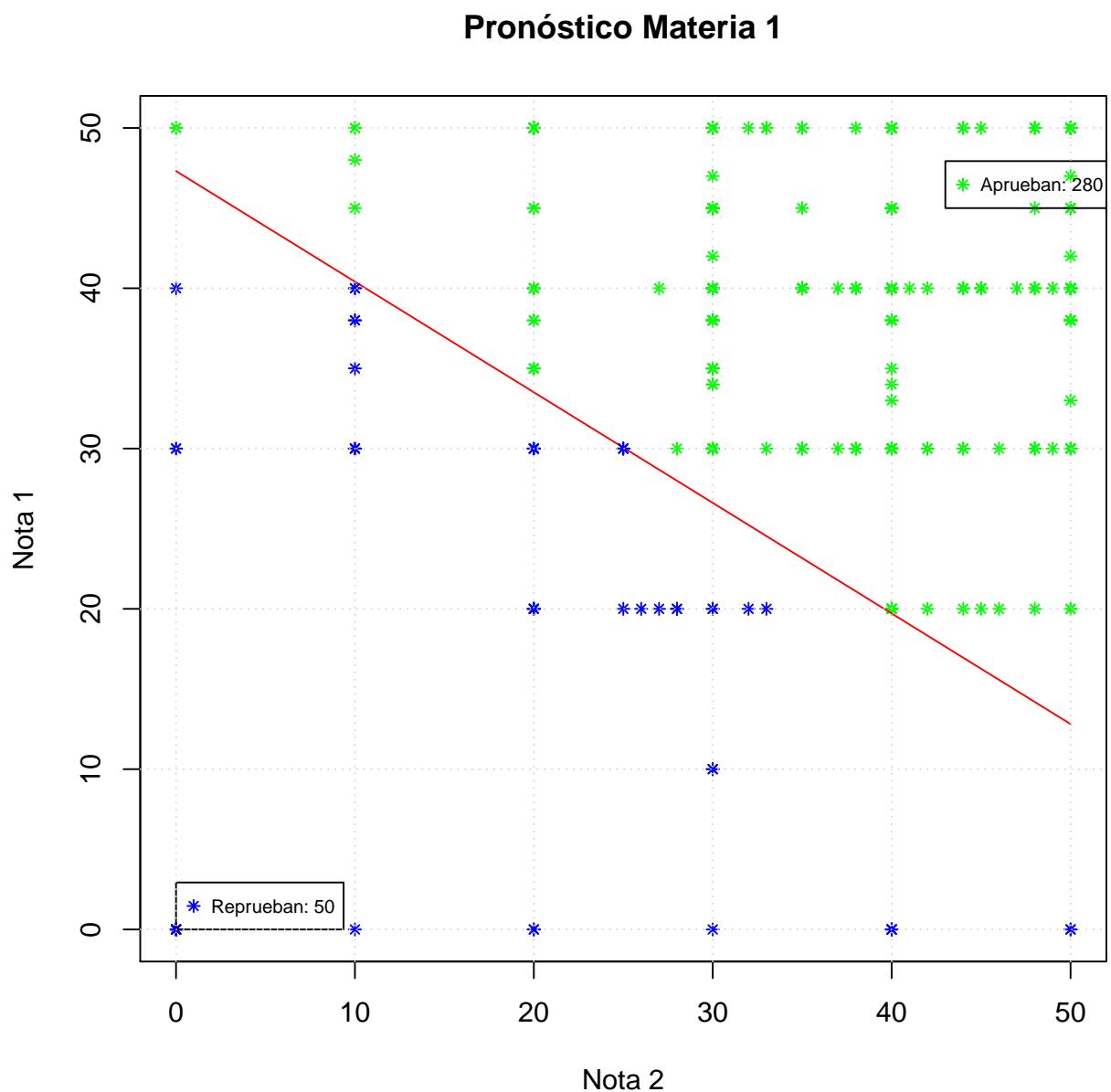
```

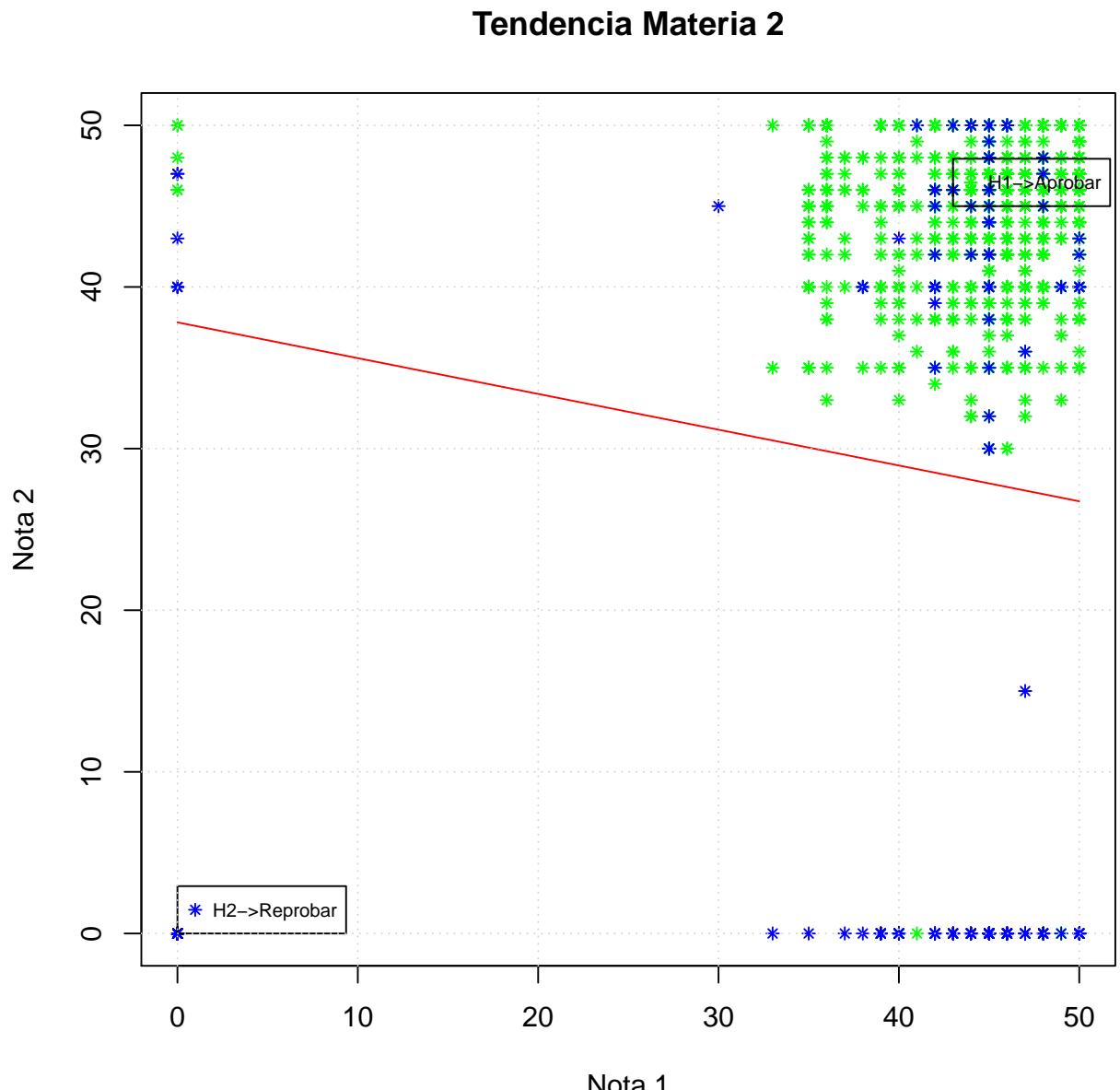
grid() #dibujar la grilla en la gráfica
#####
#####Evaluacion de datos actuales con regla de decisión#####
X_x_h1 = c() #inicializa el vector hipótesis 1 en vacío
X_x_h2 = c() #inicializa el vector hipótesis 2 en vacío
X_regla = ((-w[1]*t(X_x[,1])/w[2])+c/w[2])-t(X_x[,2]) #calculo de la diferencia entre los pu
row_X_xR=nrow(X_regla) #número de filas del vector de diferencias
col_X_xR=ncol(X_regla) #número de columnas del vector de diferencias
for(i in 1:col_X_xR){
  if(X_regla[i]>=0){ # si dif>=0 => H1->Reprobado
    X_x_h2 = rbind(X_x_h2, c(X_x[i,1], X_x[i,2]))#--Estudiantes Reprobados
  }else{
    X_x_h1 = rbind(X_x_h1, c(X_x[i,1], X_x[i,2]))#--Estudiantes Aprobados
  }
}
row_X_x_h1=nrow(X_x_h1) #Cantidad estudiantes Aprobados
row_X_x_h2=nrow(X_x_h2) #Cantidad estudiantes Reprobados
row_X_x=nrow(X_x) #Cantidad total de estudiantes
p_X_x_h1 = 100*row_X_x_h1/row_X_x #Porcentaje estudiantes Aprobados
p_X_x_h2 = 100*row_X_x_h2/row_X_x #Porcentaje estudiantes Reprobados
#####
#####Impresión de resultados en pantalla#####
print(paste0("Materia: ", as.character(materia_i))) #imprimir el nombre de la materia
print(paste0("Estudiantes aprobados: ", as.character(row_X_x_h1))) #imprimir cantidad estudiantes aprobados
print(paste0("Estudiantes reprobados: ", as.character(row_X_x_h2))) #imprimir cantidad estudiantes reprobados
print(paste0("Porcentaje estudiantes aprobados: ", as.character(p_X_x_h1))) #imprimir porcentaje de estudiantes aprobados
print(paste0("Porcentaje estudiantes reprobados: ", as.character(p_X_x_h2))) #imprimir porcentaje de estudiantes reprobados
print("//////////") #imprime separador de mensajes entre materia
#####
#####Grafica pronóstico total de materia#####
#imprime puntos de los datos clasificados como aprobados en color verde
plot(X_x_h1[,1], X_x_h1[,2], xlab = "Nota 2", ylab = "Nota 1", pch=8, col="green", cex=0.7 , lines(x1_LDA, x2_LDA ,col="red") #imprime la recta del discriminante lineal en color rojo
points(X_x_h2[,1], X_x_h2[,2], pch=8, col="blue", cex=0.7) #imprime los puntos de los datos
title(main=paste0('Pronóstico Materia ',as.character(materia_i))) #imprime encabezado de la grafica
legend(43, 45, #imprime etiqueta para hipótesis de aprobado en color verde
       legend = c(paste0("Aprueban: ",as.character(row_X_x_h1))), #imprime en la etiqueta la cantidad de estudiantes aprobados
       cex=0.7, #anchura del punto
       pch=8, col="green", #imprime símbolo de aprobados con color verde
       xjust = 0, yjust = 0)
legend(0, 0, #imprime etiqueta para hipótesis de reprobado en color azul
       legend = c(paste0("Reprueban: ",as.character(row_X_x_h2))), #imprime en la etiqueta la cantidad de estudiantes reprobados
       cex=0.7, #anchura del punto
       pch=8, col="blue", #imprime símbolo de reprobados con color azul
       xjust = 0, yjust = 0)
grid() #dibujar la grilla en la gráfica
}

```

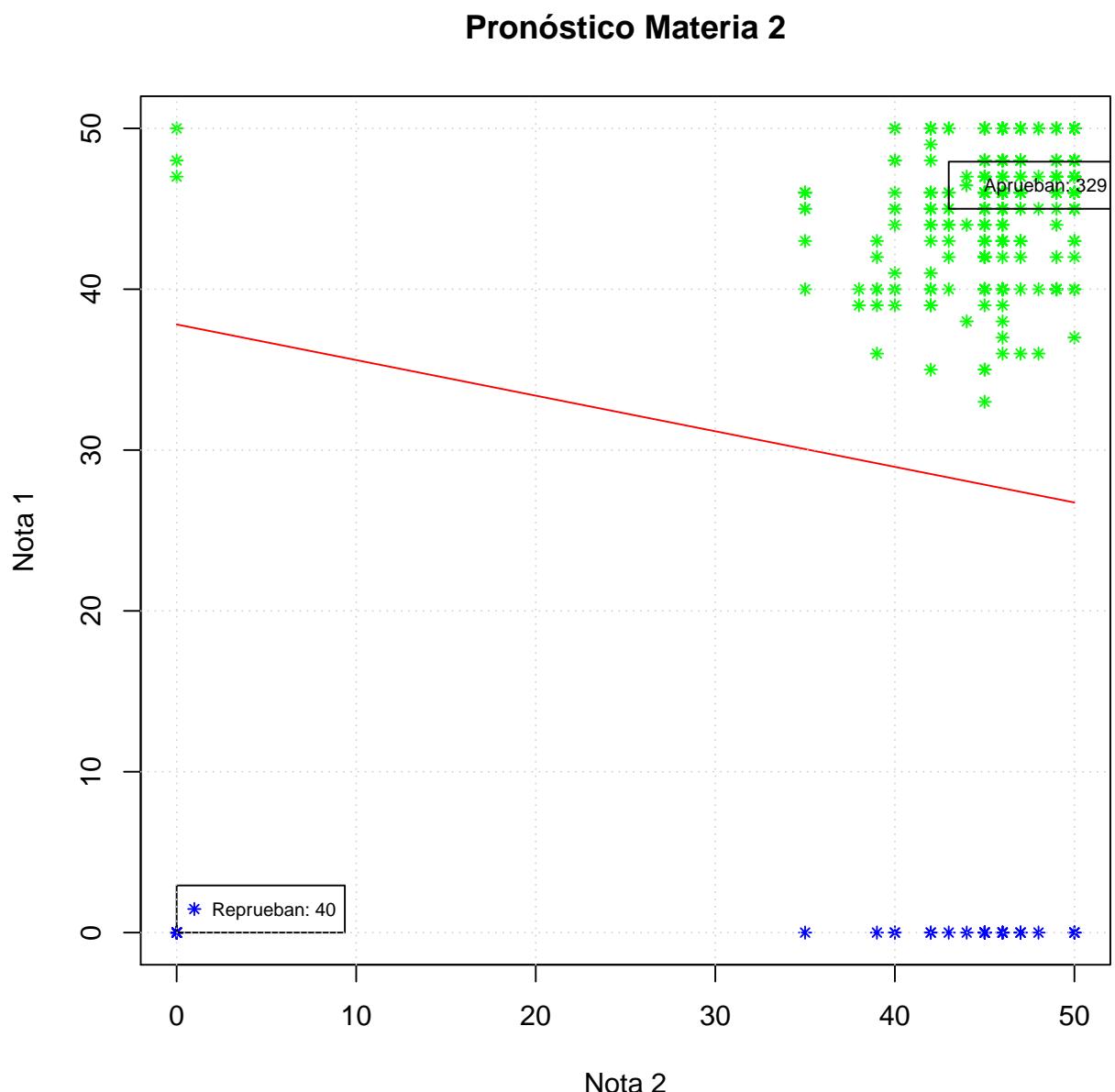


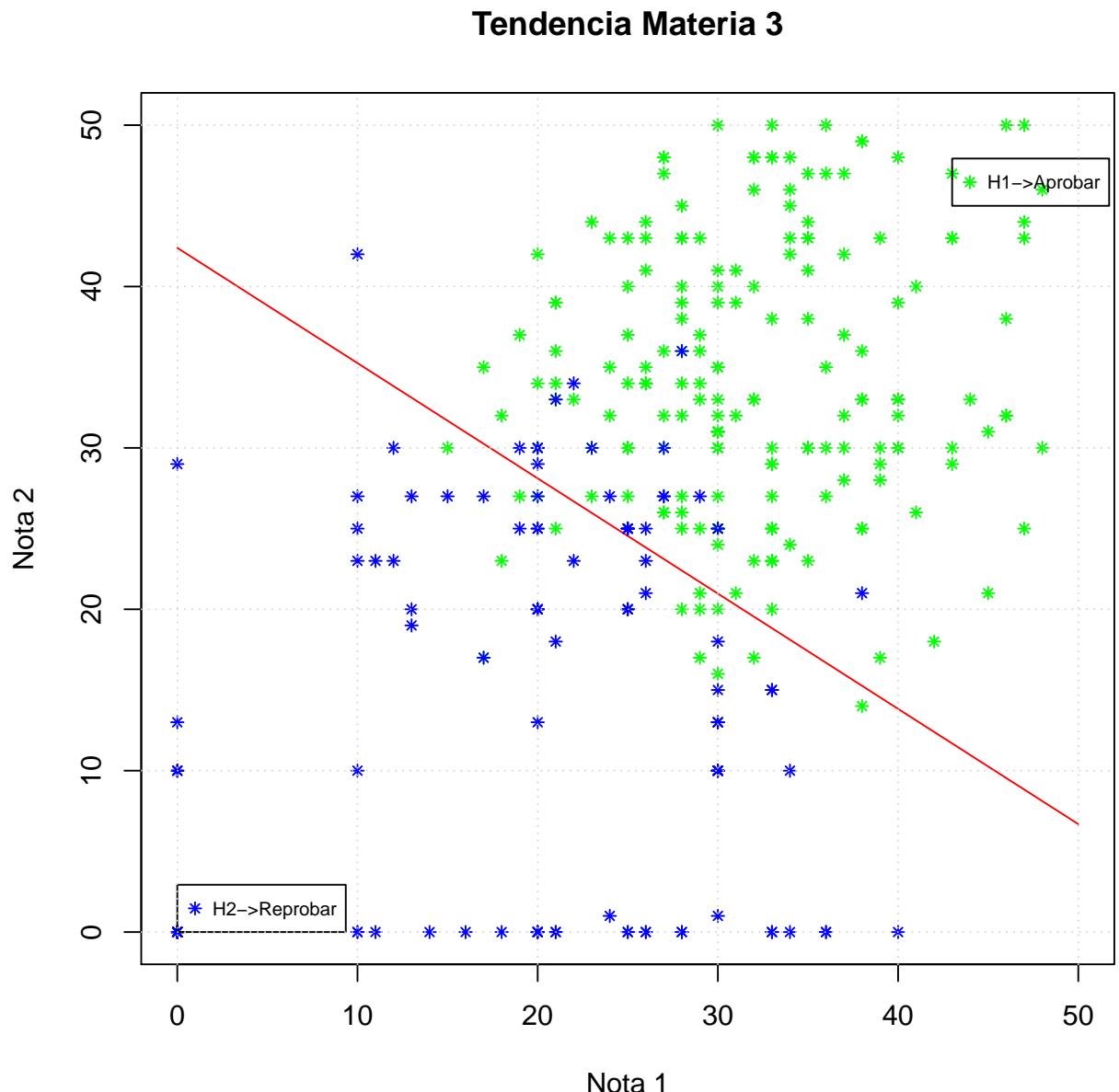
```
## [1] "Materia: 1"
## [1] "Estudiantes aprobados: 280"
## [1] "Estudiantes reprobados: 50"
## [1] "Porcentaje estudiantes aprobados: 84.8484848484848"
## [1] "Porcentaje estudiantes reprobados: 15.1515151515152"
## [1] "//////////"
```



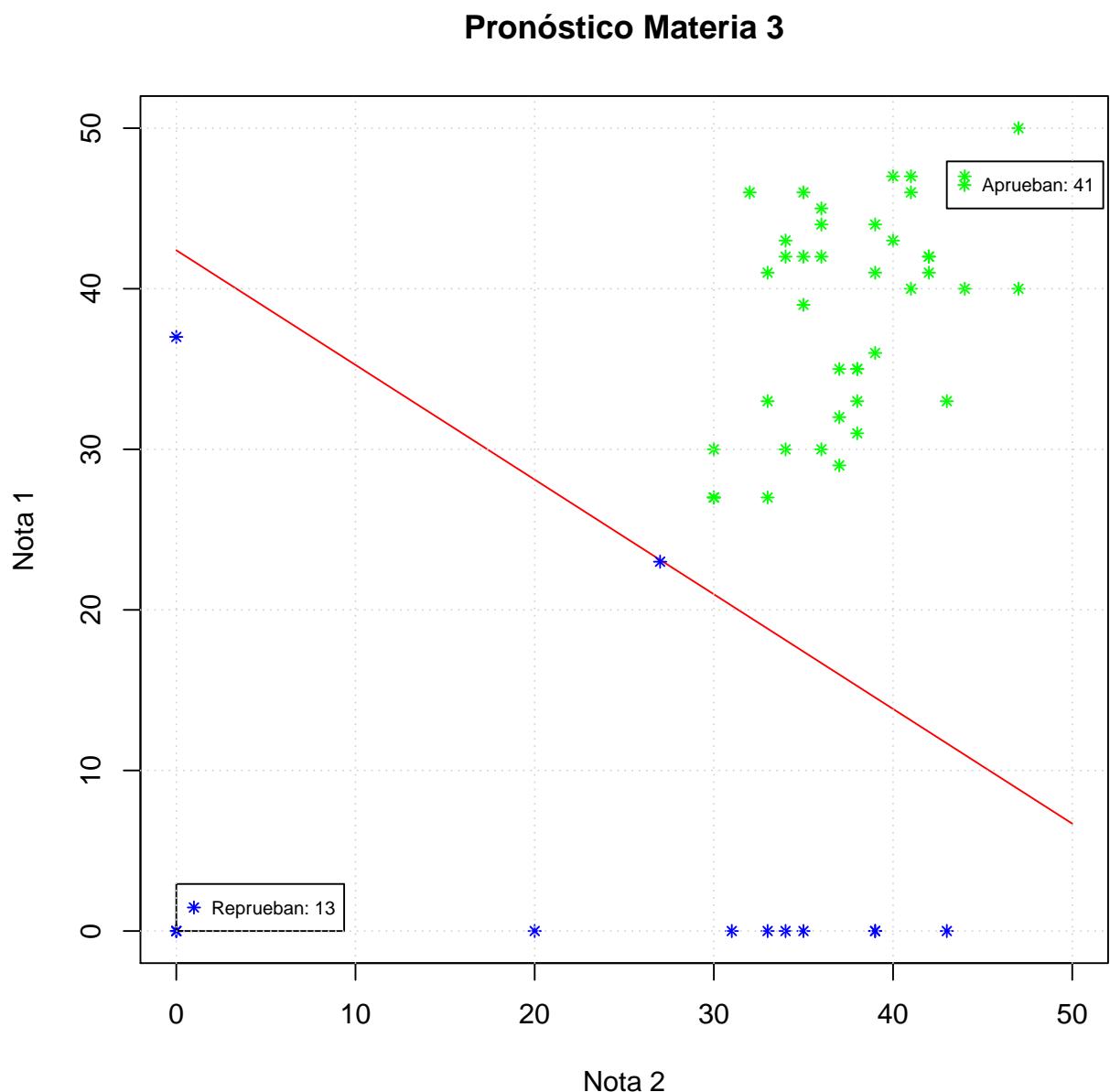


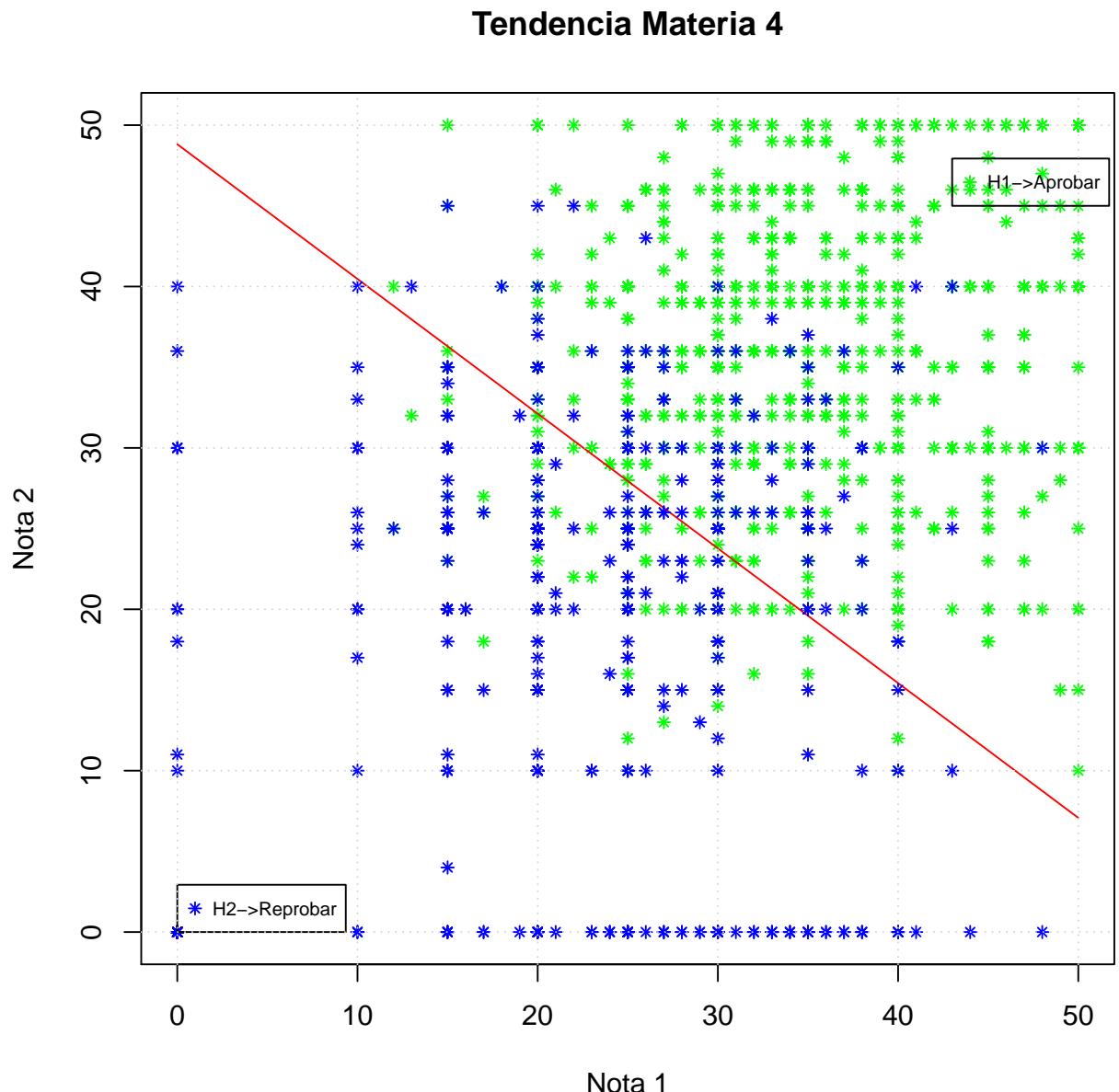
```
## [1] "Materia: 2"
## [1] "Estudiantes aprobados: 329"
## [1] "Estudiantes reprobados: 40"
## [1] "Porcentaje estudiantes aprobados: 89.159891598916"
## [1] "Porcentaje estudiantes reprobados: 10.840108401084"
## [1] "//////////"
```



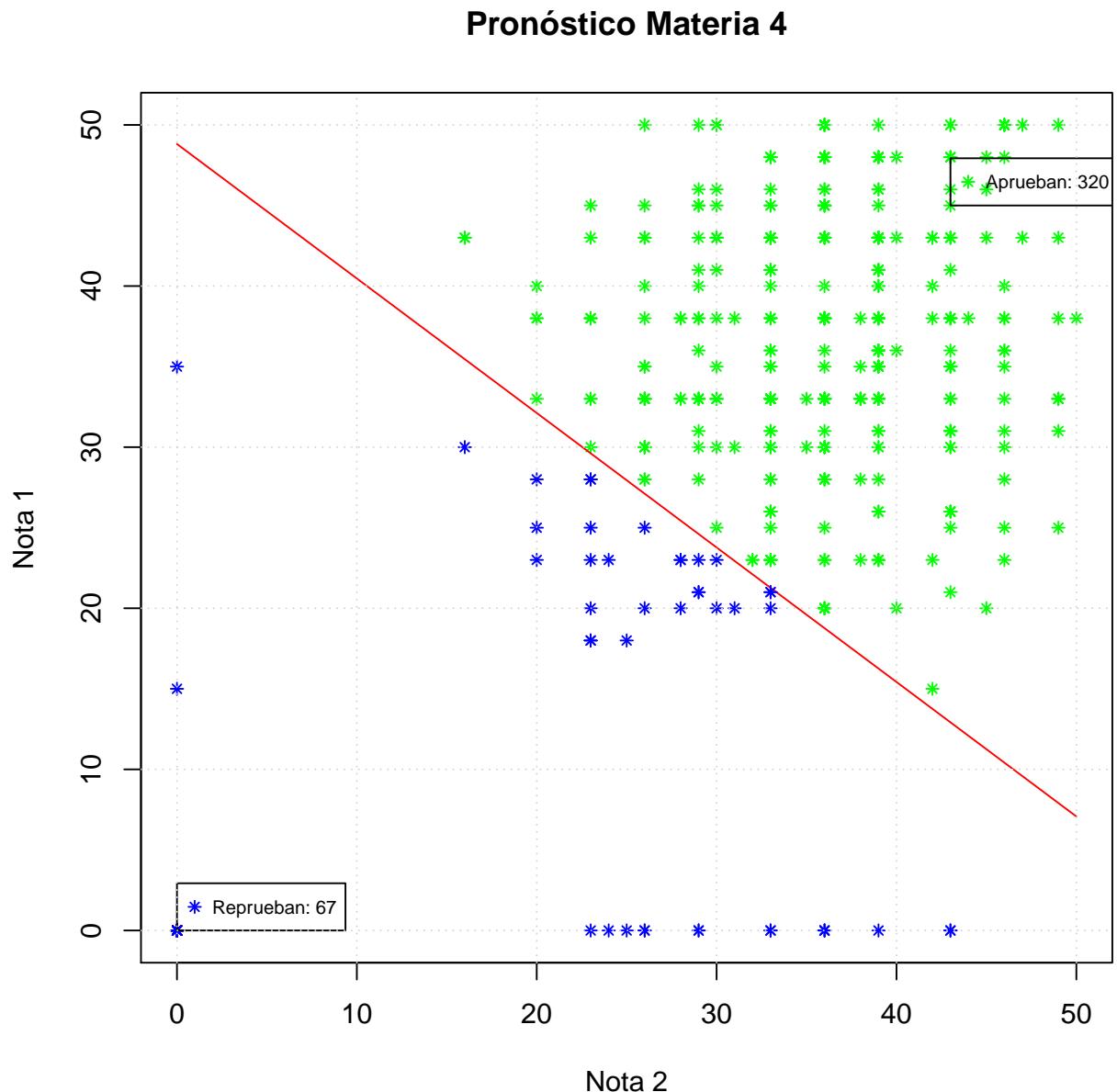


```
## [1] "Materia: 3"
## [1] "Estudiantes aprobados: 41"
## [1] "Estudiantes reprobados: 13"
## [1] "Porcentaje estudiantes aprobados: 75.9259259259259"
## [1] "Porcentaje estudiantes reprobados: 24.0740740740741"
## [1] "///////////////////////////////"
```





```
## [1] "Materia: 4"
## [1] "Estudiantes aprobados: 320"
## [1] "Estudiantes reprobados: 67"
## [1] "Porcentaje estudiantes aprobados: 82.687338501292"
## [1] "Porcentaje estudiantes reprobados: 17.312661498708"
## [1] "//////////"
```



La ejecución en R[®] entregó los mismos resultados que la anterior en MATLAB[®], por lo que se validan los cálculos y el algoritmo está listo para la etapa de implementación.

2.2. Análisis de Requerimientos

Este proyecto permite a los estudiantes de una Institución de Educación Superior consultar la tendencia de aprobación de una materia ingresando las dos notas de parciales.

2.2.1. Diagrama de Contexto

La interacción de los estudiantes, con el sistema construido se presenta en la Figura 2.11, la cual describe los siguientes pasos:

1. El estudiante realiza la autenticación de su identidad en el sistema, para el caso de este ejercicio académico no se implementa esta parte ya que le corresponderá al departamento de sistemas de la Institución de Educación Superior.
2. El Sistema responde con el menú de materias disponibles para consultar.
3. El estudiante selecciona una materia de su interés e ingresa las dos notas de los parciales N1 y N2.
4. El Sistema responde con la tendencia de aprobar esa materia en cuestión, dado el histórico de comportamiento de esa materia.
5. Salir del sistema.

La interacción del administrador, con el sistema construido se presenta en la Figura 2.11, la cual describe los siguientes pasos:

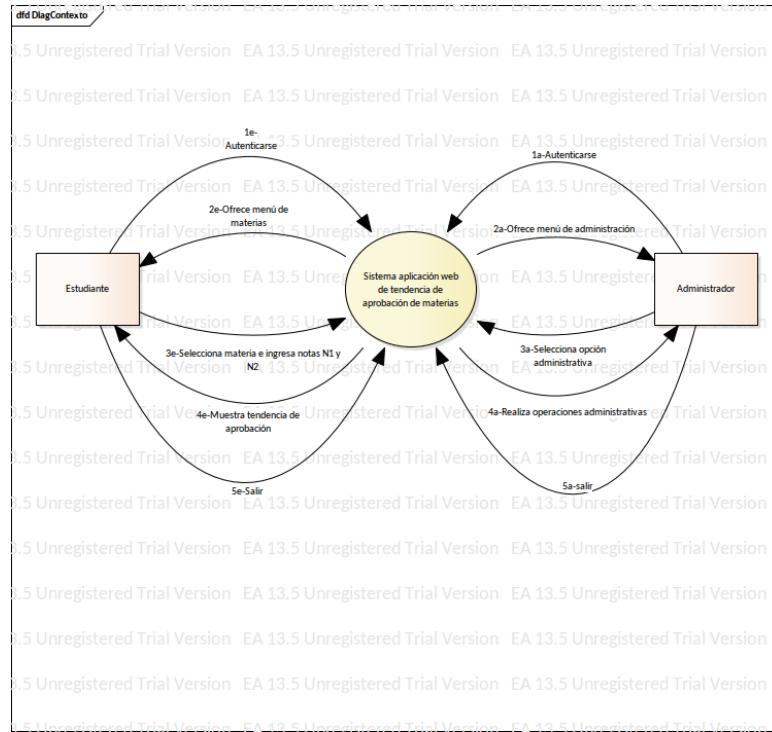
1. El administrador realiza la autenticación de su identidad en el sistema, para el caso de este ejercicio académico no se implementa esta parte ya que le corresponderá al departamento de sistemas de la Institución de Educación Superior.
2. El Sistema responde con el menú de administración, que para este ejercicio académico es el mismo que el ofrecido para estudiante.
3. El administrador selecciona la operación a realizar.
4. El Sistema realiza las operaciones seleccionadas por el administrador.
5. Salir del sistema.

2.2.2. Diagrama de Procesos

El diagrama de procesos se presenta en la Figura 2.12. En diagrama presenta las actividades que el proyecto incluye como lo son:

- La identificación del Id del estudiante.
- La oferta de las materias disponibles para experimentar.
- La captura de la materia seleccionada por el estudiante y de las notas N1 y N2 de los parciales, los cuales son los insumos para realizar los cálculos de tendencia.
- La entrega del resultado de la tendencia de aprobar la materia.

Figura 2.11: Diagrama de Contexto



2.2.3. Requerimientos Funcionales

Los requerimientos funcionales describen los diferentes servicios que el proyecto debe proveer. La Tabla 2.2 presenta los requerimientos funcionales del sistema.

Tabla 2.2: Requerimientos Funcionales

ID	RF 01
Nombre	Autenticación en el Sistema
Descripción	El estudiante debe autenticarse en el sistema ingresando su código y contraseña.
Actores	Estudiante
Observaciones	Ninguna
ID	RF 02
Nombre	Diligenciamiento de formulario con opciones de materias y campos de notas
Descripción	El estudiante debe seleccionar la materia de interés e ingresar los campos de las dos notas de exámenes, luego pulsar el botón de procesar.
Actores	Estudiante
Observaciones	Ninguna
ID	RF 03

Nombre	Calculo de tendencia de aprobación de materias
Descripción	El estudiante una vez haya pulsado el botón de procesamiento debe poder observar en pantalla el resultado del cálculo de la tendencia con la que es posible apruebe su materia de interés.
Actores	Estudiante
Observaciones	Ninguna

2.2.4. Requerimientos No Funcionales

Los requerimientos no funcionales describen diferentes características adicionales que el proyecto debe satisfacer para poder llevar a cabo los servicios que provee. La Tabla 2.3 presenta los requerimientos no funcionales.

Tabla 2.3: Requerimientos No Funcionales

ID	RNF 01
Nombre	Hosting (necesario solo si el cliente lo solicita, queda como opcional para la etapa de prototipado)
Descripción	Características hosting en el cual se alojan el sistema web y base de datos.
Observaciones	Servidor de 4 núcleos de CPU, 16GB RAM disponible, almacenamiento de 120GB SAN, ancho de banda de 2 TB, 2 IPs disponibles
ID	RNF 02
Nombre	Concurrencia (necesario solo si el cliente lo solicita, queda como opcional para la etapa de prototipado)
Descripción	El sistema deberá soportar una concurrencia del 20 por ciento de usuarios (sobre una base de 300 usuarios), donde los tiempos de respuesta se mantienen. Con un valor mayor de concurrencia el sistema sigue prestando servicio pero los tiempos de respuesta empiezan a aumentar. La concurrencia estará dada en los puntos donde se realizan consulta y carga de información en el sistema.
ID	RNF 03
Nombre	Navegadores
Descripción	El sistema debe estar conectado a internet y los navegadores actualizados.
Observaciones	Navegador Mozilla Firefox actualizado, navegador Internet Explorer actualizado, navegador Chrome actualizado y velocidad de internet desde 2MB

ID	RNF 04
Nombre	Disponibilidad (necesario solo si el cliente lo solicita, queda como opcional para la etapa de prototipado)
Descripción	El sistema y aplicación está disponible para el ingreso y consulta de los usuarios. Se requiere que el sistema tenga una disponibilidad general del 97 por ciento por año. Esto quiere decir que el sistema podrá estar caído máximo 262 horas durante el año
Observaciones	El sistema requiere una disponibilidad del 97 por ciento para el periodo diario completo, en el horario de 24x7. Esto quiere decir que el sistema sólo podrá estar caído máximo 0,3 horas (18 minutos) dentro de dicho periodo, sin contabilizar el tiempo de reinicialización de las máquinas. La disponibilidad del sistema dependerá de la disponibilidad del proveedor de acceso a Internet o de los servicios de interconexión prestados por terceros.

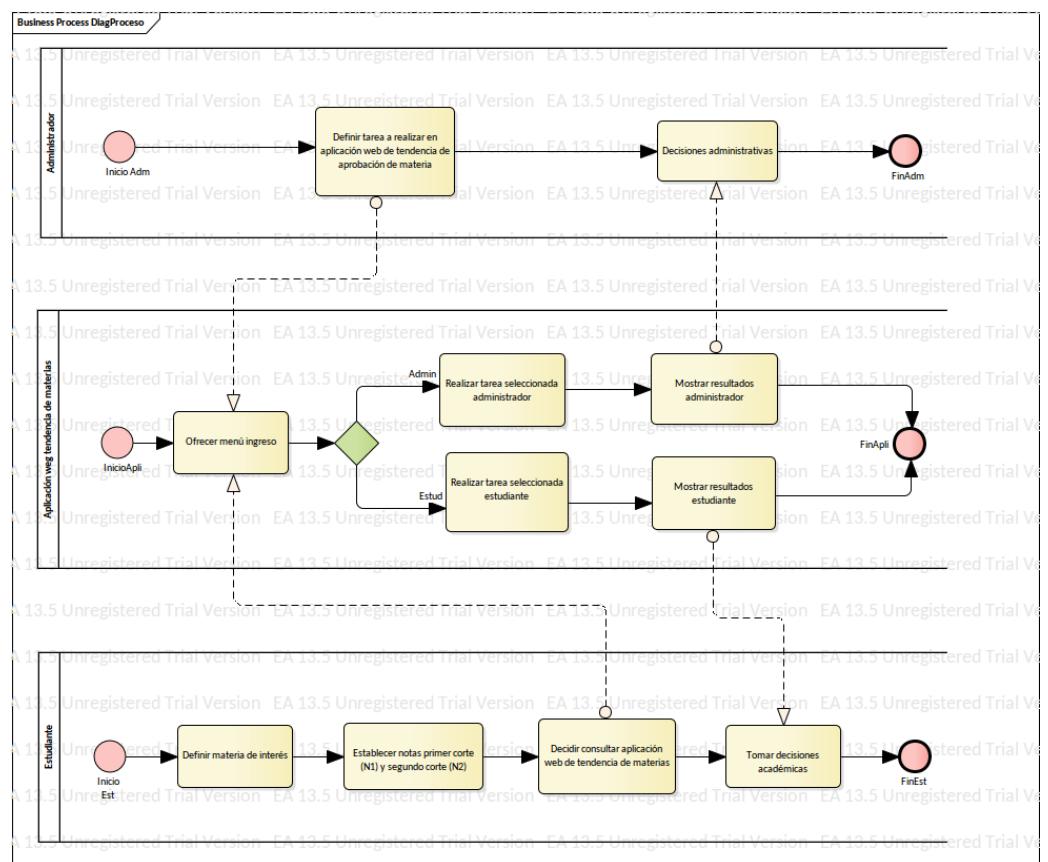
2.3. Definición de Actores

Con base en el análisis de requerimientos realizado, se establecen los actores que se presentan en la Tabla 2.4.

Tabla 2.4: Actores

Nombre	Administrador
Descripción	Actor habilitado para realizar la consultas y modificaciones en el sistema.
Atributos	Nombre, apellido, identificación, contraseña
Nombre	Estudiante
Descripción	Actor habilitado para realizar la consulta de la tendencia de aprobación de alguna materia de su interés.
Atributos	Nombre, apellido, identificación, contraseña

Figura 2.12: Diagrama de Procesos



Capítulo 3

Diseño

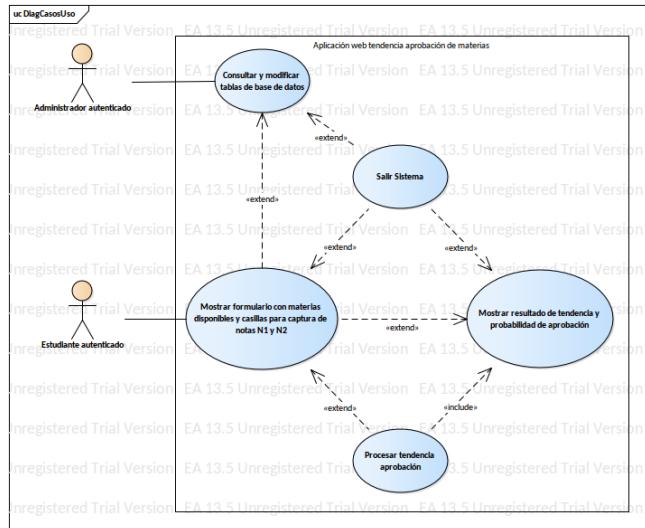
3.1. Casos de Uso

3.1.1. Diagramas de Casos de Uso

A través de el diagrama de casos de uso se representan los diferentes servicios que el Sistema de Pronóstico de Aprobación de Materias ofrece.

La Figura 3.1 presenta el diagrama de casos de uso para el Sistema.

Figura 3.1: Diagrama de Casos de Uso del Sistema



3.1.2. Documentación de Casos de Uso

La documentación de casos de uso describe las características de cada uno de ellos así como el flujo de eventos de interacción entre el actor que participa y el sistema. La Tabla 3.1 describe los casos de uso del sistema.

Tabla 3.1: Documentación de Caso de Uso

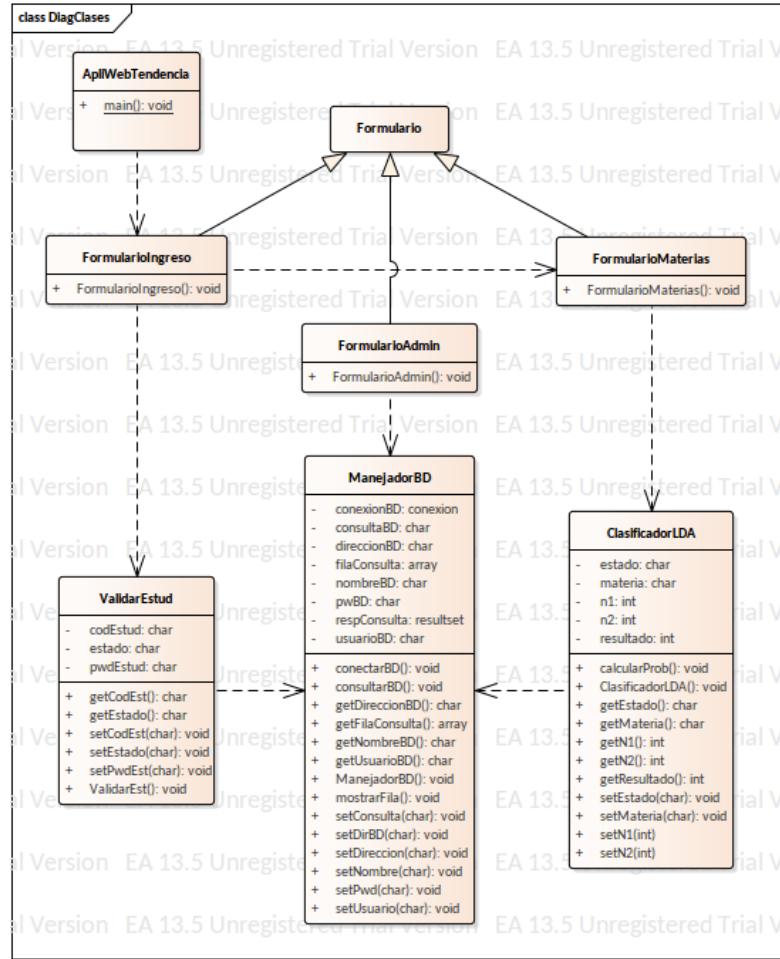
ID	CU 01	
Nombre	Autentificar Estudiante	
Descripción	El estudiante se autentifica en el sistema ingresando los datos solicitados, que para este ejercicio académico no será necesario, ya que su implementación la hará el departamento de Sistemas de la IES.	
Actor	Estudiante	
Precondicion	El estudiante debe existir en el sistema	
Poscondicion	El estudiante se autentifica en el sistema	
Eventos	Actor	Sistema
	1. Ingresa los datos solicitados	
		2. Valida la información
		3. Ingresa la información
		4. Genera un mensaje de confirmación
Excepción	Ninguno	
Autor	Diego Rey	
Fecha	Septiembre de 2017	
ID	CU 02	
Nombre	Ofrecer Listado y opción de ingreso materias	
Descripción	El sistema ofrece el listado de las materias disponibles para consultar y los campos para captura de las notas N1 y N2.	
Actor	Estudiante	
Precondicion	El estudiante debe estar autenticado en el sistema.	
Poscondicion	Los información del formulario se ha desplegado.	
Eventos	Actor	Sistema
		1. Despliega el formulario con la información de materias y campos para captura de notas N1 y N2
Excepción	Ninguno	
Autor	Diego Rey	
Fecha	Septiembre de 2017	
ID	CU 03	
Nombre	Captura de datos del formulario	

Descripción	El sistema captura los datos ingresados por el usuario	
Actor	Estudiante	
Precondicion	El estudiante debe haber seleccionado la materia de interés y digitado las notas N1 y N2.	
Poscondicion	Los datos son capturados.	
Eventos	Actor	Sistema
	1. Selecciona materia de interés	
	2. Digita notas N1 y N2	
	3. Pulsa botón de procesar	4. El sistema captura los datos del formulario
Excepción	Ninguno	
Autor	Diego Rey	
Fecha	Septiembre de 2017	
ID	CU 04	
Nombre	Entrega de resultados de predicción	
Descripción	El sistema procesa los datos ingresados por el estudiante y luego de estimar la tendencia de aprobación de la materia muestra los resultados al usuario en pantalla	
Actor	Estudiante	
Precondicion	El sistema debe haber capturado los datos ingresados por el usuario	
Poscondicion	El resultado de tendencia de aprobación es mostrado en pantalla.	
Eventos	Actor	Sistema
		1. El sistema procesa los datos del formulario y calcula la tendencia de aprobación de la materia.
		2. El sistema muestra en pantalla el resultado del cálculo.
Excepción	Ninguno	
Autor	Diego Rey	
Fecha	Septiembre de 2017	

3.2. Diagramas de Clases

Con base en los casos de uso y su documentación se crea el diagrama de clases, el cual muestra una parte estructural del sistema [5], se presenta en la Figura 3.2.

Figura 3.2: Diagrama de Clases



3.3. Diagrama de Diseño Lógico de Persistencia

El diagrama lógico de persistencia se muestra en la Figura 3.3. Se utiliza el modelo Entidad-Relación, el cual evita que haya que normalizar la base de datos[6].

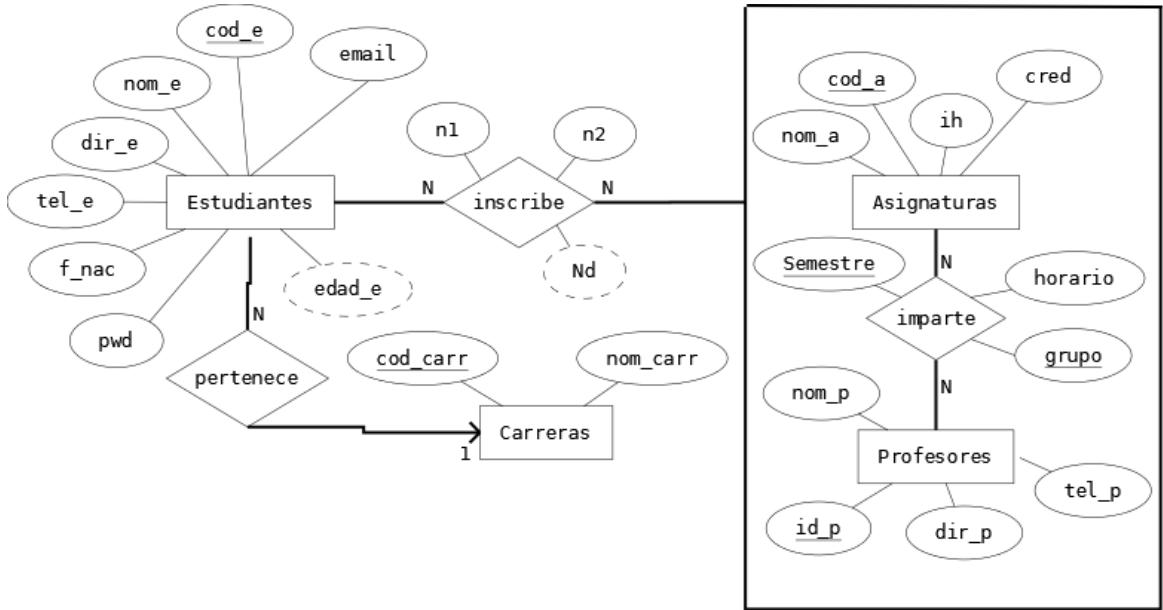
En la parte superior el modelo de diseño y en la parte inferior se muestran las tablas.

3.4. Diagramas de Actividades

Los diagramas de actividades buscan mostrar las actividades que realiza el proceso de un sistema y el flujo de control de una actividad a otra[5].

En la Figura 3.4 se muestra el diagrama de actividades para el algoritmo LDA.

Figura 3.3: Diagrama Lógico de Persistencia



Estudiantes
*cod_e
°nom_e
°dir_e
°email
°tel_e
°f_nac
°cod_carr
°pwd

Asignaturas
*cod_a
°nom_a
°ih
°cred

Profesores
*id_p
°nom_p
°dir_p
°tel_p

Carreras
*cod_carr
°nom_carr

imparte
*id_p
*cod_a
°grupo
°horario

inscribe
*cod_e
*cod_a
*id_p
°grupo
°Semestre
°n1
°n2

3.5. Diagramas de Estados

Los diagramas de estados muestran los estados del sistema y los eventos que causan las transiciones de un estado a otro[5].

En la Figura 3.5 se muestra el diagrama de estados para el sistema.

3.6. Diagramas de Secuencia

Los diagramas de secuencia son usados para modelar las interacciones entre los componentes del sistema, sin embargo agentes externos pueden también incluirse.

En la Figura 3.6 se muestra el diagrama de secuencia para estudiante.

En la Figura 3.7 se muestra el diagrama de secuencia para administrador.

3.7. Diagramas de Componentes

El diagrama de componentes representa cómo el sistema de software se ha dividido en componentes o partes y sus dependencias.

En la Figura 3.8 se muestra el diagrama de componentes del sistema.

3.8. Diagramas de Despliegue

Un diagrama de despliegue muestra cómo los componentes de software son desplegados físicamente en los procesadores. Entonces muestra el *hardware* y el *software* y los componentes intermedios usados para conectar los diferentes componentes en el sistema[5].

En la Figura 3.9 se muestra el diagrama de despliegue del sistema.

3.9. Diagramas de Arquitectura

En el diagrama de arquitectura se observan los componentes y conectores desde un nivel de abstracción mayor, por lo que se utiliza un lenguaje de descripción de arquitectura más especializado[5]. Para esta parte del modelado se utilizó *Enterprise Architect®* y su lenguaje *ArchiMate®*.

En la Figura 3.10 se muestra el diagrama de arquitectura capa de negocio.

En la Figura 3.11 se muestra el diagrama de arquitectura capa de aplicación.

En la Figura 3.12 se muestra el diagrama de arquitectura capa de tecnología.

Figura 3.4: Diagrama de Actividades LDA

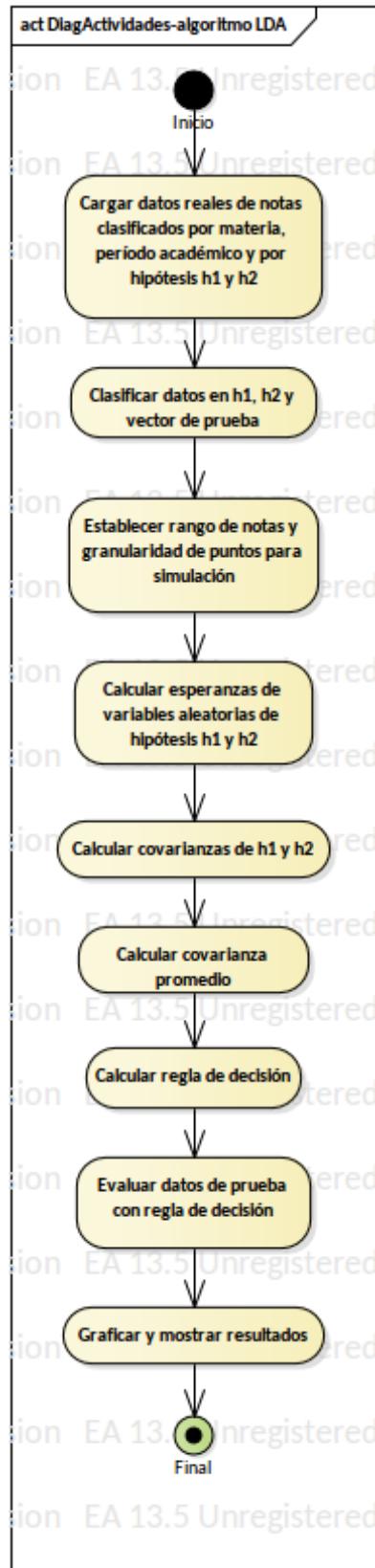


Figura 3.5: Diagrama de Estados

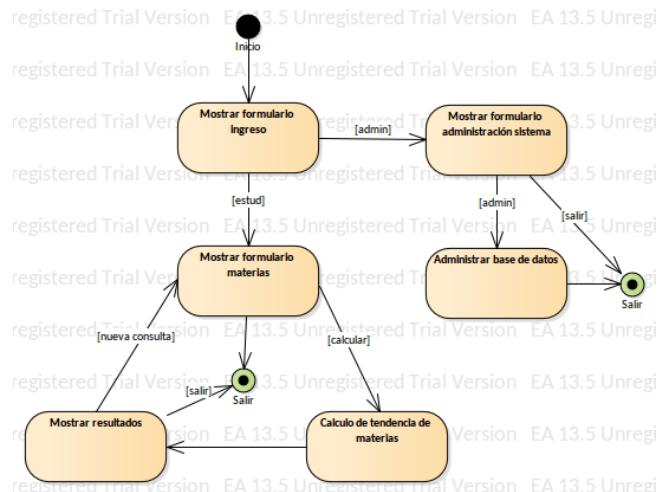


Figura 3.6: Diagrama de Secuencia Estudiante

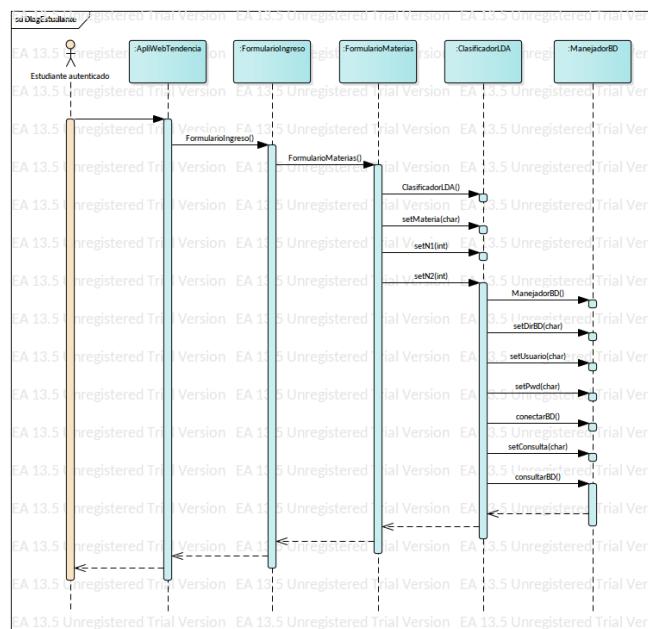


Figura 3.7: Diagrama de Secuencia Administrador

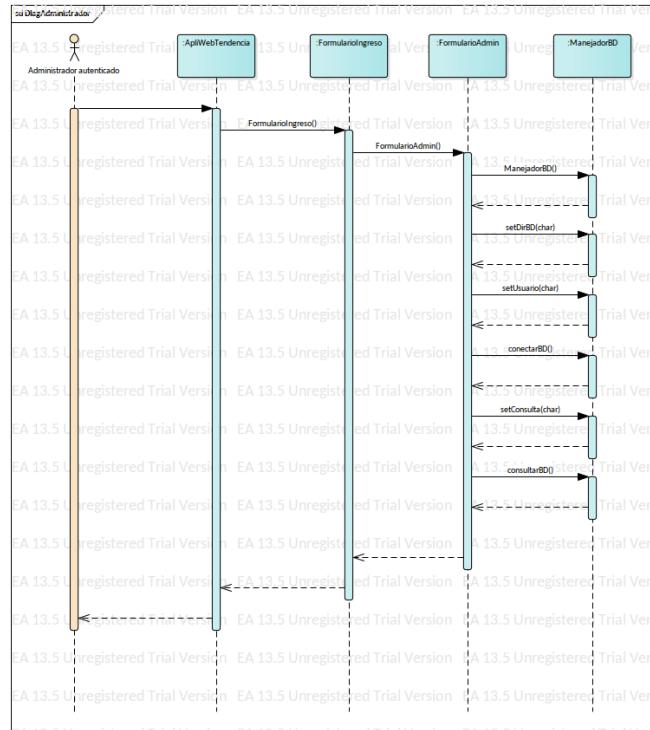


Figura 3.8: Diagrama de Componentes

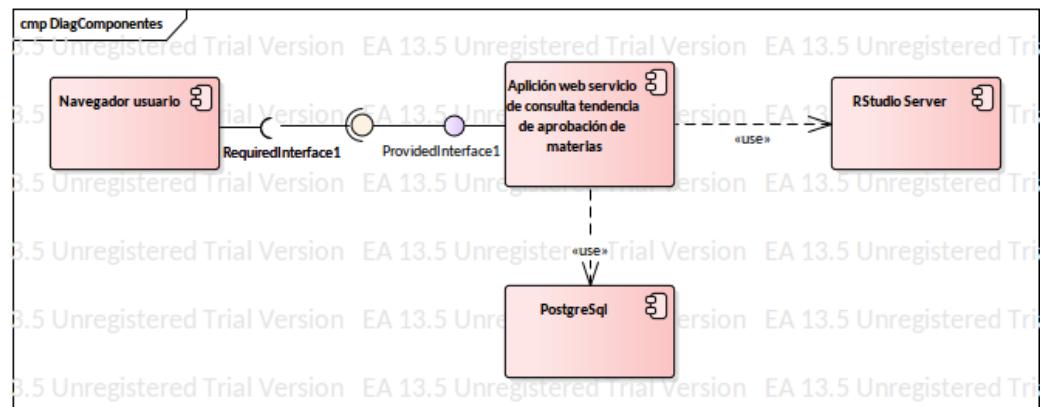


Figura 3.9: Diagrama de Despliegue

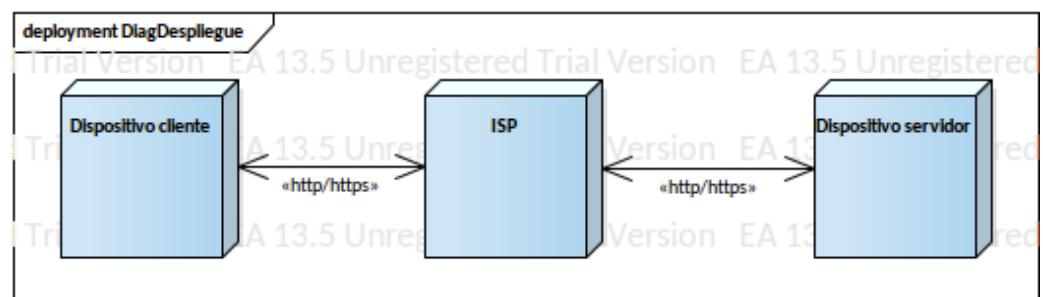


Figura 3.10: Diagrama de Arquitectura Negocio

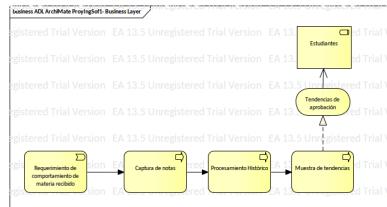


Figura 3.11: Diagrama-de-Arquitectura-Aplicación

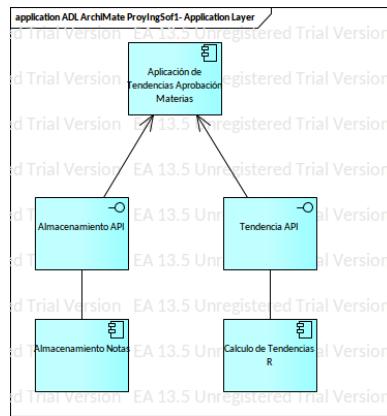
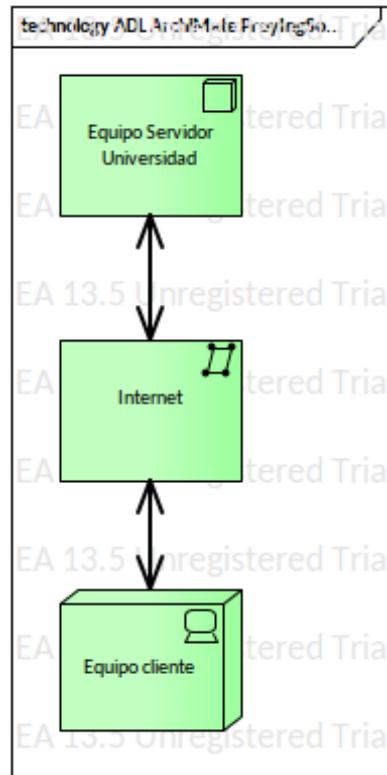


Figura 3.12: Diagrama-de-Arquitectura-tecnología



Capítulo 4

Implementación y Pruebas

En este capítulo se realiza la implementación y pruebas de la aplicación, utilizando tecnologías como Rstudio®, Microsoft Azure®y Shinyapps.io®

4.1. Servicio web rest y base de datos en Microsoft Azure(R)Cloud

Se debe de abrir una cuenta en Microsoft Azure, teniendo claros los objetivos de nuestra aplicación. En la Figura 4.1 se observa las opciones de suscripción en Azure.

Para el desarrollo de esta aplicación se adquirió una cuenta gratuita por un año y con crédito de 200USD para consumo durante un mes a partir de la apertura. En la Figura 4.2 se muestra el panel de opciones de bienvenida de Azure, para implementar nuestra aplicación se debe seleccionar 'App Services'.

Luego seleccionar 'Crear App Services'(Figura 4.3).

Seleccionar la pestaña 'Más' para desplegar el menú completo, como se muestra en la Figura 4.4. Pudiento acceder a todas las opciones.

Seleccionar la opción *Aplicación web y MySQL*, como se observa en la Figura 4.5.

Crear la aplicación, Figura 4.6.

Aparece el menú para configurar los parámetros de interés. Figura 4.7.

Se observa el estado de la creación, en donde aparecen las opciones de configuración más avanzadas.Figura 4.8.

Para configurar la base de datos elegimos el plan de pago, que para este caso es sin costo con límite de hasta 4 conexiones como se observa en la Figura 4.9. Por esta razón de cero costo es que se eligió MySql en lugar de PostgreSql®, como se tenía planeado al principio.

Figura 4.1: Opciones de suscripción en Azure



The screenshot shows the Microsoft Azure website with a navigation bar at the top. The main content area is titled "Referencia rápida de Azure en la educación". Below this is a table comparing various Azure services based on recommendation and requirements.

PROGRAMA	RECOMENDADO PARA	REQUISITOS	Clave
Microsoft Imagine	EDU	Requiere comprobación del estatus de estudiante	Educadores
Visual Studio Dev Essentials	EDU, ESTU	Requiere tarjeta de crédito y número de teléfono para comprobación	Estudiantes
Microsoft Azure for Research	EDU, ESTU, INVESTIGADOR	Requiere una aplicación	Investigador
Azure Notebooks	EDU, ESTU	No se requiere tarjeta de crédito	
Azure Machine Learning	ESTU	No se requiere tarjeta de crédito	
Cuenta de Azure gratis	EDU, ESTU, INVESTIGADOR	Requiere tarjeta de crédito y número de teléfono para comprobación	Profesional de TI
Característica Web Apps de Azure App Service	INVESTIGADOR	No se requiere tarjeta de crédito	
Cursos de Azure gratis	ESTU	No se requiere tarjeta de crédito	
Microsoft IT Pro Career Center	INVESTIGADOR	No se requiere tarjeta de crédito	

Terminamos de configurar nuestra aplicación a nuestro gusto. En la Figura 4.10 se observa que la implementación fue correcta.

Se debe de realizar la configuración de credenciales de implementación, para lo cual nos piden un nombre de usuario y contraseña, la cual debemos guardar en sitio seguro, ya que estos datos nos permiten acceder de forma remota a nuestra aplicación. Después ya podemos realizar el acceso a nuestros recursos. Figura 4.11.

El cargue de la base de datos se realizó mediante la conexión remota por la consola de MySql del servidor Xampp y se realizó el cargue de los datos utilizando comandos SQL.

Para establecer la conexión desde la terminal local hacia la base de datos remota en Azure se debe configurar el comando de conexión MySql ejecutando la siguiente intrucción:

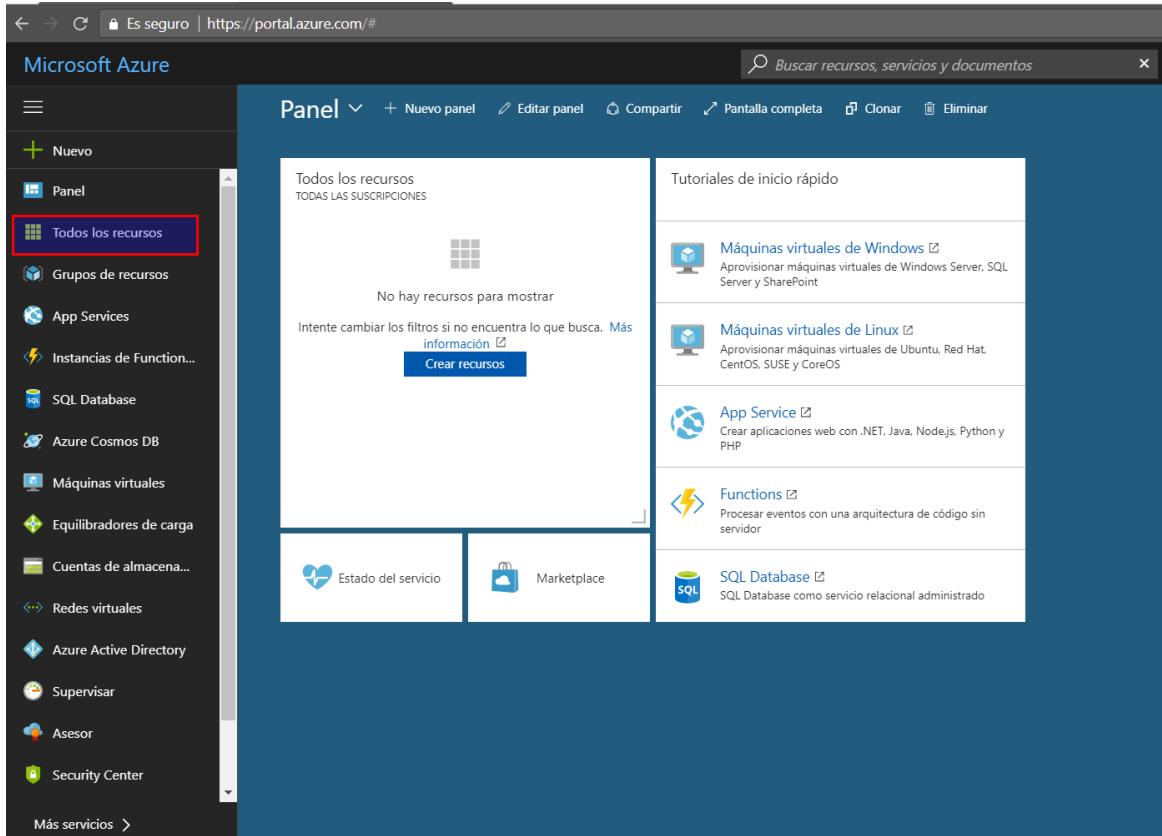
```
mysql -h[] -u[] -p[] -D[]
```

Las llaves de *h* se reemplazan por el nombre del *host* remoto, las llaves de *u* se reemplazan por el nombre de usuario, las llaves de *p* por la contraseña y las llaves de *D* por el nombre de la base de datos con la cual queremos comunicarnos, todos estos datos los obtenemos en las credenciales de la base de datos que desplegamos en pasos anteriores en Azure. Figura 4.12.

Como ejemplo se muestra el siguiente comando con la información en el formato requerido:

```
mysql -hbr-cnbr-azure-south-b.cloudapp.net -uba5337f78 -p2f8d7 -Dnotas_materias
```

Figura 4.2: Panel de opciones Azure



Para cargar la base de datos por la terminal utilizar el archivo de cague automático `notas_materias.sql` que está en la carpeta de códigos de este proyecto. Figura 4.13.

Utilizando un cliente FTP como *FileZilla®*, que puede descargarse sin costo de su página (<https://filezilla-project.org/>) conectarse con nuestra aplicación web en Azure utilizando las credenciales obtenidas en los pasos anteriores, figuras 4.14 y 4.15

Ahora reemplazar en el sitio web de Azure el archivo `index.php` por el que encuentra en la carpeta del proyecto que acompaña este documento, el código PHP®es el siguiente:

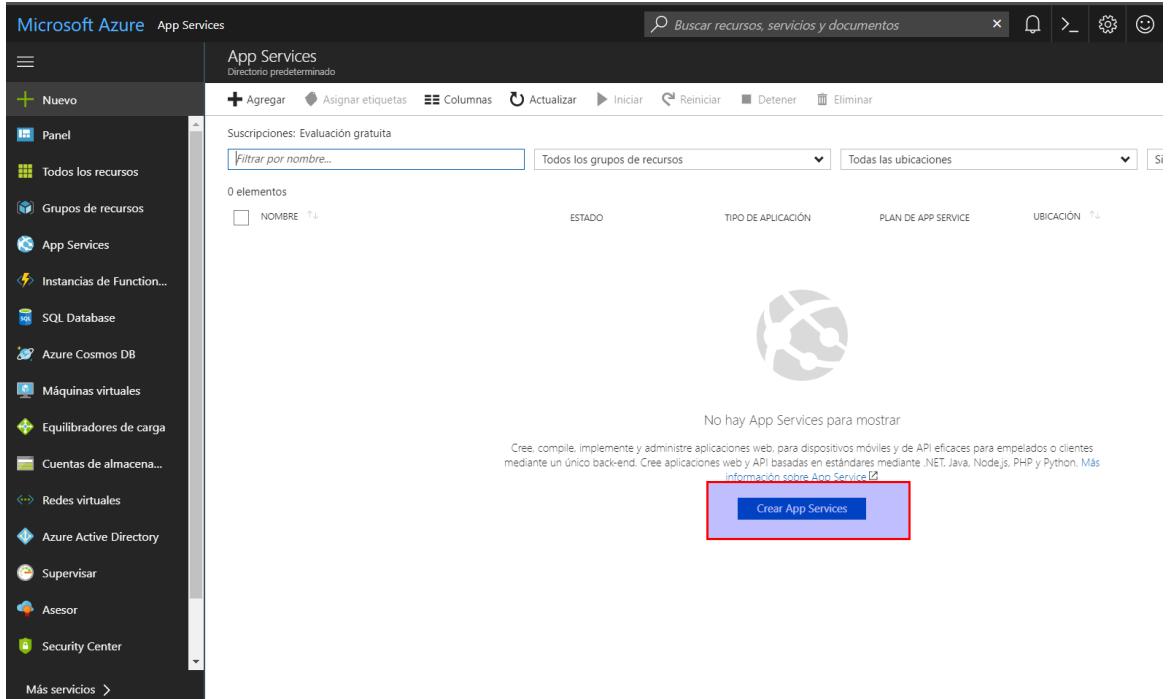
```
//PHP Web Service Azure

header('Content-Type: application/json'); //establece el formato json como formato a devolver

header("Access-Control-Allow-Origin: *");

header("Access-Control-Allow-Methods: POST, GET, OPTIONS");
```

Figura 4.3: Crear App Services



```

$username="xxx"; //nombre de usuario

$password="xxx"; //contraseña

$hostname="xxx"; //nombre de host

$dbhandle=mysql_connect($hostname, $username, $password) //manejador de la base de datos

or die("No es posible conectarse a MySQL"); //mensaje si aparece se genera error

$seleccion=mysql_select_db("notas_materias") //nombre de la base de datos

or die("Base de datos no disponible"); //mensaje si aparece se genera error

function mostrar_registros($detalle){ //funcion para mostrar los registros

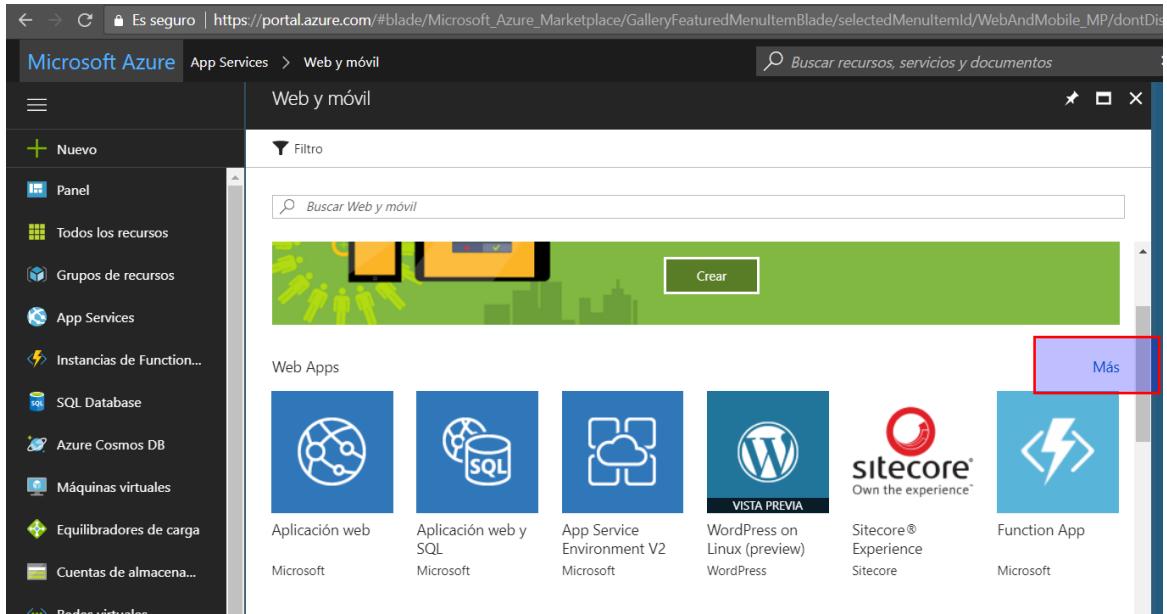
$consulta=mysql_query("SELECT * FROM ".$detalle); //establece consulta

while($fila=mysql_fetch_assoc($consulta)){ //verifica y lee los registros

$registros[]=$fila; //guarda los registros respuesta en $registros[]

```

Figura 4.4: Menú de opciones completo



```

}

return $registros; //retorna los registros de la consulta

}

//Programación de las peticiones al web service

if($_GET['peticion']=='general'){ //verbo GET

$resultados=mostrar_registros($_GET['detalle']); //si peticion==general, lle detalle

}else{

header('HTTP/1.1 405 Method Not Allowed'); //muestra mensaje si se genera error

exit;

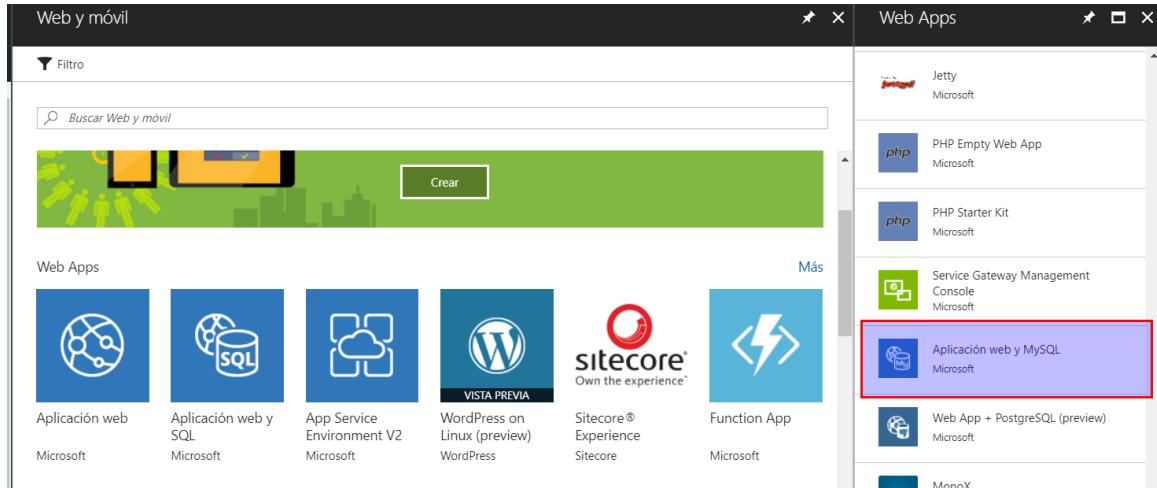
}

echo json_encode($resultados); //codificación del resultado en formato json

?>

```

Figura 4.5: Selección de aplicación web y MySQL



La prueba del servicio se hizo utilizando el cliente *POSTMAN*®, el cual se puede descargar sin costo de su página(<https://www.getpostman.com/>) como se observa en la Figura 4.16.

En la url colocar la url de nuestro servicio web en Azure, y en el verbo HTTP seleccionar GET, luego cargar las variables *peticion=general* y *detalle=materias*. El url con la petición quedaría como:

```
http://tendenciamaterias.azurewebsites.net/?peticion=general&detalle=materias.
```

Si quiero consultar cada periodo académico por materia colocamos en la variable *detalle* el discriminante adecuado, por ejemplo si quiero obtener los datos de la *materia 1*, en el periodo *20151* y de la *hipótesis 2* (reprobados) entonces la url se arma como:

```
http://tendenciamaterias.azurewebsites.net/?peticion=general&detalle=X_1_20151_h2
```

Si en cambio quiero obtener los datos en el mismo periodo *20151* y de la *hipótesis 1* (aprobados) entonces la url se arma como:

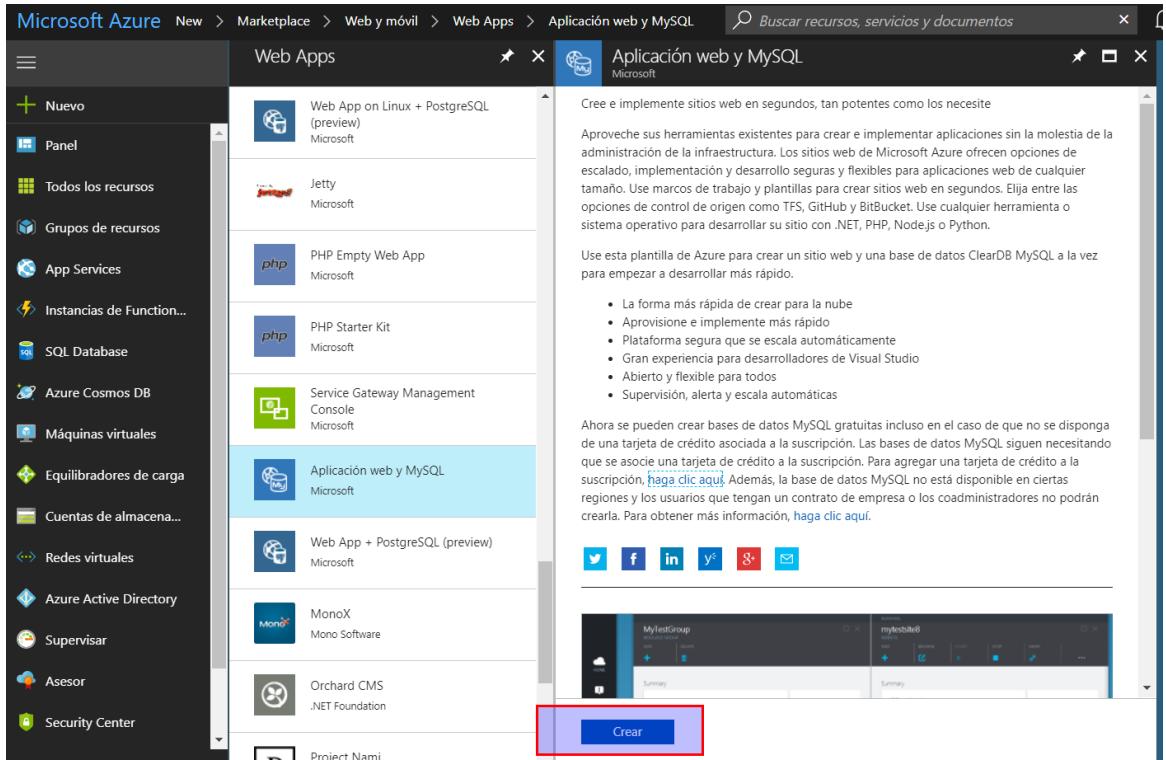
```
http://tendenciamaterias.azurewebsites.net/?peticion=general&detalle=X_1_20151_h1
```

Se puede observar los datos obtenidos después de la petición GET al web service en la figura 4.16.

4.2. Programación de aplicación en R para analítica de datos de materias

La aplicación cliente se programó en Rstudio®utilizando la librería *shiny* para programar la interface de usuario y la operación del servidor *shinyapps.io*. El despliegue se explicará en la siguiente sección. Las pruebas realizadas en servidor local de Rstudio se muestran en la Figura 4.18. Para correr este archivo localmente se debe instalar la librería *Shiny* en Rstudio®, luego correr (*Run App*) el

Figura 4.6: Crear la aplicación



archivo *app.R* que se encuentra en la carpeta de este proyecto.

Se realizaron pruebas con el servidor local y el programa en Rstudio para verificar los resultados de clasificación de estudiantes. Tener en cuenta que en la misma carpeta del proyecto deben estar los archivos de las notas, que tienen terminación *.csv* en la carpeta *DataMateriasCsv*. Se debe actualizar la ruta en el archivo de acuerdo a la ubicación del directorio:

```
ruta="~/Escritorio/DataMateriasCsv/" #ruta de las notas de las materias en equipo local
```

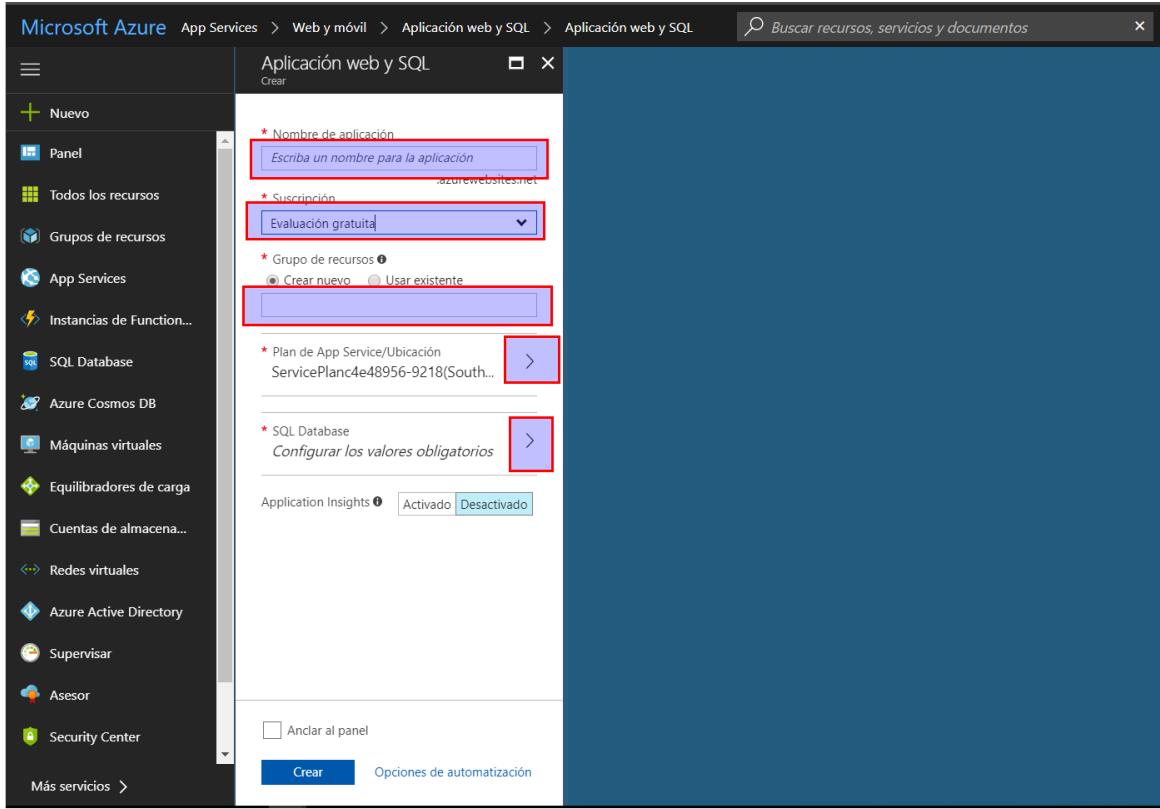
O si se está corriendo en el servidor:

```
ruta="DataMateriasCsv/" #Ruta para ejecución con Knitr, carpeta en mismo directorio
```

4.3. Despliegue de aplicación en Shinyapps.io Cloud

Para desplegar en el servidor *Shinyapps.io* se debe instalar el paquete *rsconnect* en Rstudio desde la consola, y luego abrir una cuenta en <https://shiny.rstudio.com/> con el correo electrónico, nombre de usuario y contraseña, seguir las instrucciones que dan en la página como se muestra en la figura 4.19.

Figura 4.7: Parámetros de configuración



Se debe requerir un token desde el servidor web de *Shinyapps.io*, este permite el enlace entre el servidor y nuestra aplicación en Rstudio, seguir las instrucciones de la página <https://shiny.rstudio.com/>. Se debe copiar este token con el código secreto revelado a la consola de Rstudio y ejecutar el comando. Figura 4.20

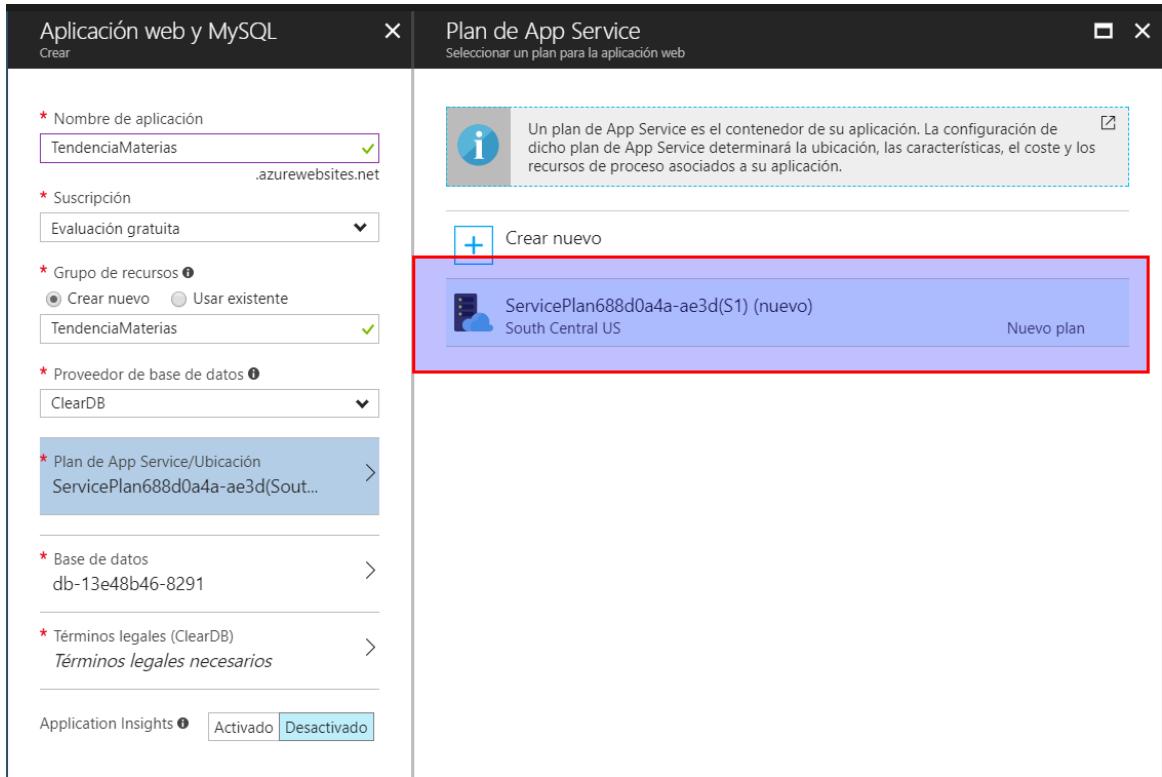
Una vez autenticada la conexión entre Rstudio y el servidor se procede a realizar el cargue de la aplicación, se debe primero ejecutarla desde Rstudio y luego presionar el botón de despliegue *Publish* en la parte superior derecha como se observa en la Figura 4.21. Como buena práctica Shinyapps recomienda que el archivo de la aplicación se llame *app.R* y que se guarde en una carpeta con todos los archivos necesarios para su funcionamiento, por ejemplo imágenes, archivos, etc.

Para esta aplicación se escogió el servicio de *Shinyapps.io* gratuito, el cual tiene restricción de uso de 24 horas mensuales, por lo que se recomienda que antes de salir del servidor se debe apagar la aplicación para que no consuma tiempo de forma innecesaria. Figura 4.22.

Las pruebas realizadas muestran el buen funcionamiento de la analítica de los datos consumidos desde el servicio web en Azure®. En la Figura 4.23 se muestra la pantalla de inicio de cargue de la aplicación corriendo en el entorno *cloud* de *Shinyapps.io*.

En la Figura 4.24 se muestra el cálculo del histórico de tendencia de aprobación de la materia 1, es decir los datos con los que se entrenó el modelo de discriminante lineal y se obtiene la línea

Figura 4.8: Opciones avanzadas



característica para la materia seleccionada, donde los puntos verdes representan los datos históricos que forman la hipótesis de aprobados (H1), los puntos en azul representan los datos históricos que forman la hipótesis de reprobados (H2). En la parte inferior de la gráfica se muestran las estadísticas exploratorias de cada variable aleatoria de hipótesis.

En la Figura 4.25 se muestra el cálculo de la tendencia de aprobación de la materia 4 con los datos de prueba de notas de estudiantes a clasificar, donde los puntos verdes representan las notas de estudiantes que forman la hipótesis de aprobados (H1), los puntos en azul representan las notas de estudiantes que forman la hipótesis de reprobados (H2).

En la esquina inferior izquierda de la gráfica aparece un recuadro con la cantidad de estudiantes reprobados, y en la esquina superior derecha aparecen en el recuadro la cantidad de estudiantes aprobados según el modelo.

En la parte inferior de la gráfica se muestran las estadísticas exploratorias de cada variable aleatoria de hipótesis ya clasificadas para los datos de prueba.

En la Figura 4.26 se puede observar la clasificación por pronóstico individual para la materia 1, donde al calcular la tendencia con las notas *Nota 1* y *Nota 2* ingresadas por el usuario se obtiene un resultado que ubica las calificaciones en la zona de **reprobados**, mostrándose el punto en color **azul** por debajo de la línea de clasificación del modelo de tendencia para esa materia.

Figura 4.9: plan de pago base de datos

The figure consists of three windows from a cloud service interface:

- Aplicación web y MySQL**: Shows basic configuration like application name, subscription, and database selection.
- New MySQL Database**: Shows the creation of a new MySQL database named "notas_materias". It includes fields for Database Name, Database Type (Shared), Location (Sur de Brasil), and Pricing Tier (Mercury).
- Choose Your Pricing Tier**: A detailed table comparing various pricing tiers:

Tier	Total connections	GB	SLA	USD/MONTH (ESTIMATED)
Mercury	4	.02	99.99%	0,00
Titan	10	.25	99.99%	3,50
Venus	15	1	99.99%	9,99
Saturn	30	5	99.99%	USD/MONTH (ESTIMATED)
Jupiter	40	10	99.99%	USD/MONTH (ESTIMATED)

En la Figura 4.27 se puede observar la clasificación por pronóstico individual para la materia 1, donde al calcular la tendencia con las notas *Nota 1* y *Nota 2* ingresadas por el usuario se obtiene un resultado que ubica las calificaciones en la zona de **aprobados**, mostrándose el punto en color **verde** por encima de la línea de clasificación del modelo de tendencia para esa materia.

Se observan excelentes resultados de la herramienta y la disponibilidad con la que ofrece los resultados al poderse consultar de forma web y con los beneficios que ofrece el servidor de Rstudio para el análisis de datos, construcción de modelos y pronóstico de resultados.

Figura 4.10: Configuración final

The screenshot shows the Azure portal interface. At the top, there are buttons for 'Compartir', 'Pantalla completa', 'Clonar', and 'Eliminar'. Below this, the 'Tutoriales de inicio rápido' section is displayed, featuring three items: 'Máquinas virtuales de Windows', 'Máquinas virtuales de Linux', and 'App Service'. To the right, a 'Notificaciones' sidebar is open, showing a single notification titled 'Implementación correcta' with the message: 'La implementación "Microsoft.WebSiteMySQLDatabase4917266a-9678" se realizó correctamente en el grupo de recursos "TendenciaMaterias".' Below the message are two buttons: 'Ir al recurso' and 'Anclar al panel', with 'Ir al recurso' highlighted by a red box.

Figura 4.11: Credenciales de implementación

The screenshot shows the Azure portal interface. On the left, a list of resources is shown with 'TendenciaMaterias' selected and highlighted by a red box. On the right, the 'Credenciales de implementación' blade is open, displaying a warning about using existing credentials for Local Git and FTP. It includes fields for 'FTP/Nombre de usuario de implementación' (set to 'azure_drey'), 'Contraseña' (a masked password), and 'Confirmar contraseña' (also a masked password). The entire 'Credenciales de implementación' blade is highlighted by a red box.

Figura 4.12: Datos de conexión base de datos Azure

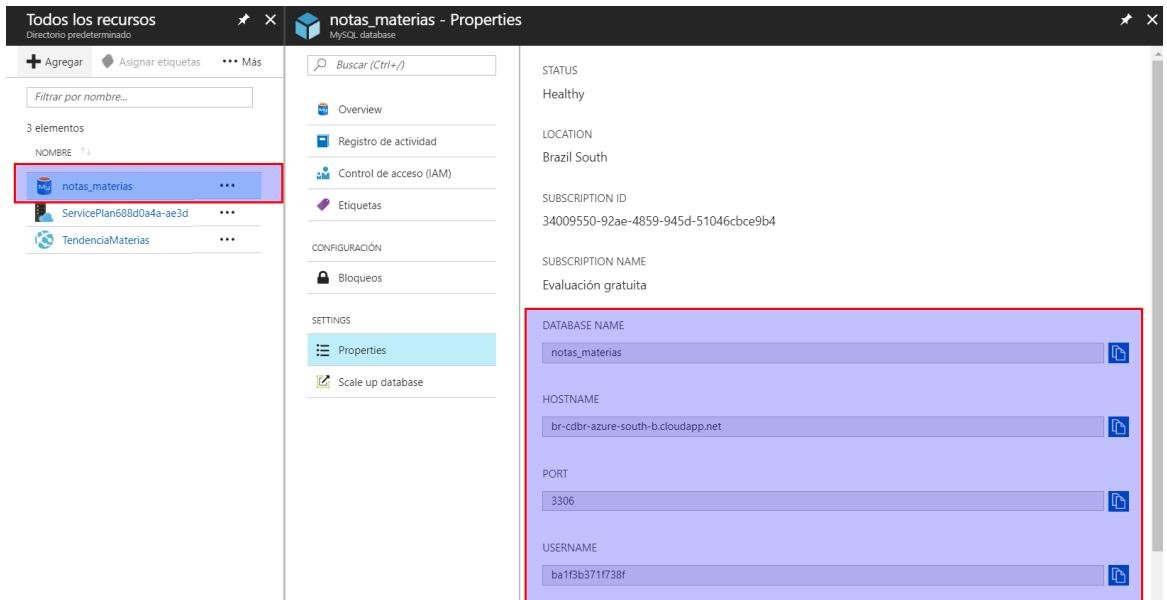


Figura 4.13: Carque de base de datos por consola

```
mysql -hbr-cdbr-azure-south-b.cloudapp.net -uba1f3b371f738f -pc2ef68d7 -Dnotas_materias
Setting environment for using XAMPP for Windows.
DIEGO RAY@ASUS c:\xampp
# mysql -hbr-cdbr-azure-south-b.cloudapp.net -uba1f3b371f738f -pc2ef68d7 -Dnotas_materias
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MySQL connection id is 44030414
Server version: 5.5.56-log MySQL Community Server (GPL)

Copyright (c) 2000, 2017, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [notas_materias]> CREATE TABLE `materias` ( `nombre_mat` varchar(20) DEF
ULT NULL) ENGINE=InnoDB DEFAULT CHARSET=utf8;
ERROR 1050 (42S01): Table 'materias' already exists
MySQL [notas_materias]> INSERT INTO `materias`(`nombre_mat`) VALUES('materia 1
'), ('materia 2'), ('materia 3'), ('materia 4');
Query OK, 4 rows affected (0.53 sec)
Records: 4  Duplicates: 0  Warnings: 0

MySQL [notas_materias]>
```

Figura 4.14: FileZilla®, cargue de credenciales

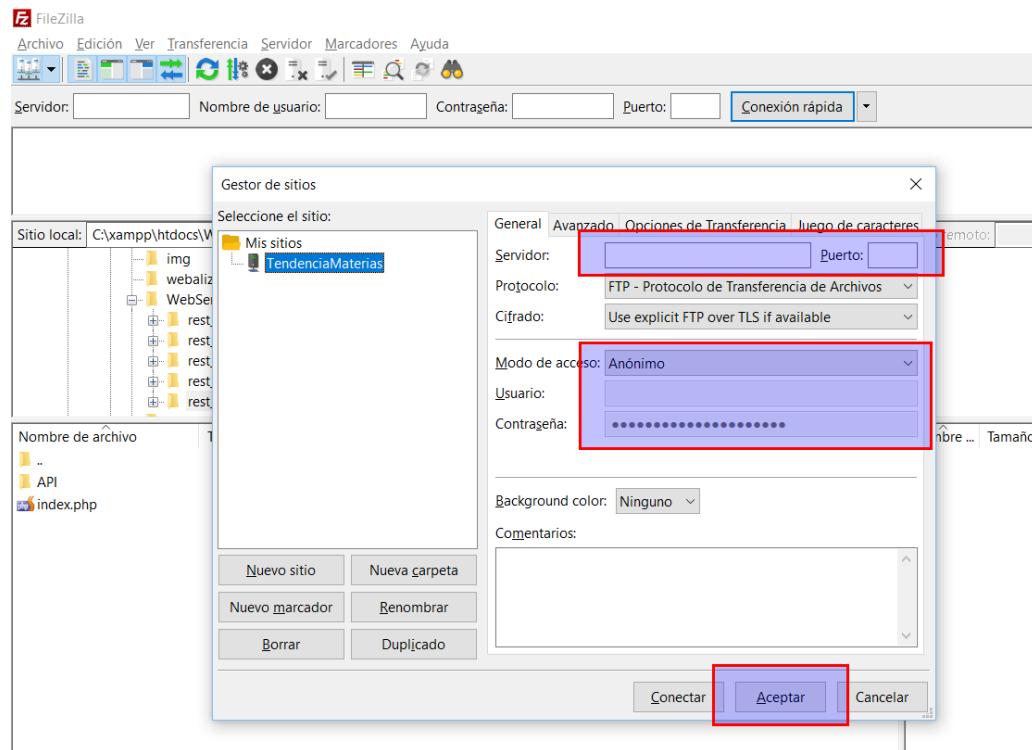


Figura 4.15: FileZilla®, entorno de archivos

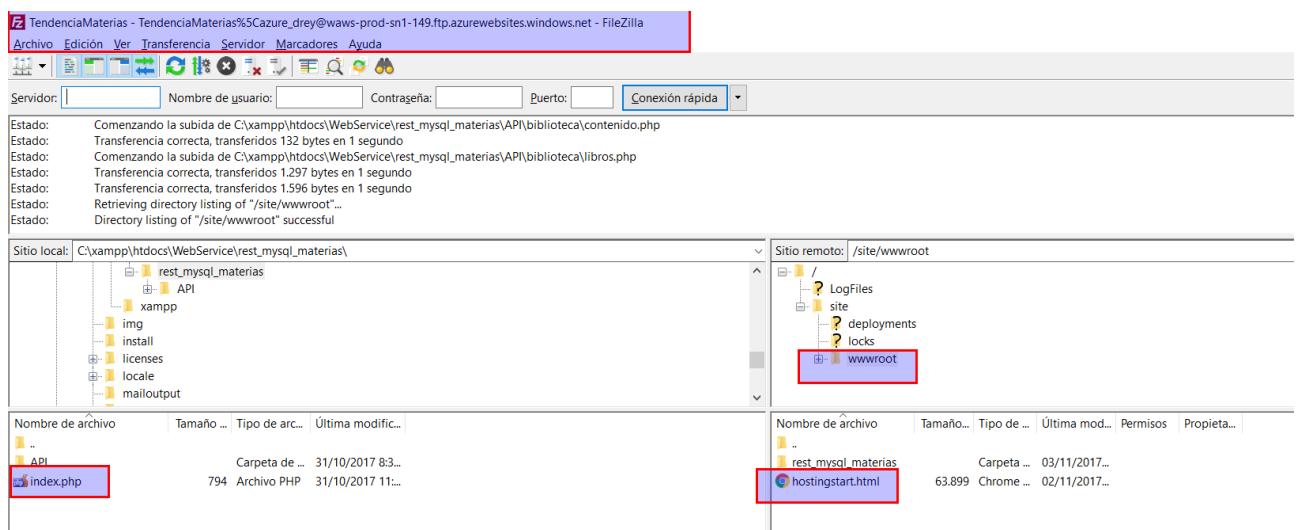


Figura 4.16: Prueba del web service, consumo por cliente POSTMAN®

The screenshot shows the POSTMAN interface with a GET request to the specified URL. The 'peticion' and 'detalle' parameters are selected and highlighted. The response body is a JSON array of objects, each with 'NOTA1' and 'NOTA2' properties.

```

[{"NOTA1": "35", "NOTA2": "35"}, {"NOTA1": "10", "NOTA2": "20"}, {"NOTA1": "15", "NOTA2": "10"}, {"NOTA1": "0", "NOTA2": "10"}, {"NOTA1": "15", "NOTA2": "15"}, {"NOTA1": "20", "NOTA2": "20"}, {"NOTA1": "15", "NOTA2": "25"}, {"NOTA1": "20", "NOTA2": "30"}, {"NOTA1": "15", "NOTA2": "35"}, {"NOTA1": "20", "NOTA2": "40"}, {"NOTA1": "15", "NOTA2": "45"}, {"NOTA1": "20", "NOTA2": "50"}, {"NOTA1": "15", "NOTA2": "55"}, {"NOTA1": "20", "NOTA2": "60"}, {"NOTA1": "15", "NOTA2": "65"}, {"NOTA1": "20", "NOTA2": "70"}, {"NOTA1": "15", "NOTA2": "75"}, {"NOTA1": "20", "NOTA2": "80"}, {"NOTA1": "15", "NOTA2": "85"}, {"NOTA1": "20", "NOTA2": "90"}, {"NOTA1": "15", "NOTA2": "95"}, {"NOTA1": "20", "NOTA2": "100"}, {"NOTA1": "0", "NOTA2": "10"}, {"NOTA1": "0", "NOTA2": "20"}, {"NOTA1": "0", "NOTA2": "30"}, {"NOTA1": "0", "NOTA2": "40"}, {"NOTA1": "0", "NOTA2": "50"}, {"NOTA1": "0", "NOTA2": "60"}, {"NOTA1": "0", "NOTA2": "70"}, {"NOTA1": "0", "NOTA2": "80"}, {"NOTA1": "0", "NOTA2": "90"}, {"NOTA1": "0", "NOTA2": "100"}, {"NOTA1": "25", "NOTA2": "25"}, {"NOTA1": "30", "NOTA2": "30"}, {"NOTA1": "35", "NOTA2": "35"}, {"NOTA1": "40", "NOTA2": "40"}, {"NOTA1": "45", "NOTA2": "45"}, {"NOTA1": "50", "NOTA2": "50"}, {"NOTA1": "55", "NOTA2": "55"}, {"NOTA1": "60", "NOTA2": "60"}, {"NOTA1": "65", "NOTA2": "65"}, {"NOTA1": "70", "NOTA2": "70"}, {"NOTA1": "75", "NOTA2": "75"}, {"NOTA1": "80", "NOTA2": "80"}, {"NOTA1": "85", "NOTA2": "85"}, {"NOTA1": "90", "NOTA2": "90"}, {"NOTA1": "95", "NOTA2": "95"}, {"NOTA1": "100", "NOTA2": "100"}]

```

Figura 4.17: Datos consumidos desde Web Service en Azure®

The screenshot shows a browser window displaying the raw JSON response from the Azure web service. The response is a large JSON array of objects, identical to the one shown in Figure 4.16.

Figura 4.18: Prueba de la aplicación localmente

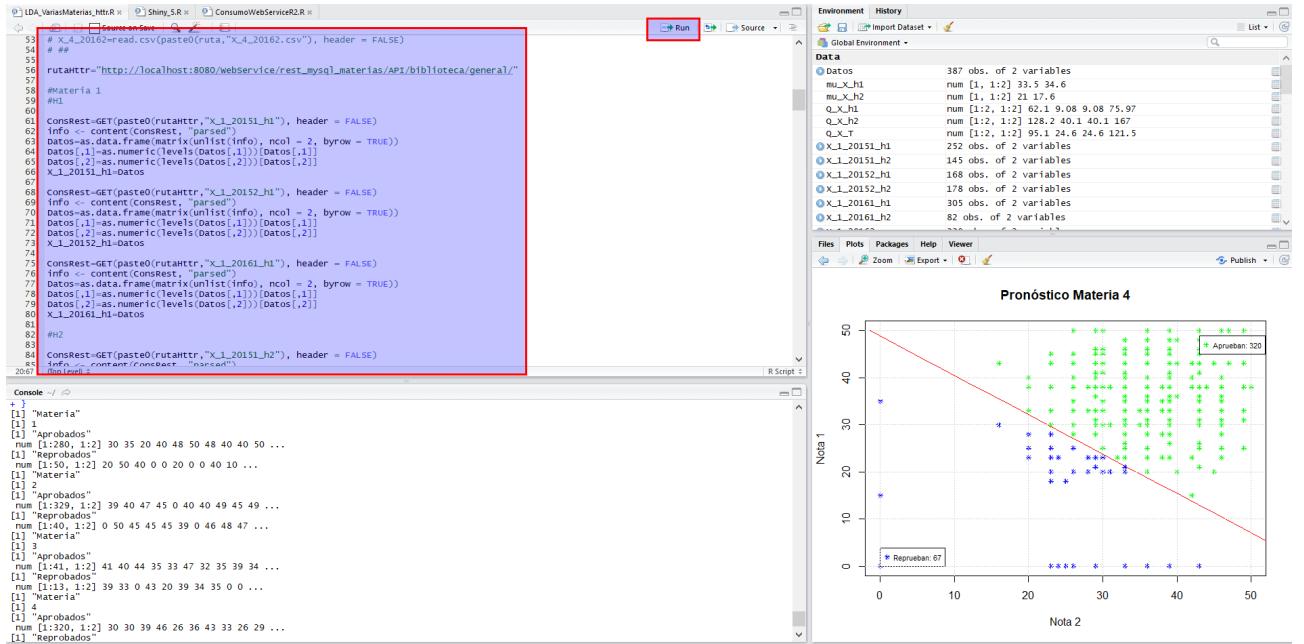


Figura 4.19: Cuenta en Shinyapps

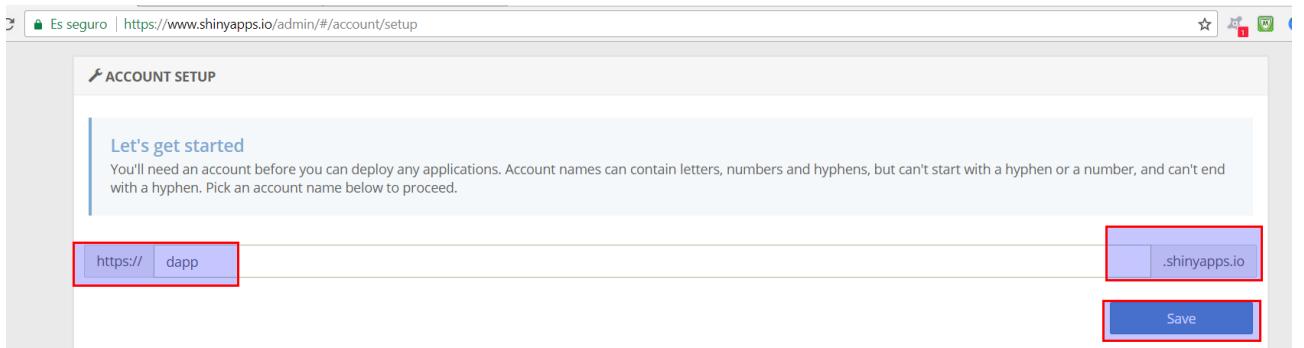


Figura 4.20: Token en Shinyapps

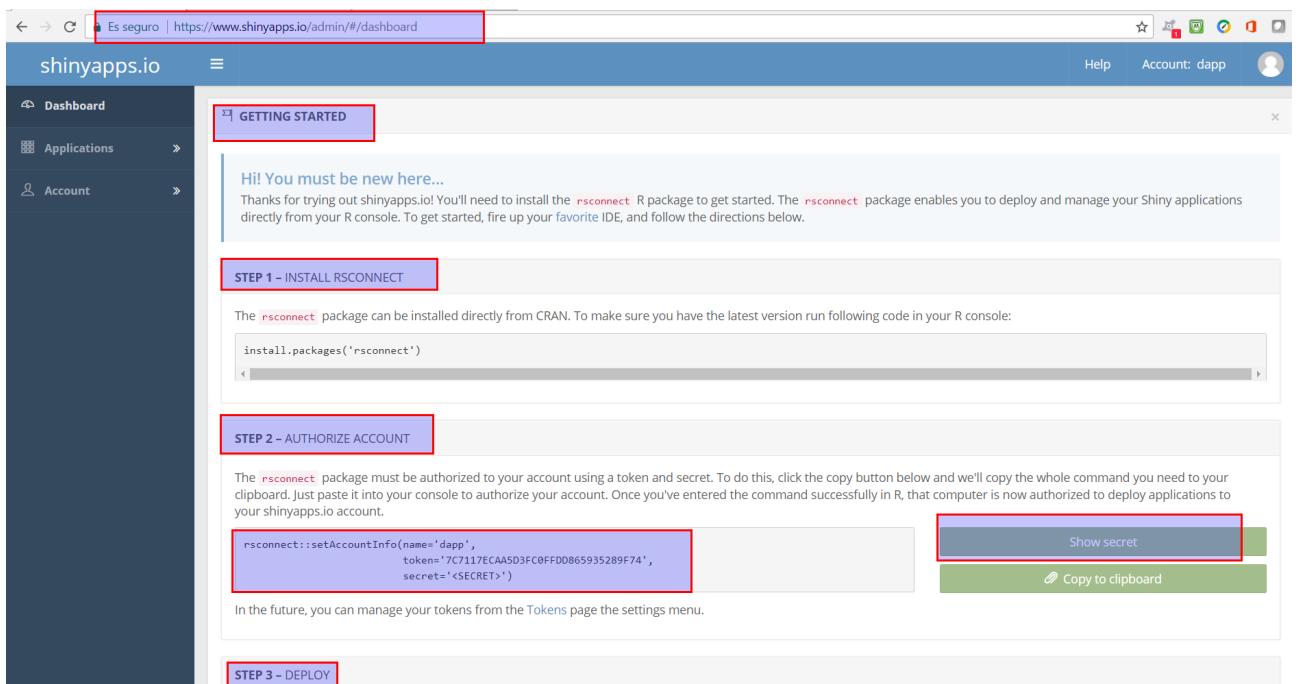


Figura 4.21: Despliegue ShinyApp

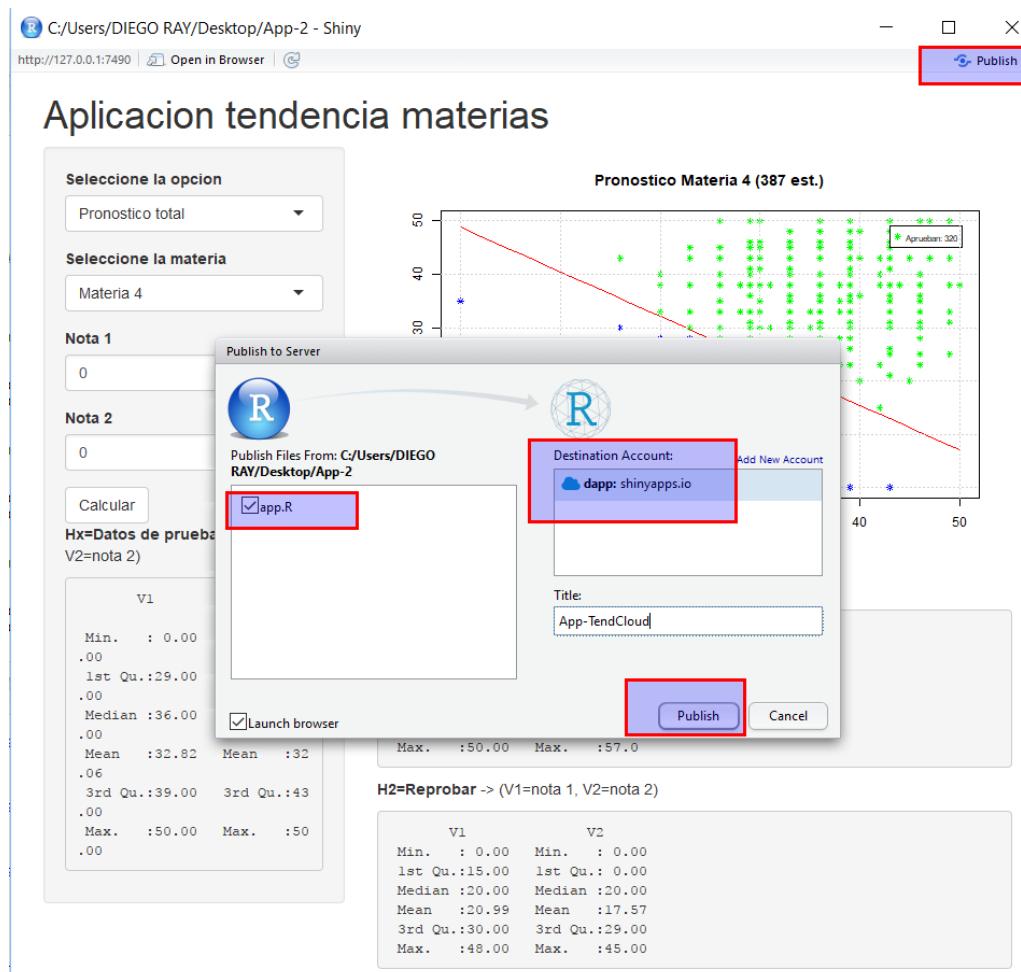


Figura 4.22: Estado y consumo de la aplicación en Shinyapps.io

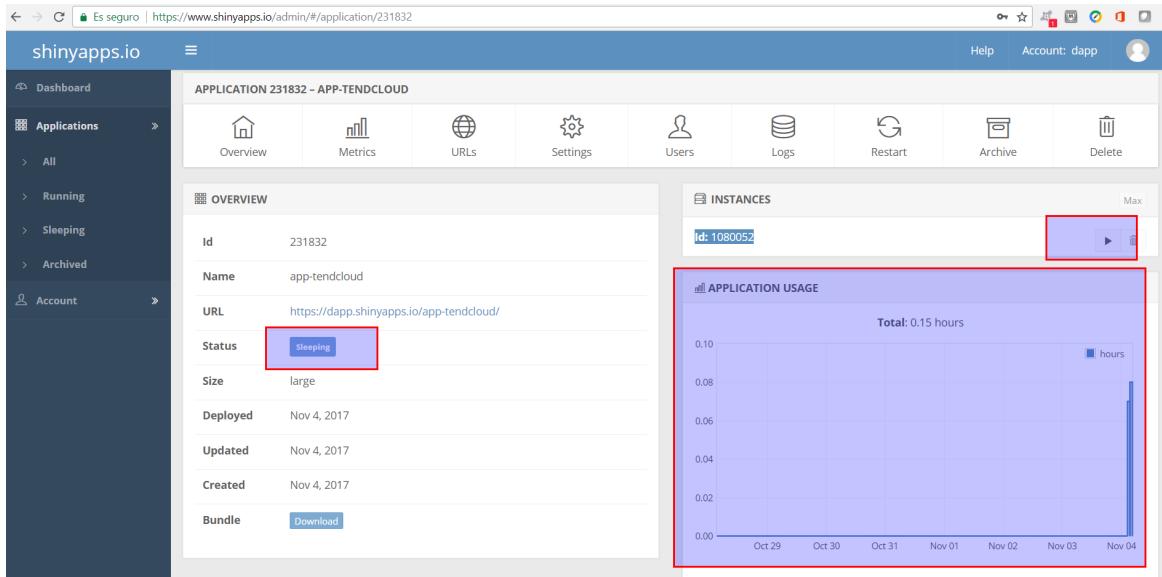


Figura 4.23: Inicio aplicación en ShinyApps.io

The screenshot shows a Shiny application titled 'Aplicacion tendencia materias'. The interface includes a dropdown menu for 'Opcion' (highlighted with a red box) with 'Tendencia' selected, another dropdown for 'Materia' (highlighted with a red box) with 'Materia 1' selected, and input fields for 'Nota 1' (0) and 'Nota 2' (0). A 'Calcular' button is also highlighted with a red box. Below the form, there is some R code: 'H1=Aprobar -> (V1=nota 1, V2=nota 2)' and 'H2=Reprobar -> (V1=nota 1, V2=nota 2)'. The browser address bar shows the URL https://dapp.shinyapps.io/app-tendencioua/.

Figura 4.24: Histórico de tendencia de aprobación de materia 1

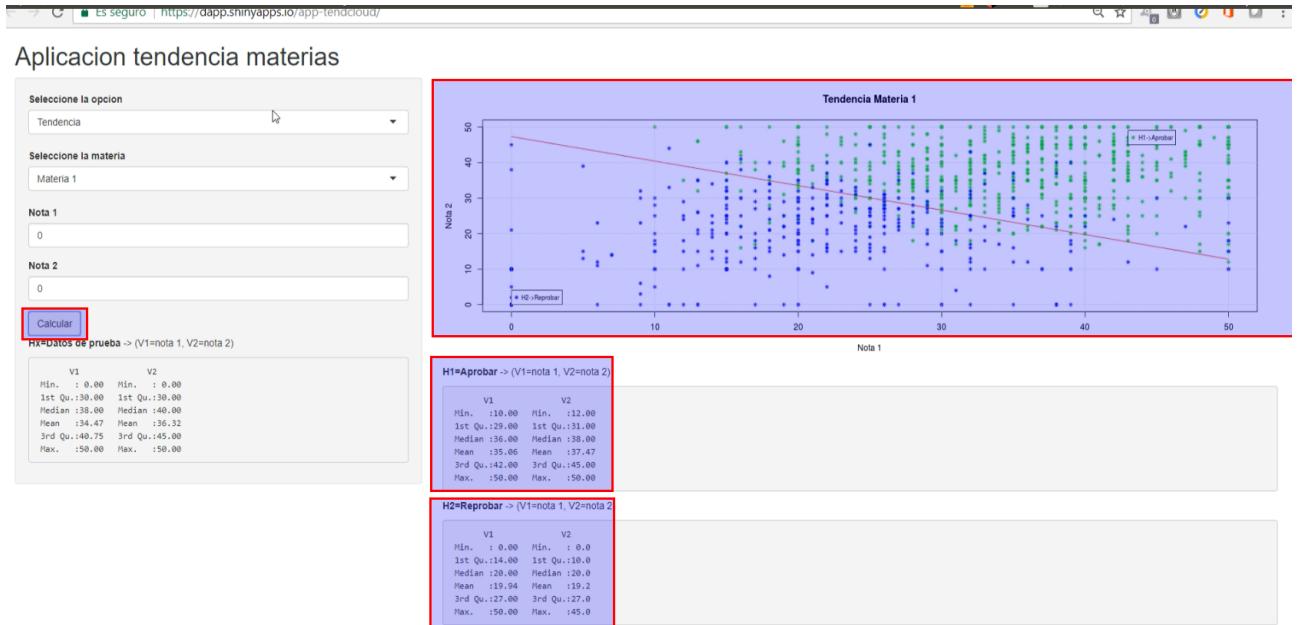


Figura 4.25: Clasificación de datos de prueba para materia 4

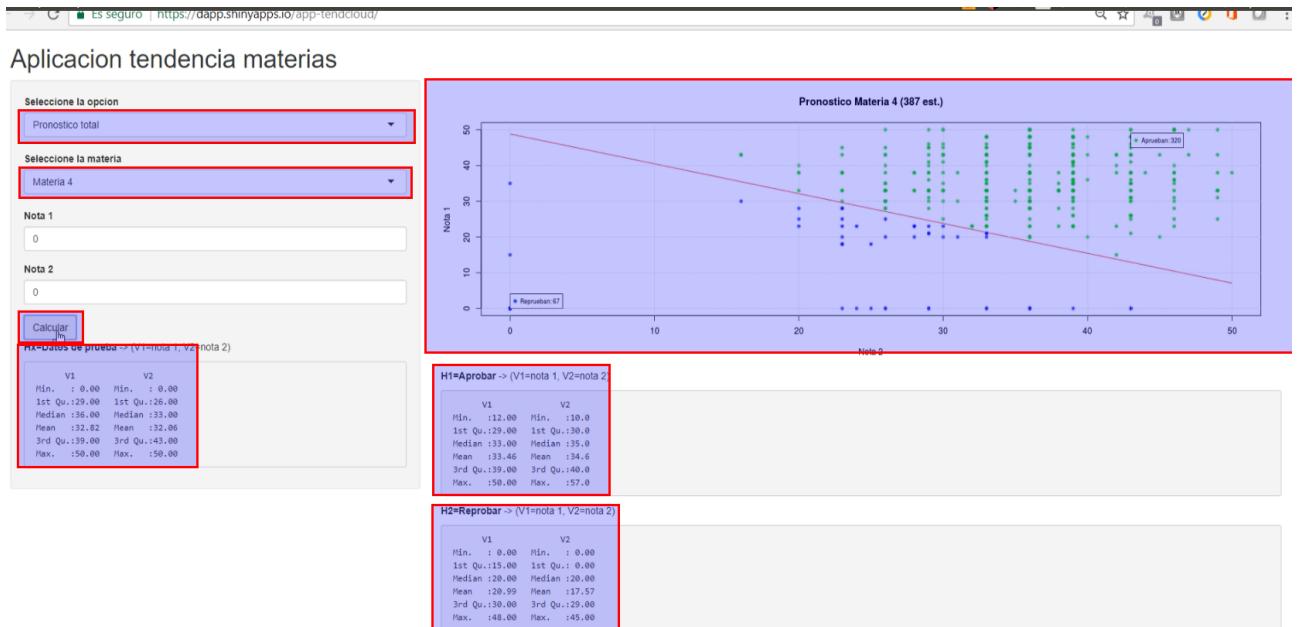
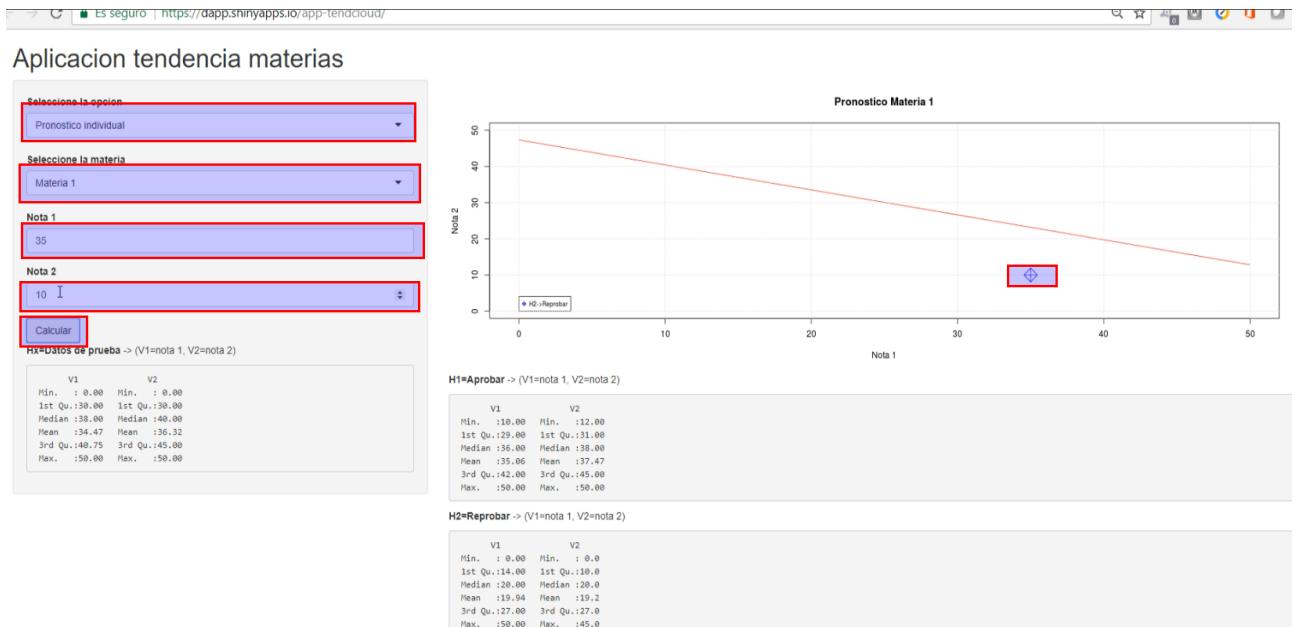
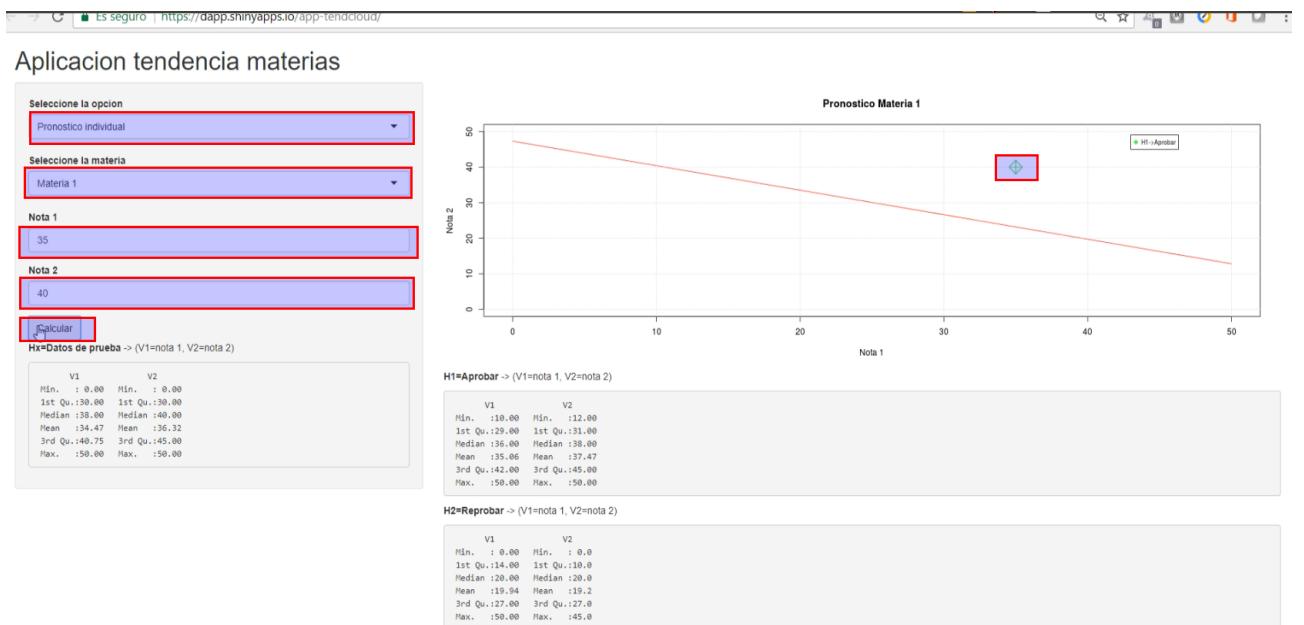


Figura 4.26: Clasificación individual de estudiante, pronóstico de tendencia *reprobado*

 Figura 4.27: Clasificación individual de estudiante, pronóstico de tendencia *aprobado*


Capítulo 5

Conclusiones

El método LDA fue suficiente para establecer una regla de decisión dadas las características de la información de periodos previos.

Una fortaleza del estudio realizado se encuentra en su indiferencia al docente que imparte cada materia, sólo se basa en los resultados de las dos primeras evaluaciones en relación con su resultado final.

Se incluyó el promedio ponderado del estudiante para mejorar el modelo, pero reveló que no tenía incidencia significativa en el resultado final de las materias.

El servicio web de tendencia de materias puede proporcionar información pertinente a los estudiantes y directivos académicos para que tomen decisiones con respecto a su desempeño.

El tener un pronóstico acerca de un posible número de estudiantes reprobados por materia permite realizar la planeación de recursos en cursos de nivelación con por lo menos un mes de anterioridad a tener la nota definitiva de cada materia.

El uso de tecnologías web permite ofrecer servicios de forma fácil y con disponibilidad de uso en distintas plataformas a un costo escalable según la demanda de consumo de información.

Las plataformas de computación en la nube son una alternativa para el desarrollo de aplicaciones de forma ágil y con costos bajos para prueba de prototipos de software como solución de negocio en diferentes contextos de la industria de la información y las comunicaciones, con opción de escalamiento y mantenibilidad.

Como trabajo futuro es posible desarrollar una regla de decisión más precisa mediante inferencia bayesiana para generar perfiles variables con menor error, e incluir más factores o variables que influyan en el resultado del proceso (aprobar o reprobar) y que permitan incluso predecir las notas de los

estudiantes de forma aproximada al comportamiento de cada materia, buscando no tener un elevado costo computacional en la implementación de los algoritmos a utilizar.

Bibliografía

- [1] M. Kantardzic, *Data Mining: Concepts, Models, Methods, and Algorithms, 2nd Edition.* John Wiley and Sons, 2nd ed., 2011.
- [2] A. Papoulis, *Probability, Random Variables and Stochastic Processes.* McGraw-Hill Companies, 3rd ed., 1991.
- [3] P. Kruchten, *The rational unified process: an introduction.* Addison-Wesley Professional, 2004.
- [4] J. Rumbaugh, I. Jacobson, and G. Booch, *The Unified Modeling Language Reference Manual.* Pearson Higher Education, 2004.
- [5] I. Sommerville, *Software engineering.* Pearson, 9th ed ed., 2011.
- [6] S. S. Abraham Silberschatz, Henry F. Korth, *Database system concepts.* McGraw-Hill, 4rev ed ed., 2002.