



UNIVERSIDAD NACIONAL AUTÓNOMA DE  
MÉXICO  
FACULTAD DE INGENIERÍA

Tesis que para obtener el Título de Ingeniero Mecatrónico

---

**Desarrollo de un Sistema de Seguimiento  
Académico de Estudiantes de Ingeniería  
Mecatrónica**

---

*Presenta:*

Rafael Ángel GUDIÑO REYES

*Tutor:*

M.I. Billy Arturo FLORES  
MEDERO NAVARRO

Marzo de 2015

## **Resumen**

Esta tesis presenta el desarrollo de un Sistema de Seguimiento Académico de Estudiantes de Ingeniería Mecatrónica de la Facultad de Ingeniería de la Universidad Nacional Autónoma de México. El sistema ayuda a los profesores asignados como responsables de coordinar el seguimiento académico de los alumnos a observar y controlar de manera más eficiente e intuitiva el progreso de los alumnos, así como a presentar dichas observaciones a las instituciones correspondientes a cargo de evaluar la situación académica de la Facultad, con el fin de ayudar a mejorar el desempeño estudiantil impactando en la calificación de los indicadores de desempeño de la UNAM.

# Índice general

|                                                        |           |
|--------------------------------------------------------|-----------|
| <b>1. Problemática</b>                                 | <b>2</b>  |
| 1.1. Introducción . . . . .                            | 2         |
| 1.2. Planteamiento del problema . . . . .              | 4         |
| 1.3. Alternativas actuales de solución . . . . .       | 5         |
| 1.4. Objetivos . . . . .                               | 6         |
| 1.4.1. Objetivo general . . . . .                      | 6         |
| 1.4.2. Objetivos específicos . . . . .                 | 6         |
| 1.5. Hipótesis . . . . .                               | 6         |
| 1.6. Alcances . . . . .                                | 7         |
| 1.7. Conclusión . . . . .                              | 7         |
| <b>2. Marco teórico</b>                                | <b>8</b>  |
| 2.1. Desarrollo de Software . . . . .                  | 8         |
| 2.2. Metodologías ágiles existentes . . . . .          | 11        |
| 2.3. Metodologías para un solo desarrollador . . . . . | 15        |
| 2.4. Conclusión . . . . .                              | 15        |
| <b>3. Desarrollo</b>                                   | <b>17</b> |
| 3.1. Diagnóstico . . . . .                             | 17        |
| 3.2. Prescripción . . . . .                            | 17        |
| 3.3. Implementación . . . . .                          | 18        |
| 3.3.1. Asignación de roles . . . . .                   | 18        |

|           |                                          |           |
|-----------|------------------------------------------|-----------|
| 3.3.2.    | Product Backlog . . . . .                | 18        |
| 3.3.3.    | Requisitos mínimos de hardware . . . . . | 19        |
| 3.4.      | Programación . . . . .                   | 20        |
| 3.4.1.    | Estructura . . . . .                     | 22        |
| 3.4.2.    | Flujo de navegación . . . . .            | 24        |
| 3.5.      | Resultados . . . . .                     | 39        |
| 3.6.      | Conclusión . . . . .                     | 39        |
| <b>4.</b> | <b>Conclusión general</b>                | <b>41</b> |
| 4.1.      | Conclusión . . . . .                     | 41        |
| 4.2.      | Trabajo a futuro . . . . .               | 42        |
|           | <b>Bibliografía</b>                      | <b>44</b> |
|           | <b>A. Manual de Usuario</b>              | <b>46</b> |

### **Agradecimientos**

Quiero agradecer a la Universidad Nacional Autónoma de México y a la Facultad de Ingeniería, que me han dado tanto, y a todos los que formaron parte de mi formación académica, incluyendo amigos, profesores y demás personas que me apoyaron en este periodo, y no me refiero sólo a con los que compartí trabajo en equipo si no también a los que hicieron de mi trayecto algo agradable. En especial a mi familia, mis padres y mis hermanas Flora y Noelia. Gracias también amigos. Gracias Pale, Carlos, Alan, Gina, Clau, Marce, Did, Marsilli, Poncho, Valente, Dieguito y todos los que me faltan que me han apoyado desde antes en otras etapas de mi vida.

Para mis padres,  
que no desistieron nunca  
e hicieron hasta lo imposible por  
mí.

# Capítulo 1

## Problemática

### 1.1. Introducción

Existen muchas formas y criterios de evaluar una institución de educación superior. Conforme a Lee (2013), el estatus de una universidad se confiere en ésta mediante rankings globales. Lee (2013) asevera que una universidad de calidad debe cumplir de manera excelente con los siguientes roles o características:

- La educación de sus estudiantes.
- Investigación, desarrollo y diseminación del conocimiento.
- Actividades que contribuyan a la comunidad y la sociedad.

La figura 1.1 describe cómo es que *Times Higher Education* evalúa a las universidades internacionales. Se puede observar que un 30% del criterio de evaluación se enfoca en la enseñanza y el entorno de estudio, incluyendo el progreso académico de los alumnos, lo cual representa un gran peso para el *ranking* de la universidad, convirtiéndolo en un foco de interés para la misma, junto con las áreas de investigación y reputación e influencia, dejando con un menor peso a la entrada a la industria y su panorama internacional.

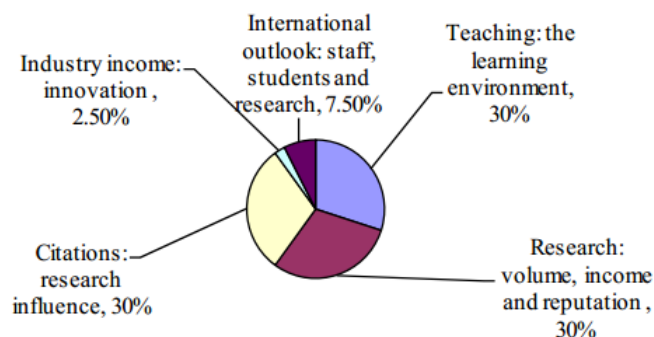


Figura 1.1: Indicadores de desempeño utilizados por *Times Higher Education World University Rankings 2012-2013*

Lee (2013)

UNESCO (1998), por su parte, indica que una de las principales motivaciones o misiones de la educación superior es:

“Educar estudiantes altamente capacitados y ciudadanos responsables, capaces de cubrir las necesidades de todos los sectores de la actividad humana, ofreciendo certificaciones relevantes, incluyendo entrenamiento profesional, el cual combina conocimiento de alto nivel y habilidades, usando cursos y contenido adaptado a las necesidades presentes y futuras de la sociedad.”

Respecto al mismo tema, la Universidad Nacional Autónoma de México está comprometida a impartir educación superior para formar profesionistas útiles a la sociedad, tratándose no solamente de que los jóvenes tengan la oportunidad de ingresar a los estudios universitarios, sino también de que permanezcan y en especial de que logren un egreso satisfactorio, de acuerdo con UNAM (2009) y UNAM (2012).

Para asegurar que la calidad de la educación de los estudiantes sea excelente, es necesario que éstos mantengan una educación de calidad continua, evitando que se atrasen o especialmente que trunquen sus estudios.

## 1.2. Planteamiento del problema

Actualmente, los Coordinadores y Jefes de Departamento de la Facultad de Ingeniería de la Universidad Nacional Autónoma de México, responsables del seguimiento académico de sus respectivos campos, no cuentan con una herramienta de software especializada para atender esta responsabilidad, lo cual dificulta la realización de dicha tarea de la forma más eficaz.

Debido a políticas de acceso de datos de la Universidad Nacional Autónoma de México, los Jefes y Coordinadores no tienen acceso total a la información académica de todos sus alumnos. A pesar de esto, los responsables del seguimiento académico tienen el derecho de pedir un avance parcial al término de cada periodo escolar, para poder llevar a cabo sus actividades de seguimiento académico.

Lamentablemente, la forma en la cual obtienen la información los responsables académicos no es la más adecuada, por lo cual, éstos se ven con la tarea de acomodar la información periodo a periodo para poder tener la información en una presentación más clara y ordenada, y así poder revisar el estado académico de los alumnos y dar paso al seguimiento académico.

Una segunda problemática con la cual se enfrentan los Jefes y Coordinadores de Departamento, es la ardua tarea que representa generar un formato de Cohortes que les es requerido desde una institución gubernamental por ser una institución de educación media superior.

Estas Cohortes son un compendio o resumen de la información más importante del progreso de los alumnos para facilitar el análisis de las instituciones correspondientes.

Mantener un estricto seguimiento académico de los alumnos es un gran paso para mejorar el estatus de la Facultad de Ingeniería, ya que permitiría elevar de manera directa el número de estudiantes titulados, mejorando la relación entre alumnos egresados y titulados, así como mejorar el promedio general de los alumnos.

Estos cambios no solo representan beneficios para la Facultad de Inge-

niería, sino también para los alumnos, base de una institución educativa. Al facilitar la identificación de los alumnos con problemas en su desempeño académico, los Coordinadores y responsables pueden invertir su tiempo en tomar acciones preventivas o correctivas para dichos alumnos, como puede ser acercarse a ellos para poder ofrecer ayuda y asistir en el correcto desarrollo escolar de los mismos.

### **1.3. Alternativas actuales de solución**

Principalmente, los Coordinadores y Jefes de Departamento se valen de la herramienta de software Microsoft Excel para realizar el seguimiento académico de sus alumnos debido a las siguientes razones:

- Es el formato en el cual reciben la información académica de sus alumnos.
- La gran familiarización con la suite ofimática de Microsoft la convierte en una herramienta relativamente fácil de usar.
- Es un software con gran penetración en el mercado, por lo cual es parcialmente fácil de compartir información bajo este software.

Al mismo tiempo, esta alternativa sufre de algunos inconvenientes, como son los siguientes:

- No es un software especializado, es decir, no está adaptado a las necesidades específicas de esta problemática.
- Es un software privativo, es decir, se requiere pagar por una licencia para poder utilizarlo.
- Sólo se puede utilizar bajo el Sistema Operativo Windows o OSX.
- Excel es manipulable, es decir, el usuario podría modificar los datos, alterando los resultados presentados.

Debido a las desventajas que representa el uso de Microsoft Windows, los responsables se encuentran con una difícil labor al tratar de acoplar sus necesidades a dicho software, por lo cual, la solución presentada en este documento deberá cubrir con todas las necesidades de los clientes y mejorar sobre lo ya establecido, superando las desventajas que supone la alternativa actual de solución.

## **1.4. Objetivos**

### **1.4.1. Objetivo general**

Desarrollar y presentar un sistema informático funcional que facilite el análisis de la información obtenida, listo para su uso por el usuario final, siendo éste el personal académico correspondiente, para que puedan, de una forma intuitiva y sencilla, revisar y analizar la información relacionada con el avance académico de los alumnos.

### **1.4.2. Objetivos específicos**

- Conocer la problemática y las diferentes formas de solución que actualmente utilizan los responsables de Departamento.
- Conocer a detalle la metodología a utilizar.
- Implementar la metodología para el desarrollo del sistema y conocer los resultados obtenidos.
- Análisis de la implementación de la metodología de acuerdo a los resultados obtenidos.

## **1.5. Hipótesis**

La problemática planteada se pretende resolver a través de la implementación de una metodología de diseño de sistemas informáticos, obteniendo un

producto final que cubra la necesidad solicitada por el cliente final, mejorando sobre la alternativa actual de solución, es decir, facilitando el análisis de la información sobre el seguimiento académico de los alumnos de Ingeniería Mecatrónica, comparado con la forma en la que se cubre esta necesidad hoy en día por parte de los Jefes y Coordinadores.

## **1.6. Alcances**

El propósito de esta tesis es desarrollar un software adaptado a las necesidades de los Jefes y Coordinadores de Departamento de la Facultad de Ingeniería de la Universidad Nacional Autónoma de México, tomando como caso de estudio a los alumnos de Ingeniería Mecatrónica, aunado a que dichas necesidades sean suficientemente generales para poder transportar lo realizado bajo este trabajo a otras carreras de la misma Facultad, generando un beneficio común entre éstas.

## **1.7. Conclusión**

En este capítulo se introdujo de manera concisa la magnitud de la problemática a enfrentar, así como el contexto en el cual se está desarrollando ésta. También se explicó cómo se maneja o sobrelleva la situación actualmente, resumiendo las ventajas y desventajas de dicho método.

Adicionalmente, se indicó a grandes rasgos cuáles serán los objetivos de éste proyecto, el resultado al que se espera llegar al finalizar el trabajo y los alcances bajo los que se desarrolla el proyecto, dejando clara la amplitud de éste.

## Capítulo 2

# Marco teórico

### 2.1. Desarrollo de Software

Hasta hace unas décadas, el desarrollo de software se conocía como una actividad caótica. De acuerdo con Bustos (2003), en la década de los 90's, la mayoría de los proyectos de desarrollo de software finalizaban excediendo ampliamente el presupuesto y con grandes retrasos de tiempo, además de que no ofrecían en su totalidad las características y funcionalidades que se planteaban originalmente.

Como menciona Letelier & Penadés (2006), hasta hace poco, el proceso de desarrollo llevaba asociado un marcado énfasis en el control del proceso mediante una definición rigurosa de actividades y documentación. Dicha metodología tradicional, comúnmente llamada Metodología Monumental o Metodología Pesada, ha demostrado ser efectiva y necesaria en proyectos de gran tamaño. Sin embargo, debido a las necesidades en la situación actual en lo que a informática se refiere, en donde los proyectos y el entorno son cambiantes y el tiempo de desarrollo debe ser reducido al máximo, este enfoque no resulta ser el más adecuado. Por consiguiente, una nueva metodología fue diseñada, con la cual se pueden observar resultados rápidos, aun cuando, a pesar de que la misión del proyecto sea clara, los requerimientos específicos

sean volátiles y evolutivos, conocida como Metodología Ágil, (Highsmith & Cockburn 2001).

**Definición 1** *“Una metodología de desarrollo de software impone un proceso disciplinado sobre el desarrollo de software con el objetivo de hacer de éste un procedimiento más predecible y eficiente. Por tanto, se plantea que una metodología define un camino reproducible para obtener resultados confiables.”*

*Bustos (2003)*

Dingsøyr, Dybå & Moe (2010) comentan que la Metodología Ágil ha tenido una influencia importante sobre cómo se conduce el desarrollo de software en los últimos tiempos. Se ha convertido en un término que funciona como plataforma para planificar desarrollos de software, coordinar el trabajo, establecer una forma de comunicación entre el cliente y los desarrolladores y organizar proyectos dentro de pequeñas, medianas y grandes empresas.

The Agile Alliance, una organización sin ánimo de lucro con membresía global comprometida a explorar y aplicar los principios de la Metodología Ágil y a promover dichos principios y conceptos (Agile Alliance 2015), acuñó el término *ágil* aplicado al desarrollo de software. Junto con dicho término, se generó en el año de 2001, en EEUU, un documento que resume este enfoque de desarrollo, conocido como el Manifiesto Ágil.

En el Manifiesto Ágil (Fowler & Highsmith 2001) se afirma que para implementar una metodología ágil, se deben tomar en cuenta los siguientes valores fundamentales:

- Los individuos e interacciones se valoran por encima de los procesos y las herramientas.
- El software funcional se valora por encima de la documentación extensiva.
- La colaboración con el cliente se valora por encima de la negociación del contrato.

- La respuesta al cambio se valora por encima del seguimiento de un plan.

Letelier & Penadés (2006) comenta, con base en los valores anteriores, que en la Metodología Ágil las personas que forman el equipo son el principal factor de éxito, construir un buen equipo es más importante que construir un buen entorno. Aclara que la regla a seguir es no producir documentación hasta la culminación del proyecto, a menos que sean de forma inmediata para tomar una decisión importante. Además, propone que se entable una comunicación o interacción constante entre el cliente y el equipo de desarrollo, la cual marcará la marcha del proyecto y asegurará su éxito. Finalmente concluye que la habilidad de responder a los cambios que surjan a lo largo del proyecto determina el éxito o fracaso del mismo, por lo tanto, la planificación no debe ser estricta sino flexible y abierta, es un tipo de metodología especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico.

En el cuadro 2.1 se resumen y comparan las características de estas dos grandes corrientes o enfoques de desarrollo de software. Como se puede observar, ambos enfoques aportan beneficios e inconvenientes, dependiendo del proyecto al que se quieran aplicar.

Después de analizar las ventajas y desventajas de cada uno de los enfoques, tomando en cuenta el proyecto en mano, se decidió que se ocuparía una metodología del tipo ágil para el desarrollo del proyecto, debido a que el proyecto se desarrollará por un solo desarrollador, se cuenta con interacción relativamente continua con el cliente, es decir, los Coordinadores y Jefes de Departamento, se prefiere un desarrollo rápido, aunque esto implique que la documentación sea realizada hasta finalizar el proyecto, no se requiere tanto énfasis en la arquitectura del desarrollo y tampoco se requiere de un proceso tan controlado o un contrato tan estricto, lo cual indica que la mejor opción en este caso es apoyarse de alguna de las metodologías ágiles existentes para el correcto desarrollo del proyecto, las cuales serán sopesadas y analizadas,

Cuadro 2.1: Comparativa entre Metodologías Ágiles y Metodologías Pesadas

| <b>Metodologías Ágiles</b>                                               | <b>Metodologías Pesadas</b>                                                        |
|--------------------------------------------------------------------------|------------------------------------------------------------------------------------|
| Basadas en heurísticas provenientes de prácticas de producción de código | Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo |
| Especialmente preparadas para cambios durante el proyecto                | Cierta resistencia a los cambios                                                   |
| Impuestas internamente                                                   | Impuestas externamente                                                             |
| Proceso poco controlado, con pocos principios                            | Proceso mucho más controlado, con varias políticas o normas                        |
| Contrato flexible                                                        | Contrato prefijado                                                                 |
| Mucha interacción con el cliente                                         | Interacción con el cliente reducida                                                |
| Grupo de desarrollo pequeño y trabajando en el mismo sitio               | Grupos grandes y posiblemente distribuidos                                         |
| Menos énfasis en la arquitectura del software                            | La arquitectura del software es esencial para el desarrollo                        |

Letelier & Penadés (2006)

para poder elegir alguna de ellas y cimentar el proyecto de Seguimiento Académico en ésta.

## 2.2. Metodologías ágiles existentes

Dentro de las metodologías ágiles, se evaluarán las principales para encontrar la que más se ajuste al ambiente de trabajo en el que se desarrollará el sistema.

Beck, Beedle, Van Bennekum, Cockburn, Cunningham, Fowler, Grenning, Highsmith, Hunt, Jeffries et al. (2001) y Letelier & Penadés (2006), resumen las metodologías ágiles y sus características y propiedades principales de la siguiente manera:

## ■ eXtreme Programming (XP)

Esta metodología es una de las metodologías ágiles más recurridas. Desarrollada por Kent Beck, es una metodología centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo del software, promoviendo el trabajo en equipo y propiciando un buen clima de trabajo.

Las características de su proceso de trabajo son las siguientes:

- Comienza con la creación de *user stories*, requerimientos del cliente.
- El equipo de trabajo evalúa cada *user story* y le asigna un costo, tiempo estimado de desarrollo.
- Las *user stories* son agrupadas para generar entregables incrementales.
- El equipo se compromete a una fecha de entrega para cada entregable.
- Después de la primera entrega, se analiza la velocidad de desarrollo para ayudar a definir las siguientes fechas de entrega.

## ■ Scrum

Propuesta por Ken Schwaber, Jeff Sutherland y Mike Beedle, Scrum es una metodología ágil indicada para proyectos con un cambio rápido de requisitos.

Define un marco para la gestión de proyectos que se ha utilizado con éxito durante los últimos años. Es un enfoque empírico que aplica las ideologías del control de procesos industriales al desarrollo de sistemas, lo cual reintroduce las ideas de flexibilidad, adaptabilidad y productividad.

Al comenzar, se genera una lista conocida como *Product Backlog*, en ésta se escriben todos los requerimientos conocidos, ordenándolos por

prioridad. Esta lista se actualiza constantemente con nuevos requerimientos, requerimientos más detallados y se reordena.

Consecuentemente, el desarrollo del software se realiza mediante iteraciones, denominadas *sprints*, donde se agrupan ciertos requerimientos del *Product Backlog* a producir, junto con un tiempo determinado de desarrollo.

El resultado de cada *sprint* es un incremento ejecutable que se muestra al cliente, conocido como *demo*, para verificar que el producto este cumpliendo con las expectativas de éste y que se dirija en el camino correcto.

Otra característica importante de la metodología Scrum son las juntas frecuentes con el cliente, las cuales tienen la característica de ser breves, donde se generan nuevos requerimientos para ser agregados a la lista previa.

#### ■ Crystal Methodologies

Se trata de un conjunto de metodologías para el desarrollo de software caracterizadas por estar centradas en las personas que componen el equipo y la simplificación del código. Han sido desarrolladas por Alistair Cockburn.

Sus características principales son:

- Es una familia de modelos de procesos que permiten *maniobrabilidad* basada en las características de los problemas.
- Comunicación cara a cara es enfatizada.
- Sugiere el uso de *reflection workshops* para revisar y analizar los hábitos de trabajo del equipo.
- En realidad es un conjunto de ejemplos de procesos ágiles a seguir que han resultado ser efectivos para diferentes tipos de proyectos.

- El objetivo es asistir al equipo de desarrollo a escoger el método de Crystal Methodologies más apropiado para el entorno de trabajo del proyecto.

#### ■ **Adaptive Software Development (ASD)**

Originalmente propuesta por Jim Highsmith, es una técnica para construir software complejo.

Se distingue por:

- Ser iterativo.
- Planeamiento basado en objetivos.
- Utiliza *time-boxing*, una técnica donde el tiempo estimado de entrega se divide en pequeños periodos de tiempo, donde cada periodo tiene su propio entregable y presupuesto.
- Enfatiza el aprendizaje en cada iteración.
- Cada ciclo se compone de tres fases esenciales: especulación, colaboración y aprendizaje.

#### ■ **Dynamic Systems Development Method (DSDM)**

Nace en 1994 con el objetivo de crear una metodología RAD unificada, promovida por el DSDM Consortium. En muchos campos es bastante similar a XP o ASD y es especialmente adecuada para sistemas que deben cumplir con un tiempo estimado de entrega corto.

Se basa en el *Principio de Pareto*, también conocido como la regla 80-20, indicando que el 80 % de la aplicación se puede entregar en el 20 % del tiempo que tomaría entregar el 100 % de la aplicación.

Propone cinco fases: estudio de viabilidad, estudio de negocio, modelado funcional, diseño y construcción y finalmente implementación. Las tres últimas son iterativas y además debe existir retroalimentación en todas las fases.

### 2.3. Metodologías para un solo desarrollador

Todas las metodologías mencionadas anteriormente, se apoyan demasiado en la suposición de que el desarrollo se realiza con un grupo de trabajo. Dado que este proyecto en específico se realizará mediante un solo programador, se realizarán algunos ajustes para poder aprovechar al máximo los beneficios del uso de una metodología, tratando de mantener el núcleo de las características, alterándolas lo menos posible para no corromper el propósito de éstas.

La metodología a seguir para este proyecto será la metodología Scrum, con las siguientes modificaciones para adaptarla a las características del entorno de éste proyecto:

- Las reuniones breves se mantendrán y serán juntas entre el desarrollador y el Jefe de Departamento y/o el Coordinador de Departamento.
- El desarrollador será el que fungirá como TODOS los roles dentro del esquema de personal de la metodología Scrum, es decir, el desarrollador fungirá como Developer Team y a su vez también será el Scrum Master.
- No se hará uso de la técnica Pair Programming.
- El *Product Backlog*, o lista de requerimientos, se desarrollará dentro de las juntas entre el desarrollador y el Jefe de Departamento o Coordinador de Departamento.
- Al final de cada *sprint*, el desarrollador presentará el avance mediante una junta con los responsables de Departamento.

### 2.4. Conclusión

En esta sección se valoraron los diferentes tipos de metodologías que permitirían asistir en el desarrollo del sistema, así como varias de las metodologías ágiles que se utilizan en la industria y por desarrolladores inde-

pendientes que han probado ser efectivas en diferentes tipos de proyectos. A lo largo de la investigación, se encontró que dichas metodologías confían en que el proyecto se realice mediante varios desarrolladores, lo cual no es el caso en éste, por lo tanto se propusieron ciertas modificaciones a la metodología Scrum, considerada como la más adecuada para el proyecto, en aras de poder aprovechar todos los beneficios que el uso de una metodología ofrece.

## Capítulo 3

# Desarrollo

### 3.1. Diagnóstico

Como ya se ha explicado anteriormente, en la sección de Introducción, los responsables de Departamento se enfrentan con la problemática de mantener la información de seguimiento académico ordenada y organizada, visible y lista para ser analizada. Otra de las problemáticas que se esperan resolver es la generación automática de Cohortes, así como mejorar sobre las desventajas que representa la alternativa actual de solución, en este caso el software Microsoft Excel, que fueron desarrolladas en el primer capítulo de este documento.

### 3.2. Prescripción

La solución que se propone es el desarrollo de un sistema informático adaptado específicamente a las necesidades del cliente, utilizando una alteración de la metodología Scrum, para poder ser desarrollada mediante un solo desarrollador.

Se ha seleccionado Java como principal lenguaje de programación, debido a que facilita enormemente la portabilidad del sistema entre varios sistemas

operativos y computadoras de diferentes características, además de que es una herramienta suficientemente poderosa para cumplir todos los requisitos del sistema, y se simulará la estructura de una base de datos utilizando la programación orientada a objetos, para mantener la simplicidad del proyecto.

### 3.3. Implementación

#### 3.3.1. Asignación de roles

En la metodología Scrum, hay tres tipos de roles, los cuales serán descritos a continuación, junto con sus representantes durante el desarrollo de este sistema.

- ***Product Owner***: Es el rol de las personas responsables del proyecto, quienes dictan cuales son las necesidades que se necesitan satisfacer, y en este caso serán los responsables de Departamento, es decir, el Coordinador y el Jefe de Departamento.
- ***Developer Team***: Es el equipo de trabajo que será encargado de desarrollar el sistema, escribiendo el código de la aplicación. En este caso será sólo uno, el desarrollador del sistema.
- ***Scrum Master***: Es el equivalente al *Project Manager*. También será representado por el desarrollador del sistema.

#### 3.3.2. Product Backlog

También conocidos como requerimientos, el siguiente listado describe todas las necesidades que se necesitaban cubrir, en voz de los responsables de Departamento, es decir, los *Product Owners*:

- Presentar de manera ordenada y concisa la información relacionada al avance académico de los alumnos.

- Ayudar al responsable a identificar a los alumnos que se encuentren en una situación de riesgo respecto a su avance académico.
- Ser una plataforma de fácil uso y mantenimiento para el usuario.
- Realizar análisis generales automáticos sobre la información obtenida a interpretar por el usuario.
- Generar de manera automática los formatos de Cohortes requeridos por las instituciones pertinentes.
- Permitir el uso simultáneo del sistema y la información ingresada entre un número reducido de responsables.
- Tener una estructura escalable, permitiendo así que el sistema se pueda utilizar por muchas generaciones y con una gran cantidad de alumnos.
- Presentar una interfaz sencilla e intuitiva para el ingreso de nuevos datos.
- Ayudar a elevar el nivel académico de la Facultad de Ingeniería mediante el tratamiento personal de casos académicos de bajo rendimiento.

### **3.3.3. Requisitos mínimos de hardware**

En conformidad con las especificaciones indicadas en la documentación de Java (Oracle 2015), los requisitos mínimos para ejecutar la aplicación son los siguientes:

- Sistema operativo:
  - Windows XP, Server, Vista, 7, 8+.
  - Oracle Linux 5.5+, Red Hat Enterprise, Ubuntu Linux 12.04+., Suse Linux 10+.

- Mac OSX 10.8.3+.
- RAM: 128MB
- Espacio en disco: 124 MB para JRE, 2 MB para Java Update.
- Procesador: Mínimo Pentium 2 a 266 MHz.
- Explorador: IE9+, Firefox, Chrome.

### 3.4. Programación

Todo el proyecto fue programado utilizando el IDE Eclipse, bajo el estándar de Java Beans y la arquitectura de programación de 3 capas.

Las tres capas fueron desarrolladas en este proyecto con los nombres de *db*, *controller* y *view*, para el estándar de tres capas de *modelo*, *controlador* y *vista*, respectivamente, junto con una carpeta *res* de recursos visuales, como se puede apreciar en la Fig. 3.1.

En la capa de modelo, en este caso *db*, fueron establecidas las clases básicas que servirán como transporte o envase de la información, y, como será explicado posteriormente, adoptaron una estructura muy similar a la de una base de datos relacional, pero con la ventaja de no tener que instalar un servicio en alguna computadora central o servidor, como en la bases de datos relacionales.

Para la capa de controlador, en este proyecto *controller*, se desarrollaron 4 clases controlador, cada una con un propósito diferente y especial. El primer controlador, denominado *DBController*, se encarga de hacer todas las lecturas y escrituras a la que será virtualmente una base de datos, pero con un esquema de Programación Orientada a Objetos. El segundo controlador, llamado *ExcelController*, se encarga de realizar todas las lecturas a los archivos de Excel y llevar la información a objetos y clases. El siguiente controlador se llama *FilesController*, y se encarga de guardar todos los archivos

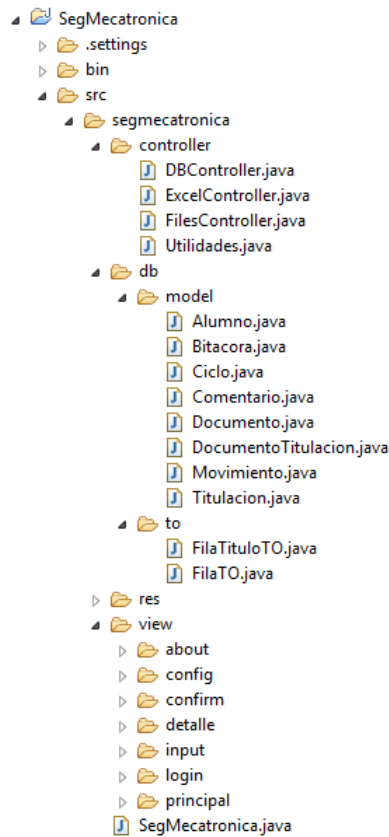


Figura 3.1: Arquitectura del proyecto

que sean necesarios en el disco duro. Por último tenemos el controlador *Utilidades*, que tiene la función de realizar operaciones internas comunes dentro del sistema.

Por último, la capa de la vista, *view* en este proyecto, se compone de todas las pantallas que serán mostradas al usuario final. Éstas fueron desarrolladas utilizando la tecnología conocida como JavaFX, la cual permite un gran nivel de personalización en el diseño de las mismas y que se ajusta a la forma de programación del lenguaje Java.

Para la distribución de este sistema se utiliza el formato de compresión de archivos JAR, el default para el lenguaje de programación Java, en una

configuración de *fat-jar*, el cual indica que las librerías y recursos que sean necesarios para la correcta ejecución del programa serán integrados dentro del mismo archivo JAR y no en un folder externo separado del resto del código.

### 3.4.1. Estructura

La estructura de la base de datos que se simulará en este proyecto queda representada de la siguiente manera, como se puede observar en la Fig. 3.2.

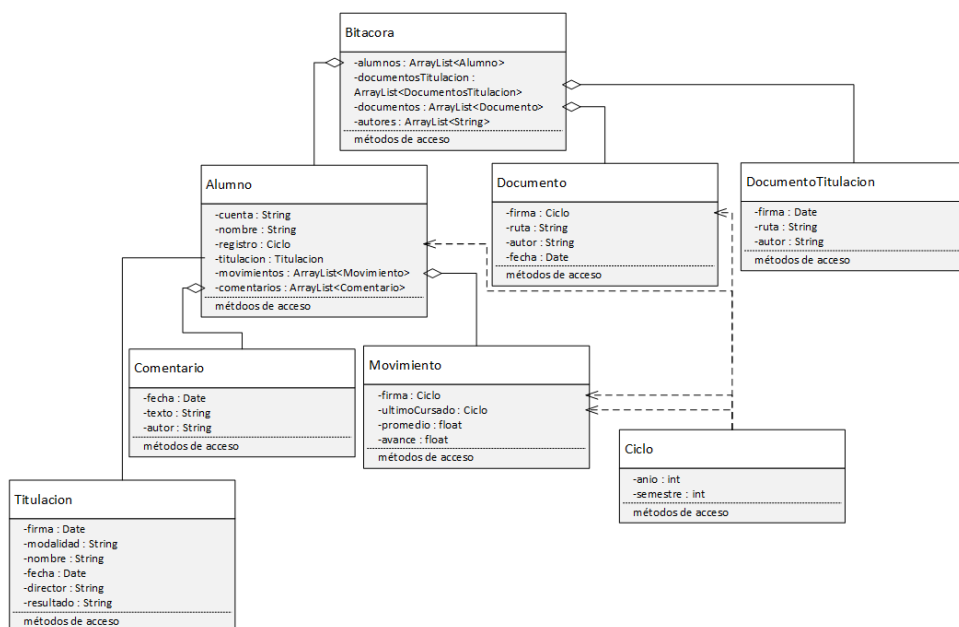


Figura 3.2: Estructura de la simulación de base de datos

La clase *Bitacora*, será un objeto en el cual queden guardados todos los registros que se vayan ingresando al sistema. Éste será el contenedor de toda la información, y el objeto que será guardado en memoria para poder acceder a él al cerrar el programa.

Antes de ser guardado en memoria, el sistema encripta la información mediante el algoritmo DES, Data Encryption Standard, mediante el cual se

oculta la información para que no pueda ser leída directamente al archivo que contiene el objeto *Bitacora*. Para que este algoritmo funcione, se debe guardar un archivo conteniendo la llave de encriptación, que será ubicada junto al archivo que contiene la bitácora. Ambos archivos llevarán el nombre de *configKey.dat* y *bitacora.ser*, respectivamente.

Dicha bitácora contiene tres listas principales, en las que está ordenada la información. Estas tres listas corresponden a la información de los alumnos y la información de los documentos que fueron ingresados para obtener la información de los alumnos. Ésta última se divide en dos, lista de documentos de avance académico, y listas de titulación, completando así las tres listas.

Cada registro de alumno se compone del número de cuenta, nombre completo, ciclo de registro, información sobre su titulación, y dos listas, las cuales contienen información sobre sus avances académicos, y sobre comentarios de algún responsable de Departamento, respectivamente.

La información sobre titulación se encuentra en una clase especial, llamada de igual manera, que se compone de una firma, para reconocer a qué documento de titulación pertenece, es decir, representa lo equivalente a una llave primaria en una base de datos relacional convencional. También incluye el tipo de modalidad, nombre (en caso de ser un trabajo escrito), fecha de titulación, director de tesis y resultado.

La información sobre los comentarios, al igual que la información sobre titulación, será guardada en una clase especial, llamada *Comentario*, que contiene la fecha de edición, el texto y el autor.

De igual manera, la información sobre los avances académicos, se guardará en una clase llamada *Movimiento*, la cual incluye el ciclo en la cual fue agregado, información sobre el último curso al que se inscribió el alumno, el promedio actual y el avance de créditos en porcentaje.

Para guardar la información sobre los documentos, se generaron dos clases especiales, una para documentos de avance académico, llamada sim-

plemente *Documento*, y otra para los documentos de titulación, llamada *DocumentoTitulacion*. Ambas clases contienen información sobre la fecha en la cual fueron agregados, la ruta de la cual fueron extraídos, y el autor, es decir, el usuario que las agregó.

Adicionalmente, se generó una clase auxiliar llamada *Ciclo*, una entidad que guarda información sobre un ciclo escolar, la cual incluye un año y un periodo semestral y apoya en la forma de ingreso de datos sobre ciclos escolares en las demás clases.

### 3.4.2. Flujo de navegación

Al iniciar el programa, éste guiará al usuario por una serie de pantallas que lo asistirán de manera ordenada en el correcto funcionamiento del sistema.

Todas las pantallas importantes cuentan con un botón ayuda en la parte superior derecha, con la cual se lanzará una pequeña pantalla mostrando los créditos y abriendo una liga al Manual de Usuario en formato PDF, como se puede visualizar en la Fig. 3.3.



Figura 3.3: Pantalla de Ayuda

También es importante aclarar que esas mismas pantallas tienen la capa-

cidad de mostrar advertencias en toda la parte superior de las mismas, con una barra de fondo amarillo y con letras negras, con las cuales se informa al usuario de las acciones que se están llevando a cabo o de mensajes de error.

#### 3.4.2.1. Selección de ruta de datos



Figura 3.4: Pantalla de Selección de ruta de datos

En la primera pantalla que se muestra (Fig. 3.4), en una computadora en la cual es nuevo el programa y no se ha corrido antes el mismo, se puede observar un campo de texto y un botón denominado Examinar. Se debe elegir una ruta en la cual se persistirán todos los datos, es decir, una instanciación de la clase *Bitacora*, que, como se explica en la sección anterior, almacena toda la información general de la aplicación.

**Definición 2** “*Instanciación se refiere al objeto en memoria generado a partir de una variable de tipo Clase.*”

Dicha ruta se almacena en los registros de aplicación del sistema operativo, haciendo uso de la Core Java Preferences API. Al inicio del programa, lo primero que se hace es verificar si ya existe un registro. En caso de que ya exista, indica que la aplicación ha sido ejecutada en dicha computadora y que ya se tiene la ruta en la cual se guardarán todos los datos. En caso de

que no exista, se abre esta pantalla y se invita al usuario al ingresar dicha ruta, sin la cual no puede seguir con el procedimiento.

#### 3.4.2.2. Selección de usuario

En esta pantalla, se muestra una lista de selección de usuario, como se observa en la Fig. 3.5.



Figura 3.5: Pantalla de Selección de usuario

En esta pantalla se tiene que elegir un usuario para poder acceder a la siguiente pantalla, permitiendo que todas las ediciones que realice en esta sesión se registren a nombre de dicho usuario. Dicha lista de usuarios se obtiene abriendo el archivo persistido que contiene la instanciación de la clase *Bitacora* en el disco duro, y accediendo a la misma.

Se omitió la inclusión de un sistema de verificación de identidad como una contraseña debido a que el número de usuarios será mínimo y que se maneja cierto nivel de confianza entre éstos, por lo cual se espera que cada usuario elija siempre sólo su propio usuario.

Otras opciones que se pueden realizar en esta pantalla son agregar y eliminar un nuevo usuario, tomando en cuenta que al eliminar un usuario no se elimina ningún otro tipo de información, ya que el nombre del usuario es sólo una referencia.

#### 3.4.2.3. Actualización de datos

**Sistema de Seguimiento Académico**

**2. Actualice calificaciones**

| Año                 | Sem | Agregador por | Agregado en |
|---------------------|-----|---------------|-------------|
| Tabla sin contenido |     |               |             |

Año Sem **Agregar** Eliminar

**3. Actualice datos de titulación**

Última Actualización

**Actualizar** **Quitar última**

**4. Actualice datos de carrera**

# total de semestres

**Atrás** **Aceptar**

Figura 3.6: Pantalla de Actualización de datos

En la pantalla de Actualización de datos (Fig. 3.6), se pueden modificar los datos con lo que se alimenta el sistema, es decir, ingresar y eliminar datos de calificaciones y avance de créditos, así como también los datos de titulación.

**Datos de calificaciones y avance de créditos** Para agregar datos de calificación y avance de créditos, se tiene que ingresar el documento en formato Excel 2010, junto con su correspondiente ciclo.

Primero se revisa que los datos correspondientes al año y al sean correctos y después se revisa que dicho ciclo no haya sido ya ingresado previamente. Si no es así, entonces se procede a pedir la ubicación del archivo que contiene los datos de dicho ciclo. A continuación se verifica que dicho archivo corresponda al formato de Excel 2010 XLSX y que contenga al menos una hoja incluida dentro de éste, utilizando la librería para Java de Apache, conocida como Apache POI, la cual permite leer y escribir documentos en formato Excel sin necesidad de tener instalado la suite ofimática de Microsoft.

Al pasar este filtro, se examina la fila que contiene los títulos de las columnas para obtener la posición específica del documento en la cual se encuentra cada columna. Esto se realiza mediante un algoritmo de medición de similitud de cadenas conocido como Distancia Jaro-Winkler.

La Distancia Jaro-Winkler se define mediante la Fórmula 3.1.

$$d_w = d_j + (lp(1 - d_j)) \quad (3.1)$$

donde  $d_w$  define a la distancia Jaro-Winkler,  $l$  se refiere a la longitud del prefijo común al inicio de ambas cadenas, con un máximo de 4,  $p$  es un escalar estándar que indica el peso que se le dará al prefijo y  $d_j$  indica la distancia Jaro, la cual se define con la Fórmula 3.2.

$$d_j = \begin{cases} 0 & m = 0 \\ \frac{1}{3} \left( \frac{m}{|s_1|} + \frac{m}{|s_2|} + \frac{m-t}{m} \right) & m < 0 \end{cases} \quad (3.2)$$

donde  $m$  es el número de caracteres que se repiten en ambas cadenas y  $t$  es la mitad del número de transposiciones que ocurren entre ambas.

Con este método, se compara cada título de columna del archivo de Excel con un conjunto de cadenas preestablecidas con las que tiene que concordar con un valor de 0.8 en la distancia Jaro-Winkler para poder ser identificado como coincidencia.

Si no encuentra al menos un valor con el que haga una coincidencia para alguna de las columnas necesarias, será rechazado con un mensaje de *Estructura del archivo incorrecta* en la barra de notificación.

Las columnas que son necesarias para este tipo de archivos en estricto orden son las siguientes:

1. CUENTA
2. NOMBRE
3. REGISTRO
4. ULTIMO\_CURSADO
5. PROMEDIO
6. AVANCE

En caso de que si se encuentren coincidencias para todas las columnas necesarias, se guardarán en la instancia de la clase *Bitacora* todos los registros de alumnos que cuenten con información válida. Esta validación se compone de varios filtros:

- Las columnas de CUENTA, NOMBRE, REGISTRO y ULTIMO\_CURSADO correspondan a un formato de texto.
- Las columnas de PROMEDIO y AVANCE correspondan a un formato numérico.
- La columna de CUENTA contenga un total de 9 dígitos.
- Las columnas de REGISTRO y ULTIMO\_CURSADO contengan un total de 5 dígitos cada una.

Cada fila del documento de Excel generará dos objetos en la base de datos, uno que contiene la información del alumno, instanciación de la clase

*Alumno* y otro objeto, instanciación de la clase *Movimiento*, la cual contiene información sobre los datos relevantes de dicho ciclo para tal alumno.

El sistema buscará si ya existe la instanciación de la clase *Alumno* para dicho alumno y en caso de que ya exista, no la sobrescribirá, sólo agregará la información actualizada para ese ciclo escolar a ese objeto. Si no encuentra registro de tal alumno, entonces pasará a ingresar ambos objetos a la base.

A la “base de datos” serán ingresados con una firma correspondiente al ciclo escolar al que pertenecen, lo cual permitirá que sean identificados de manera posterior, es decir, para saber cuándo fueron ingresados al sistema dichos datos.

De igual forma, se agregará una instanciación de la clase *Documento*, incluyendo firma, ciclo, autor y fecha, a la lista de documentos dentro del objeto *Bitacora*.

Al finalizar la indexación de todas las filas de dicho documento, se notificará al usuario en la barra de notificación de la pantalla de Actualización de datos.

Para eliminar el registro de información de algún ciclo escolar, se ingresa el ciclo escolar en los campos correspondientes, y el programa identificará dicho ciclo escolar, después lo comparará con la información actual dentro del sistema, y eliminará los que contengan una firma que coincida con el ciclo escolar ingresado. Además, se eliminará el objeto de *Documento* correspondiente en la *Bitacora*.

De nueva cuenta, al terminar de realizar dicho proceso, se notificará al usuario mediante el uso de la barra de notificación.

**Datos de titulación** Para agregar datos de titulación es necesario contar con un archivo de Excel con las siguientes columnas presentes en estricto orden:

1. CUENTA
2. MODALIDAD

3. TITULO
4. FECHA
5. DIRECTOR
6. RESULTADO

Después de verificar si el archivo contiene al menos una hoja, así como en el caso anterior de Datos de calificaciones y avance, el sistema buscará mediante el método de Jaro-Winkler, la ubicación de estas columnas dentro del archivo.

Si no las encontrara notificará al usuario de *Estructura del archivo inválida*.

Además de ubicarlas, revisará que éstas tengan el siguiente formato:

- Todas las columnas necesarias se compongan de un formato ya sea de texto o numérico.
- La columna CUENTA contenga en formato de texto un total de 8 o 9 dígitos.

Al reconocer las filas que cumplan con estas características, generará un objeto de la clase *Titulacion* por cada una de dichas filas y tratará de agregárselas a las instanciaciones ya almacenadas dentro de la *Bitacora*. Si no encuentra el alumno al que hace referencia, entonces no podrá salvar dicha información de titulación, ya que el objeto *Titulacion* es un atributo de la clase *Alumno*, y no puede ser almacenada sin alguna relación directa con un objeto *Alumno*. Dicha instanciación de la clase *Titulación* llevará como firma la fecha y hora en la cual fue agregada.

En caso de que ya exista un archivo de titulación asociado a algún alumno almacenado previamente, no sobrescribirá la información, sólo la descartará.

Adicionalmente, agregará un nuevo registro de la clase *DocumentoTitulacion* a la lista de esta clase dentro del objeto *Bitacora*, también llevando

como firma la fecha y hora en la que fue agregado, así como el autor responsable de subir dicha información.

Para eliminar un registro de titulación al accionar el botón de Quitar última actualización, se borrarán todos los objetos *Titulacion* que correspondan al último movimiento que se realizó y se tenga registrado en la lista de *DocumentoTitulacion* dentro del objeto *Bitacora*, reconociéndolo por la firma que contiene la fecha y la hora.

**Datos de carrera** En esta sección, se le pide al usuario que ingrese el valor que corresponda al número de semestres del que se compone el plan de estudios de la carrera que se encuentra analizando.

La lista de posibles opciones contiene las siguientes posibilidades: 8, 9, 10 y 11 semestres, ya que éstos son los más comunes dentro de los planes de estudio. Dependiendo del número de semestres que se elija y la información de avance y titulación de cada alumno, el programa podrá detectar el estatus en el que se encuentre el estudiante:

- **Titulado**, significa que el estudiante ha completado el 100 % de créditos y cumplió con el trámite de titulación.
- **Pasante**, se refiere a un estudiante que ha completado el 100 % de créditos pero no ha cumplido con el trámite de titulación.
- **Ok**, con el cual se identifica a un alumno que va al corriente en su avance académico, es decir, tiene el número de créditos correspondientes al semestre que se encuentra cursando.
- **Atrasado**, representa a un alumno que no tiene el número de créditos correspondiente a su semestre académico.
- **Art. 22**, significa que dicho alumno ha rebasado el número de semestres permitidos para poder tener el derecho a inscripción regular a sus materias. Dicho límite es el 50 % de la duración del plan de estudios

respectivo adicionado a la duración regular, 14 o 15 semestres para todas las carreras de la DIMEI. Esto viene expresado en el artículo 22 de la Legislación Universitaria.

- **Art. 24**, se refiere a un estudiante que ha rebasado el tiempo límite para el cumplimiento de la totalidad de créditos, el cual es el doble de tiempo establecido en el plan de estudios correspondiente. Después del cual, siendo estrictos, se debería realizar baja de estudios por parte la Universidad, como está estipulado en el artículo 24 de la Legislación Universitaria.

Éstos son calculados mediante el siguiente algoritmo:

$$\text{estatus} = \begin{cases} \text{Art. 24} & c \geq (s \times 2) \\ \text{Art. 22} & (s \times 2) > c \geq (s \times 1,5) \\ \text{Atrasado} & (s \times 1,5) > c \geq s \vee \\ & a < ((s - c) \times \frac{100}{s}) - tol \\ \text{Pasante} & a \geq 100 \wedge \neg \text{titulado} \\ \text{Titulado} & a \geq 100 \wedge \text{titulado} \end{cases} \quad (3.3)$$

donde  $c$  representa el número de ciclos cursados por el alumno,  $s$ , el número de semestres totales de la carrera,  $a$ , el avance actual del estudiante,  $tol$ , un rango en porcentaje de tolerancia respecto a la cantidad total de créditos que debería llevar cierto alumno para dado número de ciclos cursados, es decir, qué porcentaje de créditos se le permite al alumno atrasarse, antes de ser catalogado con estatus de Atrasado. Generalmente se ocupará el valor de 4 para la variable  $tol$ .

En caso de no caer en ninguna de las posibilidades anteriores, quiere decir que el estudiante todavía no completa créditos ni se ha titulado, pero tampoco se ha retrasado en su avance académico, por lo cual se le otorga el estatus de *Ok*.

| Cuenta               | Nombre                  | Registro<br>Año Sem | Último Cursado<br>Año Sem | Promedio | Avance | Status   | Comentario |
|----------------------|-------------------------|---------------------|---------------------------|----------|--------|----------|------------|
| 00000000000000000000 | ALFONSO, CARLOS ROBERTO | 2000 2              | 2007 1                    | 8.98     | 100.00 | Titulado |            |
| 00000000000000000000 | ALFONSO, CARLOS ROBERTO | 2000 2              | 2005 2                    | 8.72     | 100.00 | Titulado |            |
| 00000000000000000000 | ALFONSO, CARLOS ROBERTO | 2001 1              | 2006 1                    | 8.26     | 100.00 | Titulado |            |
| 00000000000000000000 | ALFONSO, CARLOS ROBERTO | 2001 1              | 2006 1                    | 9.40     | 104.03 | Titulado |            |
| 00000000000000000000 | ALFONSO, CARLOS ROBERTO | 2001 1              | 2005 2                    | 8.81     | 100.00 | Titulado |            |
| 00000000000000000000 | ALFONSO, CARLOS ROBERTO | 2001 1              | 2007 2                    | 8.46     | 107.16 | Titulado |            |
| 00000000000000000000 | ALFONSO, CARLOS ROBERTO | 2001 1              | 2006 1                    | 8.30     | 100.00 | Titulado |            |
| 00000000000000000000 | ALFONSO, CARLOS ROBERTO | 2001 1              | 2006 1                    | 8.43     | 100.00 | Titulado |            |
| 00000000000000000000 | ALFONSO, CARLOS ROBERTO | 2001 1              | 2005 2                    | 9.02     | 100.00 | Titulado |            |
| 00000000000000000000 | ALFONSO, CARLOS ROBERTO | 2001 2              | 2006 1                    | 9.06     | 100.00 | Titulado |            |
| 00000000000000000000 | ALFONSO, CARLOS ROBERTO | 2001 2              | 2008 1                    | 8.04     | 100.95 | Pasante  |            |
| 00000000000000000000 | ALFONSO, CARLOS ROBERTO | 2001 2              | 2006 2                    | 8.38     | 100.00 | Titulado |            |
| 00000000000000000000 | ALFONSO, CARLOS ROBERTO | 2001 2              | 2007 1                    | 8.11     | 100.00 | Titulado |            |
| 00000000000000000000 | ALFONSO, CARLOS ROBERTO | 2001 2              | 2006 1                    | 9.59     | 100.45 | Titulado |            |
| 00000000000000000000 | ALFONSO, CARLOS ROBERTO | 2001 2              | 2006 1                    | 9.07     | 100.00 | Titulado |            |

Figura 3.7: Pantalla principal en sección de Último Semestre

#### 3.4.2.4. Información de último semestre

Una vez que se actualizan todos los datos, se le permite al usuario ingresar a la pantalla principal, por default en la pestaña llamada Último Semestre, como se observa en la Fig. 3.7.

En esta sección se puede revisar el estado actual de todos los alumnos que se han ingresado en el sistema. Se compone de tres subsecciones. En la parte superior se encuentran los filtros, con los que se puede reorganizar y buscar estudiantes con ciertas características. Dichos filtros siempre están siendo monitoreados por cualquier cambio que se les realice, por lo que todos los cambios son en tiempo real.

Cabe destacar que se puede aplicar cualquier cantidad de filtros al mismo tiempo, para definir mejor la búsqueda. Además, los campos en los que se utilizan datos numéricos están configurados para ser límites inclusivos.

En los campos de Registro y Último Cursado, se espera que se ingrese un ciclo escolar en ambos casos, por lo cual, sólo se actualizará la tabla hasta

que se ingresen 5 dígitos en estos campos, que representen los 4 dígitos del año, y un último dígito que represente el semestre, juntos, sin espacios o guiones.

Todos estos cambios serán temporales y sólo modifican la forma en la que se presentan los datos. Usando estos filtros no se altera la información en el sistema de ninguna forma.

Para ordenar la lista en cualquiera de sus campos, la utilería que se utilizó, TableView, incluye un sistema de ordenamiento con múltiples criterios, es decir, utilizando múltiples columnas como criterio de ordenamiento, ascendente o descendente.

Respecto a la columna de comentarios, ésta sirve para ingresar observaciones respecto a cualquier alumno del que se tenga registro. Al dar doble clic en el espacio, se abrirá un campo de texto, y al finalizar el comentario y dar ENTER, éste guardará este comentario en una instanciación de la clase *Comentario*, y lo agregará al usuario seleccionado, antes de persistir dicho cambio a la “base de datos”. Esta instanciación contiene el nombre del autor de dicho comentario y la fecha en la cual fue añadido, para que estos detalles después sean visibles para otros usuarios.

Para eliminar un comentario lo único que se debe hacer es entrar al modo de edición de nuevo y eliminar el texto que ha sido escrito, dando ENTER al final para guardar los cambios.

Finalmente, en la parte inferior de esta sección se encuentra la opción de exportar la tabla que se muestra en ese momento a un formato de texto o Excel. Se puede exportar en un formato de vectores separados por comas, con una extensión tipo TXT, la cual puede ser leída por la mayoría de los editores de texto. En el caso de Excel se genera un documento con extensión XLSX para poder ser leído por la suite ofimática de Microsoft.

| Cuenta | Nombre | 2014-1 |        | 2014-2 |        | ****-* |        | ****-* |        | ****-* |        |
|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
|        |        | Prom.  | Avance | Prom.  | Avance | Prom.  | Avance | Prom.  | Avance | Prom.  | Avance |
|        |        | 6.90   | 11.16  | 6.90   | 11.16  | -      | -      | -      | -      | -      | -      |
|        |        | 6.75   | 21.38  | 6.75   | 21.38  | -      | -      | -      | -      | -      | -      |
|        |        | 9.25   | 95.25  | 9.26   | 99.29  | -      | -      | -      | -      | -      | -      |
|        |        | 8.38   | 15.44  | 8.38   | 15.44  | -      | -      | -      | -      | -      | -      |
|        |        | 8.74   | 93.35  | 8.78   | 96.67  | -      | -      | -      | -      | -      | -      |
|        |        | 8.74   | 85.04  | 8.74   | 99.05  | -      | -      | -      | -      | -      | -      |
|        |        | 8.38   | 76.48  | 8.27   | 82.66  | -      | -      | -      | -      | -      | -      |
|        |        | 7.51   | 65.32  | 7.51   | 65.32  | -      | -      | -      | -      | -      | -      |
|        |        | 7.79   | 74.35  | 8.02   | 85.04  | -      | -      | -      | -      | -      | -      |
|        |        | 8.59   | 91.45  | 8.59   | 97.15  | -      | -      | -      | -      | -      | -      |
|        |        | 8.25   | 93.59  | 8.31   | 100.24 | -      | -      | -      | -      | -      | -      |
|        |        | 7.53   | 75.53  | 7.53   | 82.42  | -      | -      | -      | -      | -      | -      |
|        |        | 9.00   | 82.66  | 9.09   | 92.87  | -      | -      | -      | -      | -      | -      |
|        |        | 6.14   | 4.99   | 6.63   | 9.03   | -      | -      | -      | -      | -      | -      |
|        |        | 9.11   | 90.97  | 9.16   | 96.20  | -      | -      | -      | -      | -      | -      |
|        |        | 7.45   | 57.48  | 7.51   | 68.41  | -      | -      | -      | -      | -      | -      |
|        |        | 9.47   | 84.32  | 9.52   | 100.48 | -      | -      | -      | -      | -      | -      |
|        |        | 8.53   | 86.46  | 8.67   | 96.30  | -      | -      | -      | -      | -      | -      |

Figura 3.8: Pantalla Principal en sección de Historia Académica

### 3.4.2.5. Información de historia académica

En la pestaña denominada Historia Académica, mostrada en la Fig. 3.8, de la pantalla principal, se puede ver a detalle el progreso de cada alumno registrado en el sistema.

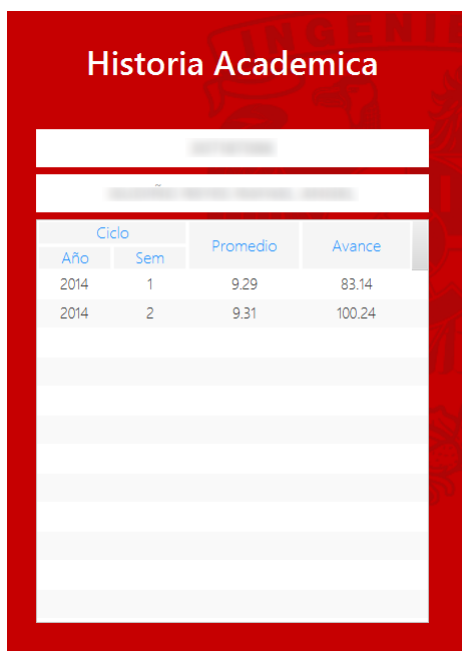
En esta sección sólo se encuentra un tipo de filtro, el de cuenta y nombre, aunque de igual manera que en la pestaña anterior, Último Semestre, todos los cambios que se hagan se harán en tiempo real, modificando el contenido de la tabla. Así mismo también se puede controlar el orden y el criterio mediante las columnas de la tabla.

En la tabla se muestra la historia académica de los últimos cinco ciclos escolares de los alumnos registrados en el sistema, y se muestra su avance en créditos y en promedio en cada uno de dichos ciclos. Como ayuda visual, se presentan con un fondo rojo o verde, dependiendo si comparado con el ciclo anterior fue una mejora o un atraso académico.

En el caso del avance en créditos, se cuenta como mejora el hecho de haber avanzado los créditos correspondientes a un ciclo escolar. Respecto a

la comparativa entre promedio, se cuenta como mejora sólo si el promedio general aumentó.

En caso de que se desee analizar toda la historia académica de alguno de los alumnos y no sólo de los último cinco ciclos, se puede dar clic en el número de cuenta de dicho alumno y se mostrará una segunda pantalla que asistirá en dicha tarea, como se muestra en la Fig. 3.9.



| Historia Académica |     |          |        |
|--------------------|-----|----------|--------|
| Ciclo              |     |          |        |
| Año                | Sem | Promedio | Avance |
| 2014               | 1   | 9.29     | 83.14  |
| 2014               | 2   | 9.31     | 100.24 |
|                    |     |          |        |
|                    |     |          |        |
|                    |     |          |        |
|                    |     |          |        |
|                    |     |          |        |
|                    |     |          |        |

Figura 3.9: Pantalla Secundaria de detalle de Historia Académica

En tal pantalla se mostrará en forma de lista toda la información respecto al avance en créditos y promedio del alumno, sin importar el número de actualizaciones que tenga éste.

#### 3.4.2.6. Información de análisis

La última sección de la pantalla principal, llamada Análisis (Fig. 3.10), se utiliza para mostrar al usuario resúmenes e información condensada del conjunto de datos ingresados en el sistema.

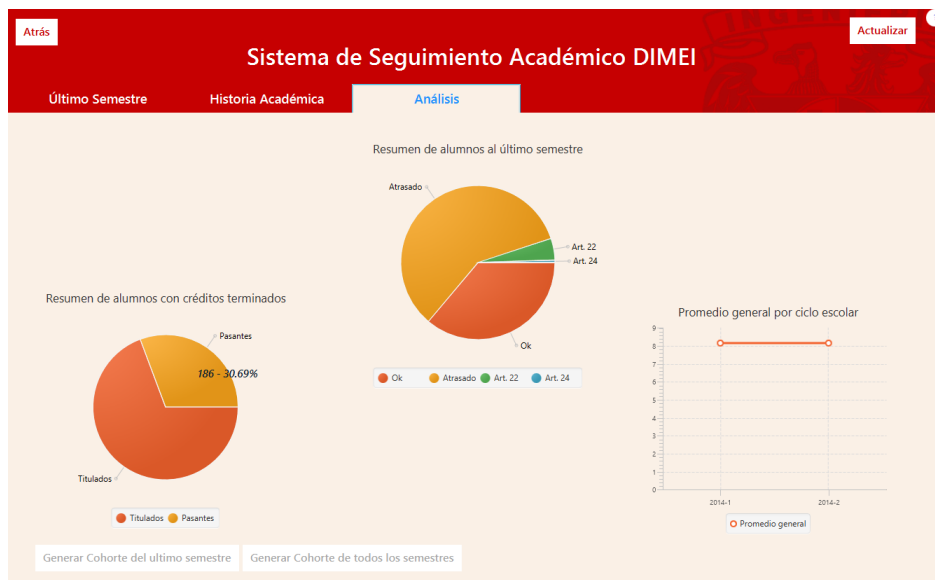


Figura 3.10: Pantalla Principal en sección de Análisis

En la sección superior de la pantalla se encuentran tres gráficas que ayudan a visualizar de manera más resumida toda la información que se encuentra contenida en el sistema.

La primera de ellas se enfoca en los alumnos que ya han completado todos sus créditos correspondientes. Muestra cuántos de ellos se han titulado y cuántos de ellos faltan por titularse.

La segunda, a su vez, se enfoca en aquellos alumnos que siguen cursando sus estudios, es decir, que no han concluido su 100 % de créditos. Resalta cuáles de ellos se encuentran en qué estatus, Ok, Atrasado, Art. 22 o Art. 24.

En estas dos primeras gráficas, que se muestran en forma de gráfica de pastel, se pueden ver a detalle los datos en los que están basadas dando clic derecho en cualquiera de sus secciones.

La última gráfica muestra simplemente el promedio general de los estudiantes que todavía no terminan sus créditos, semestre a semestre.

Por otro lado, en la sección inferior, se encuentran dos botones que tiene la capacidad de generar archivos de Excel con extensión XLSX, que contengan las Cohortes que describen. El primero de los botones generará la Cohorte del Último semestre, es decir, la Cohorte en la cual se muestran los porcentajes de créditos terminados al semestre actual de todos los alumnos, divididos por ciclo escolar de Registro, también incluyendo información sobre qué alumnos que ya terminaron sus créditos siguen sin titularse.

El segundo botón generará la Cohorte de todos los semestres, la cual incluye, generación a generación, cuántos alumnos continuaron con sus estudios.

### **3.5. Resultados**

Al terminar el desarrollo del sistema se comprimió todo el código en un archivo JAR, para poder distribuirlo a los clientes, como usuarios finales, y pudieran realizar pruebas en sus computadoras personales.

Durante la sesión de pruebas se encontraron mejoras menores que pudieron realizarse en un corto lapso de tiempo para poder entregar el producto final.

Con la finalización del producto y la sesión de pruebas se dio por terminada la implementación de la metodología utilizada.

### **3.6. Conclusión**

En este capítulo se describe paso a paso la implementación que se realizó de la metodología propuesta en el capítulo anterior.

Se desarrolló de manera detallada y minuciosa la forma en la cual ha sido programado el sistema completo, desde los inicios de planificación y bases del diseño, incluyendo requerimientos y definición de roles de metodología y requisitos mínimos del sistema para poder ser ejecutado, hasta explicar la forma en la que se estructuró el programa y todos los algoritmos que se

implementaron en éste.

Se desplegó toda la información relevante del código del programa, comentarios y explicaciones de la forma en la que se fue desarrollando.

Al optar por un esquema de metodología de un solo desarrollador, se observó que al implementarlo se generan cambios que, aunque parezcan mínimos u obvios, permiten una mejor organización para éste, manteniendo los márgenes de propiedad del software indicados e impulsando a generar ciclos de desarrollo cortos, promoviendo el contacto constante con los clientes para recibir retroalimentación directa y, en general, entregando un sistema de manera más temprana y funcional.

## Capítulo 4

# Conclusión general

### 4.1. Conclusión

El continuo desarrollo de las formas de evaluación de las instituciones de educación superior asegura que los estudiantes mantengan un buen nivel académico. La realización de este proyecto no sólo ayudará a los responsables de Departamento si no que mejorará la calidad de la educación que obtengan los alumnos de Ingeniería Mecatrónica en la Facultad de Ingeniería.

El uso de una metodología como la que fue utilizada en este proyecto también ayuda a definirlo y a organizar el progreso del mismo, manteniendo metas periodos cortos, minimizando el tiempo en el cual se pueden ver resultados funcionales. Se recomienda utilizar una metodología como la que ha sido utilizada en este proyecto para poder mantener un control sobre el desarrollo del sistema, aunque se trate de un solo desarrollador. Se observó que a pesar de no haber encontrado mucha información sobre metodologías de un solo desarrollador, las metodologías existentes para equipos de trabajo se pueden extrapolar con ciertas modificaciones para que funcionen del mismo modo, o al menos de forma similar, que lo harían para un conjunto de varios desarrolladores.

Se espera que con todas las formas de configuración del programa, éste

pueda ser utilizado por muchos ciclos escolares, independientemente de cuánta información se almacene dentro del sistema o de los cambios internos de la facultad respecto a planes de estudio o forma de ingreso a las carreras.

Se observó una reacción positiva ante la presentación del producto final del Sistema de Seguimiento Académico por parte de los responsables de Departamento, que serán los propietarios y usuarios finales.

También se espera que la facultad y sus alumnos se vean beneficiados mediante la integración de este sistema a la forma en la que los responsables de Departamento realizan sus obligaciones de seguimiento académico y que los resultados sean visibles para ambas partes.

Se concluye que el sistema fue completado con las funciones propuestas por los propietarios del software y necesarias para su correcto funcionamiento, mediante el uso de la adaptación de una metodología popular del desarrollo de software, la metodología SCRUM, con modificaciones para ser implementada por un solo desarrollador.

## **4.2. Trabajo a futuro**

Se pretende agregar una sección que incluya a los alumnos que ya han empezado su proceso de titulación, pero por alguna razón todavía no lo han acabado. Dicha sección funcionará de manera muy similar a la actual categoría de Titulados, con la misma forma de ingreso de datos y visualización de resultados y resúmenes.

Se recomienda continuar con el desarrollo del sistema, mejorando la forma en la que se determina si un estudiante se ha atrasado o no respecto a su plan de estudios, mediante modificaciones a los rangos utilizados en el estado actual del algoritmo en este proyecto para la categorización de alumnos por estatus. Posiblemente pueda ser incluida una sección de ingreso del Plan de estudios para poder identificar exactamente en qué materias se atrasan los alumnos, y poder prestar atención a las materias con más incidencia en atrasos académicos, incluso pudiéndolo detallar hasta los grupos en los

cuales se están atrasando dichos alumnos.

Otra recomendación que se hace para la mejora del sistema sería realizar pruebas para diferentes carreras de la Facultad, no solo las incluidas en la División de Ingeniería Mecánica e Industrial. Ya que se puede plantear que este sistema posiblemente funcione también para otras carreras que tengan el mismo tipo de acceso a la información cómo lo tiene la DIMEI.

Otro aporte que se le puede agregar al sistema es una forma de contactar a los alumnos directamente desde el programa, ya sea en forma de correo electrónico a los alumnos o a sus tutores y así llevar un contacto más cercano con éstos.

El código fuente será otorgado en su totalidad a los propietarios del software, responsables de Departamento, para que tengan control íntegro sobre el sistema y puedan implementar en su debido tiempo los cambios, modificaciones y mejoras que quisieran agregar al sistema.

# Bibliografía

- Agile Alliance, T. (2015), ‘Home’, <http://www.agilealliance.org/>.
- Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R. et al. (2001), ‘Manifesto for agile software development’.
- Bustos, R. A. G. (2003), ‘Métodos de desarrollo de software: El desafío pendiente de la estandarización’, *Theoria* **12**, 23–42.
- Dingsøyr, T., Dybå, T. & Moe, N. B. (2010), Agile software development: an introduction and overview, *in* ‘Agile Software Development’, Springer, pp. 1–13.
- Fowler, M. & Highsmith, J. (2001), ‘The agile manifesto’, *Software Development* **9**(8), 28–35.
- Highsmith, J. & Cockburn, A. (2001), ‘Agile software development: The business of innovation’, *Computer* **34**(9), 120–127.
- Lee, L.-S. S. (2013), ‘Achieving quality assurance and moving to a world class university in the 21st century.’, *Online Submission* .
- Letelier, P. & Penadés, M. C. (2006), ‘Metodologías ágiles para el desarrollo de software: extreme programming (xp)’.
- Oracle (2015), ‘What are the system requirements for java?’, <http://java.com/en/download/help/sysreq.xml>.

UNAM (2009), 'Acerca de la unam', <http://www.unam.mx/acercaunam/es/organizacion/index.html>.

UNAM (2012), 'Plan de desarrollo de la universidad 2011-2015', [http://www.dgi.unam.mx/rector/informes\\_pdf/PDI2011-2015.pdf](http://www.dgi.unam.mx/rector/informes_pdf/PDI2011-2015.pdf).

UNESCO (1998), 'World declaration on higher education for the twenty-first century: Vision and action.', [http://www.unesco.org/education/educprog/wche/declaration\\_eng.htm](http://www.unesco.org/education/educprog/wche/declaration_eng.htm).

## Apéndice A

# Manual de Usuario

El siguiente documento es el Manual de Usuario que será integrado en el sistema de forma digital, el cual guía al usuario a navegar por el sistema y resuelve dudas comunes respecto a su uso.



*Sistema de Seguimiento  
Académico DIMEI*

MANUAL DE USUARIO



**Rafael Ángel Gudiño Reyes**

2015

# Índice general

|                                          |           |
|------------------------------------------|-----------|
| <b>1. Requisitos mínimos</b>             | <b>2</b>  |
| <b>2. Instalación</b>                    | <b>4</b>  |
| <b>3. Ayuda y acerca de</b>              | <b>5</b>  |
| <b>4. Archivos de configuración</b>      | <b>6</b>  |
| <b>5. Usuarios</b>                       | <b>9</b>  |
| <b>6. Actualización de datos</b>         | <b>11</b> |
| 6.1. Datos de avance académico . . . . . | 11        |
| 6.2. Datos de titulación . . . . .       | 13        |
| 6.3. Datos de carrera . . . . .          | 15        |
| <b>7. Pantalla principal</b>             | <b>16</b> |
| 7.1. Último semestre . . . . .           | 16        |
| 7.2. Historia académica . . . . .        | 19        |
| 7.3. Análisis . . . . .                  | 21        |

# Capítulo 1

## Requisitos mínimos

El primer requisito para poder correr la aplicación es tener instalado Java Runtime Environment 8 o superior en una computadora, lo cual implica cumplir con las siguientes condiciones:

- Sistema operativo:
  - Windows XP, Server, Vista, 7, 8+.
  - Oracle Linux 5.5+, Red Hat Enterprise, Ubuntu Linux 12.04+., Suse Linux 10+.
  - Mac OSX 10.8.3+.
- RAM: 128MB
- Espacio en disco: 124 MB para JRE, 2 MB para Java Update.
- Procesador: Mínimo Pentium 2 266 MHz.
- Explorador: IE9+, Firefox, Chrome.

Se hace especial énfasis en la versión del Java Runtime Environment, debido a que el programa no podrá ser ejecutado con versiones anteriores a la versión 8 del mismo.

Adicionalmente, se recomienda tener instalado un lector de archivos PDF, debido a que el programa incluye este mismo manual de usuario de forma electrónica en formato PDF.

Es importante indicar que este programa no necesita de conexión de ningún tipo, Internet o Intranet, para poder ser ejecutado, aunque es posible que, si así fue configurado, requiera una conexión a Intranet para poder

acceder a datos remotos, más no para ser ejecutado (más información en la sección de Archivos de Configuración).

## Capítulo 2

# Instalación

El sistema por sí solo no requiere de ningún tipo de instalación, es suficiente con dar doble clic en el archivo ejecutable JAR (Fig. 3.1) para correr inmediatamente el programa, lo cual permite un gran nivel de portabilidad.

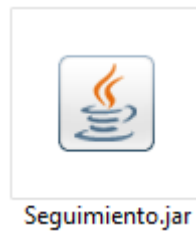


Figura 2.1: Archivo ejecutable JAR

A pesar de que el programa no está instalado en la computadora, la información con la que se alimentará el programa si se encontrará almacenada en el disco duro de la computadora.

## Capítulo 3

# Ayuda y acerca de

En todas las pantallas principales del sistema, ubicado en la esquina superior derecha, se encontrará un botón blanco con un signo de interrogación ? que llevará a la ventana de Ayuda y acerca de.



Figura 3.1: Ventana de Ayuda y acerca de

En esta ventana se muestran los créditos del sistema, y al dar clic en el botón identificado como Manual de Usuario se puede acceder a una copia electrónica en formato PDF de este Manual de Usuario.

## Capítulo 4

# Archivos de configuración

La primera pantalla que se muestra una vez que se ejecuta el archivo JAR, es la pantalla de selección de ruta para almacenar los archivos de configuración, los cuales incluyen toda la información de los alumnos.



Figura 4.1: Pantalla de Archivos de configuración

En la ubicación que sea elegida, el programa determinará si existe ya un archivo de configuración previo para poder cargarlo, en caso de que no encuentre ninguno, generará un nuevo archivo de configuración en esa ubicación.

El archivo de configuración se caracteriza por ser una carpeta con el nombre de *SeguimientoCONFIG*, dentro de la cual se encuentran dos archivos, uno denominado *bitacora.ser* y el otro llamado *configKey.dat*.

Estos dos archivos son imprescindibles para la correcta lectura de información, así como el nombre de la carpeta en la que se encuentran.

Es importante resaltar que si se quieren abrir archivos existentes, la ubicación proporcionada no debe apuntar a la carpeta *SeguimientoCONFIG*, si no a la carpeta en la cual se encuentra almacenada la carpeta *SeguimientoCONFIG*.

**Ejemplo 4.0.1** *Si se tienen archivos de configuración cargados previamente en la carpeta AlmacenamientoPrueba, como se observa la Fig. 4.2.*

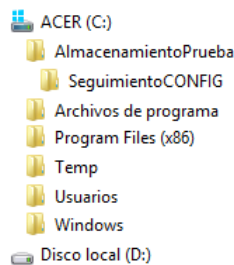


Figura 4.2: Pantalla de Archivos de Configuración

*Entonces, la ruta que deberá ser ingresada para que el sistema reconozca los archivos de configuración es la siguiente.*

***C://AlmacenamientoPrueba***

*Por el contrario, si se ingresa la siguiente ruta, causará que el programa no reconozca la información guardada y genere un nuevo archivo en blanco.*

***C://AlmacenamientoPrueba/SeguimientoCONFIG***

También es importante recalcar que la ubicación de estos archivos de configuración no necesariamente debe ser dentro del disco duro de la computadora que está ejecutando el programa. Estos archivos pueden estar en una ubicación de red disponible y visible para el usuario ejecutando el programa.

Una vez que ha sido escogida una ubicación, ésta queda guardada en los Registros del Sistema. Las próximas veces que el programa sea ejecutado, ya no pedirá que ingrese de nuevo esta información, ya que la leerá directamente de los Registros del Sistema, en el Editor del Registro del Sistema Operativo que se esté ocupando.

En caso de que se quiera modificar de nuevo la ubicación de los archivos de configuración que han sido determinados, se puede ingresar a esta pantalla mediante el botón de Configuración ubicado en la vista de Selección de Usuario.

## Capítulo 5

# Usuarios

En esta vista se requiere elegir de un nombre de usuario, para poder asignar una referencia a todos los cambios que se produzcan durante esa sesión. Debido a que ésta, como ya se ha dicho, será sólo una referencia, no se requiere de una contraseña para verificar la identidad.

The image shows a web application interface with a red background. At the top, the title 'Sistema de Seguimiento Académico' is displayed in white text, with a small blue circle containing a white question mark in the top right corner. Below the title, the instruction '1. Seleccione un usuario' is written in white. A large white rectangular input field is centered on the screen. Below this field are two buttons: 'Agregar' with blue text on a white background, and 'Eliminar' with black text on a white background. At the bottom of the interface, there are two more buttons: 'Configuración' with red text on a white background, and 'Continuar' with green text on a white background. A faint, stylized watermark of a classical building facade is visible in the background.

Figura 5.1: Pantalla de Selección de Usuario

La primera vez que se ejecute el programa en una computadora, si se

generaron archivos de configuración nuevos, no se encontrará ningún usuario en la lista de usuarios, por lo cual se debe dar clic en el botón Agregar, y escribir el nombre con el cual se identificará.

En caso de querer remover un usuario, se debe seleccionar al usuario a remover, y dar clic en el botón de Eliminar. Debe recordar que el nombre de usuario es sólo una referencia, por lo cual al eliminar un usuario no se borra ninguna información con respecto al avance académico de los alumnos.

Para poder acceder a la siguiente pantalla es imperativo escoger alguno de los usuarios de la lista y dar clic en Aceptar.

El botón de configuración llevará al usuario a la pantalla de Archivos de Configuración, para poder revisar la ruta en la cual se encuentra almacenada toda la información académica, y cambiarla en caso de que se requiera.

## Capítulo 6

# Actualización de datos

En esta pantalla se pueden ingresar los datos de información académica de los alumnos. Es indispensable ingresar al menos un archivo de calificaciones y avance de créditos para poder pasar a la siguiente pantalla o ingresar información de titulación.

Si se presiona el botón identificado con el texto Atrás, se cerrará esa ventana y se regresará a la ventana de selección de usuario.

### 6.1. Datos de avance académico

Para ingresar un archivo de calificaciones se requiere ingresar el año y semestre al que pertenecen en los campos correspondientes, utilizando 4 dígitos para el primero y un dígito para el segundo, y después dar clic en el botón de Aceptar, lo cual abrirá un asistente que le permitirá elegir un archivo de Excel, con extensión de Microsoft Excel 2010 XLSX.

El archivo elegido, para poder ser reconocido por el sistema, debe tener la siguiente configuración:

- No debe estar protegido por contraseña.
- Debe contener 6 columnas.
- Las columnas deben tener los siguientes nombres:
  1. CUENTA
  2. NOMBRE
  3. REGISTRO
  4. ULTIMO\_CURSADO

**Sistema de Seguimiento Académico**

**2. Actualice calificaciones**

| Año                 | Sem | Agregador por | Agregado en |
|---------------------|-----|---------------|-------------|
| Tabla sin contenido |     |               |             |

Año Sem **Agregar** Eliminar

**3. Actualice datos de titulación**

Última Actualización

**Actualizar** **Quitar última**

**4. Actualice datos de carrera**

# total de semestres

**Atrás** **Aceptar**

Figura 6.1: Pantalla de Ingreso de Datos

5. PROMEDIO

6. AVANCE

- El título de las columnas sólo debe ocupar una fila, la número 1.
- La columna de CUENTA siempre debe de contener la información en un formato de 9 dígitos.
- Las columnas de REGISTRO y ULTIMO\_CURSADO deben contener la información del ciclo en un formato e 5 dígitos juntos, sin espacios o guiones. Es decir, el siguiente formato *aaaas*, las letras *a* representando los cuatro dígitos del año, y la letra *s* representando el número de semestre, es decir 1 o 2.
- Las columnas de CUENTA, NOMBRE, REGISTRO y ULTIMO\_CURSADO deben estar en celdas con formato de texto.

- Las columnas de PROMEDIO y AVANCE deben estar en celdas con formato numérico.

En la Fig. 6.2 se puede observar un ejemplo del formato y estructura que debe llevar el archivo de información de calificaciones.

|   | A      | B      | C        | D              | E           | F         |
|---|--------|--------|----------|----------------|-------------|-----------|
| 1 | CUENTA | NOMBRE | REGISTRO | ULTIMO_CURSADO | PROMEDIO    | AVANCE    |
| 2 |        |        | 20002    | 20071          | 8.980392456 | 100       |
| 3 |        |        | 20002    | 20052          | 8.72222328  | 100       |
| 4 |        |        | 20011    | 20061          | 8.259259224 | 100       |
| 5 |        |        | 20011    | 20061          | 9.403845787 | 104.02684 |
| 6 |        |        | 20011    | 20052          | 8.814814568 | 100       |
| 7 |        |        | 20011    | 20072          | 8.462963104 | 107.15884 |

Figura 6.2: Ejemplo de Estructura de Archivo de Calificaciones

Una vez que se ha seleccionado un archivo correcto, el programa transferirá toda la información contenida en el archivo de Excel a los archivos de configuración internos del programa, los cuales se encuentran almacenados en la ruta proporcionada en la primer pantalla del sistema, por lo cual no es necesario mantener el archivo de Excel, debido a que los datos ya han sido copiados al sistema.

En caso de que se quiera eliminar la información de cierto ciclo, se debe escribir primero el año y semestre de los datos que se quieren remover y después dar clic en el botón Eliminar. Al realizar esta acción sólo se borrarán los datos de ese semestre, cualquier información sobre otros semestres permanecerá almacenada.

Se recomienda actualizar los datos de calificaciones y avance en créditos una vez que se haya terminado el ciclo escolar, y que no se puedan realizar más cambios a las calificaciones de los alumnos, preferiblemente al inicio del siguiente ciclo, para poder asegurar que los datos ingresados al sistema son los correctos y no cambiarán posteriormente.

## 6.2. Datos de titulación

Para actualizar la información de titulación, no es necesario ingresar ningún ciclo en específico, ya que se puede actualizar la lista de titulados en cualquier momento. Para ingresar nuevos datos lo único que se necesita es dar clic en el botón Actualizar y un asistente se mostrará para poder seleccionar un archivo de Excel, con extensión de Microsoft Excel 2010 XLSX, el cual debe tener la siguiente estructura interna:

- No debe estar protegido por contraseña.
- Debe contener al menos 6 columnas.
- Las columnas con las que debe contar son las siguientes:
  1. CUENTA
  2. MODALIDAD
  3. TÍTULO DEL TRABAJO ESCRITO
  4. FECHA DE TITULACIÓN
  5. DIRECTOR
  6. RESULTADO
- El título de las columnas sólo debe ocupar una fila.
- La columna de FECHA DE TITULACIÓN debe estar en celdas en formato de fecha, así como la siguiente estructura: *dd/mm/aaaa*. Las letras *d* representando el número el día del mes a dos dígitos, las letras *m* representando el número del mes a dos dígitos y las letras *a* representando el año en cuatro dígitos.

En la Fig. 6.3 se puede observar un ejemplo del forma y estructura que puede llevar un archivo de titulación válido.

|   | A                      | P                | R                                | U                          | AL                  | AM                     |
|---|------------------------|------------------|----------------------------------|----------------------------|---------------------|------------------------|
| 4 | <b>CUENTA</b>          | <b>MODALIDAD</b> | <b>TÍTULO DE TRABAJO ESCRITO</b> | <b>FECHA DE TITULACIÓN</b> | <b>DIRECTOR</b>     | <b>RESULTADO</b>       |
| 5 | TESIS O TESIS Y EXAMEN | PROFESIONAL      | TOTALIDAD DE CRÉDITOS Y          | ALTO NIVEL ACADÉMICO       | TITULACIÓN MEDIANTE | TESIS O TESIS Y EXAMEN |
| 6 | PROFESIONAL            | PROFESIONAL      | PROFESIONAL                      | PROFESIONAL                | PROFESIONAL         | PROFESIONAL            |
| 7 | PROFESIONAL            | PROFESIONAL      | PROFESIONAL                      | PROFESIONAL                | PROFESIONAL         | PROFESIONAL            |
| 8 | PROFESIONAL            | PROFESIONAL      | PROFESIONAL                      | PROFESIONAL                | PROFESIONAL         | PROFESIONAL            |

Figura 6.3: Ejemplo de Estructura de Archivo de Titulación

Cuando se ha ingresado exitosamente el archivo de datos de titulación, estos son copiados a los archivos de configuración del sistema, por lo cual no es necesario mantener el archivo de Excel original. Es importante notar que el archivo de titulación no tiene porqué incluir todos los registros anteriores que ya han sido ingresado al sistema. El sistema va acumulando la información, entonces no es necesario volver a ingresar los datos anteriores, a pesar de que

no afecta de ninguna manera que estos datos sean ingresados nuevamente, no se sobrescribirán. Se recomienda que se ingresen todos los datos posibles, aunque estos incluyan información vieja.

Al dar clic en el botón de Quitar Última, se eliminarán sólo los nuevos datos que fueron ingresados al sistema por la última actualización. En el campo de texto respectivo se muestra en qué fecha fue actualizado el sistema, y al accionar el botón Quitar Última, éste mostrará la actualización previa inmediata a la última.

En el caso de los datos de titulación, se recomienda realizar actualizaciones cada que sea posible, es decir, no se tiene que esperar a que acabe un ciclo escolar para ingresar nuevos datos, debido a que estos van cambiando constantemente independientemente del transcurso de algún ciclo escolar, a diferencia de los datos de calificaciones y avance en créditos, los cuales sólo son modificados al término de un ciclo escolar.

### **6.3. Datos de carrera**

Para poder calcular el atraso académico de los estudiantes es necesario saber de cuántos semestres se compone el plan de estudios de la carrera, lo cual se debe indicar en el campo que se encuentra en esta sección. Basta con dar clic en la flecha que apunta hacia abajo en este campo para ver un menú desplegable de todas las posibles opciones de número de semestres en el plan de estudios, y seleccionar la correcta.

## Capítulo 7

# Pantalla principal

Al ingresar a la pantalla principal se pueden observar en la parte superior varios detalles.



Figura 7.1: Sección superior de la Pantalla Principal

En la esquina superior izquierda se encuentra un botón de Atrás, con el cual se puede regresar a la pantalla de ingreso de datos.

En la esquina superior derecha, a un lado del botón de Ayuda, se encuentra un botón de Actualizar. La función de este botón es volver a cargar los datos contenidos en los archivos de configuración, en caso de que los datos de configuración se encuentren en una red y hayan sido modificados simultáneamente por otro usuario.

Finalmente se encuentra una cinta de pestañas que llevan a las tres diferentes secciones del programa: Último Semestre, Historia Académica y Análisis. Éstas serán explicadas a fondo en las siguientes secciones del capítulo. Por defecto, el sistema entra a la primer pestaña que se denomina Último Semestre.

### 7.1. Último semestre

En esta sección de la pantalla principal se observan los datos al último semestre de los alumnos, es decir, su promedio y avance actuales, así como el último semestre en el cual estuvieron activos.

| Cuenta               | Nombre                  | Registro<br>Año Sem | Último Cursado<br>Año Sem | Promedio | Avance | Status   | Comentario |
|----------------------|-------------------------|---------------------|---------------------------|----------|--------|----------|------------|
| 00000000000000000000 | ALFONSO ALBERTO ALFONSO | 2000 2              | 2007 1                    | 8.98     | 100.00 | Titulado |            |
| 00000000000000000000 | ALFONSO ALBERTO ALFONSO | 2000 2              | 2005 2                    | 8.72     | 100.00 | Titulado |            |
| 00000000000000000000 | ALFONSO ALBERTO ALFONSO | 2001 1              | 2006 1                    | 8.26     | 100.00 | Titulado |            |
| 00000000000000000000 | ALFONSO ALBERTO ALFONSO | 2001 1              | 2006 1                    | 9.40     | 104.03 | Titulado |            |
| 00000000000000000000 | ALFONSO ALBERTO ALFONSO | 2001 1              | 2005 2                    | 8.81     | 100.00 | Titulado |            |
| 00000000000000000000 | ALFONSO ALBERTO ALFONSO | 2001 1              | 2007 2                    | 8.46     | 107.16 | Titulado |            |
| 00000000000000000000 | ALFONSO ALBERTO ALFONSO | 2001 1              | 2006 1                    | 8.30     | 100.00 | Titulado |            |
| 00000000000000000000 | ALFONSO ALBERTO ALFONSO | 2001 1              | 2006 1                    | 8.43     | 100.00 | Titulado |            |
| 00000000000000000000 | ALFONSO ALBERTO ALFONSO | 2001 1              | 2005 2                    | 9.02     | 100.00 | Titulado |            |
| 00000000000000000000 | ALFONSO ALBERTO ALFONSO | 2001 2              | 2006 1                    | 9.06     | 100.00 | Titulado |            |
| 00000000000000000000 | ALFONSO ALBERTO ALFONSO | 2001 2              | 2008 1                    | 8.04     | 100.95 | Pasante  |            |
| 00000000000000000000 | ALFONSO ALBERTO ALFONSO | 2001 2              | 2006 2                    | 8.38     | 100.00 | Titulado |            |
| 00000000000000000000 | ALFONSO ALBERTO ALFONSO | 2001 2              | 2007 1                    | 8.11     | 100.00 | Titulado |            |
| 00000000000000000000 | ALFONSO ALBERTO ALFONSO | 2001 2              | 2006 1                    | 9.59     | 100.45 | Titulado |            |
| 00000000000000000000 | ALFONSO ALBERTO ALFONSO | 2001 2              | 2006 1                    | 9.07     | 100.00 | Titulado |            |

Figura 7.2: Pantalla principal en sección de Último Semestre

Una de las funcionalidades más útiles de ésta pantalla es poder asignarle a cada alumno un status basado en su rendimiento académico. Este status cae en alguna de las seis categorías existentes:

- **Titulado**, significa que el estudiante ha completado el 100 % de créditos y cumplió con el trámite de titulación.
- **Pasante**, se refiere a un estudiante que ha completado el 100 % de créditos pero no ha cumplido con el trámite de titulación.
- **Ok**, con el cual se identifica a un alumno que va al corriente en su avance académico, es decir, tiene el número de créditos correspondientes al semestre que se encuentra cursando.
- **Atrasado**, representa a un alumno que no tiene el número de créditos correspondiente a su semestre académico.
- **Art. 22**, significa que dicho alumno ha rebasado el número de semestres permitidos para poder tener el derecho a inscripción regular a sus materias. Dicho límite es el 50 % de la duración del plan de estudios respectivo adicionado a la duración regular, 14 o 15 semestres para todas las carreras de la DIMEI. Esto viene expresado en el artículo 22 de la Legislación Universitaria.

- **Art. 24**, se refiere a un estudiante que ha rebasado el tiempo límite para el cumplimiento de la totalidad de créditos, el cual es el doble de tiempo establecido en el plan de estudios correspondiente. Después del cual, siendo estrictos, se debería realizar baja de estudios por parte la Universidad, como está estipulado en el artículo 24 de la Legislación Universitaria. Dicho proceso concierne a la administración de la Universidad, más no a la Facultad de Ingeniería, y se ha observado que es poco usual que se proceda con la baja del alumno, por lo cual no es recomendable descartar dichos alumnos.

Así mismo, mediante el panel de filtros de esta sección, se pueden realizar todo tipo de filtros a la tabla mostrada. Estos filtros son acumulativos, es decir, que se pueden aplicar varios filtros al mismo tiempo dentro de la misma tabla para poder obtener sólo los resultados que se buscan. Cabe destacar que todos los rangos ingresados en los filtros son inclusivos.

|          |               |           |               |          |         |                |
|----------|---------------|-----------|---------------|----------|---------|----------------|
| Avance   | menor - mayor | Registro  | menor - mayor | Cuenta   | ▼       | Borrar filtros |
| Promedio | menor - mayor | Último c. | menor - mayor | Titulado | Pasante | Ok             |
|          |               |           |               | Atrasado | Art. 22 | Art. 24        |

Figura 7.3: Panel de filtros en sección Último Semestre

Los filtros de Avance y Promedio se deben escribir utilizando una notación de punto decimal, más no coma decimal. Por su parte, los filtros de Registro y Último Cursado, se deben escribir con un formato *aaaas*, las letras *a* representando los 4 dígitos del año, y la letra *s* representando el semestre, los 5 dígitos juntos, sin espacios o guiones.

**Ejemplo 7.1.1** *Supongamos que se quiere filtrar a los alumnos que ingresaron en el ciclo 2010-1 o mayor, entonces se debe ingresar en el filtro de Registro, dentro del campo indicado como Menor, el siguiente conjunto de 5 dígitos:*

**20101**

Si se quiere empezar una búsqueda nueva y eliminar todos los campos ingresados o modificados en los filtros basta con dar clic en el botón Borrar Filtros para empezar de cero con la lista completa de alumnos.

También se cuenta con la opción de ordenar la lista en orden ascendente y descendente en cualquiera de las columnas provistas. Para activar o desactivar dicha funcionalidad basta con dar uno o dos clics en el nombre de

la columna, dependiendo el orden que se desee. Para ordenar la lista con múltiples criterios de orden con diferentes columnas es necesario mantener presionada la tecla SHIFT mientras se hace clic en las columnas que se quiere ordenar.

Entrando en detalle en la información de la tabla, se tienen las columnas de Cuenta y Nombre, en las cuales al dar doble clic se puede seleccionar el texto completo automáticamente, para fácil copiado de estos. En el caso de la columna de Status, se tiene que el texto dentro de esta columna variará dependiendo del mismo, y en caso de que el estudiante tenga un status de titulado, basta con dejar el mouse en el texto *Titulado* por unos segundos para ver más información sobre su forma de titulación.

Por último, en la columna de Comentario, se ha implementado la funcionalidad de escribir un comentario sobre el progreso académico del alumno. Para ingresar un nuevo comentario es necesario dar doble clic en el espacio de esa columna y comenzar a escribir, cuando se haya terminado de editar el comentario, es imprescindible presionar la tecla ENTER, para guardar los cambios.

Para eliminar un comentario se sigue el mismo procedimiento, pero al editar el comentario, se borra todo el texto utilizando la tecla de ATRÁS o de SUPRIMIR, sin olvidar presionar la tecla ENTER al finalizar los cambios para poder guardar la información actualizada.

Si se quiere saber más información sobre algún comentario, como puede ser el autor de dicho comentario o la fecha en la que fue editado, se puede dejar el mouse sobre dicho comentario por unos segundos para que aparezca un mensaje emergente indicando dicha información.

Adicionalmente, una última funcionalidad que fue incluida dentro de esta sección fue la posibilidad de exportar el contenido actual de la tabla, a un archivo de texto o un archivo de Excel, mediante el botón de Exportar en la esquina inferior izquierda. Al apretar el botón, aparecerá un asistente para poder guardar el archivo en la ruta deseada.

## 7.2. Historia académica

En esta segunda sección de la pantalla principal se muestra la historia académica de los últimos cinco semestres que han sido ingresados al sistema de todos los alumnos.

Como se muestra en la Fig. 7.4, la sección de filtro ha sido reducida a un campo, en el cual se puede buscar por número de cuenta o nombre.

Aparte, se observa que al tener más de un semestre registrado en el



### 7.3. Análisis

Finalmente, en la sección de Análisis de la pantalla principal, se pueden observar varias gráficas que representan un resumen del estado actual del alumnado en general.

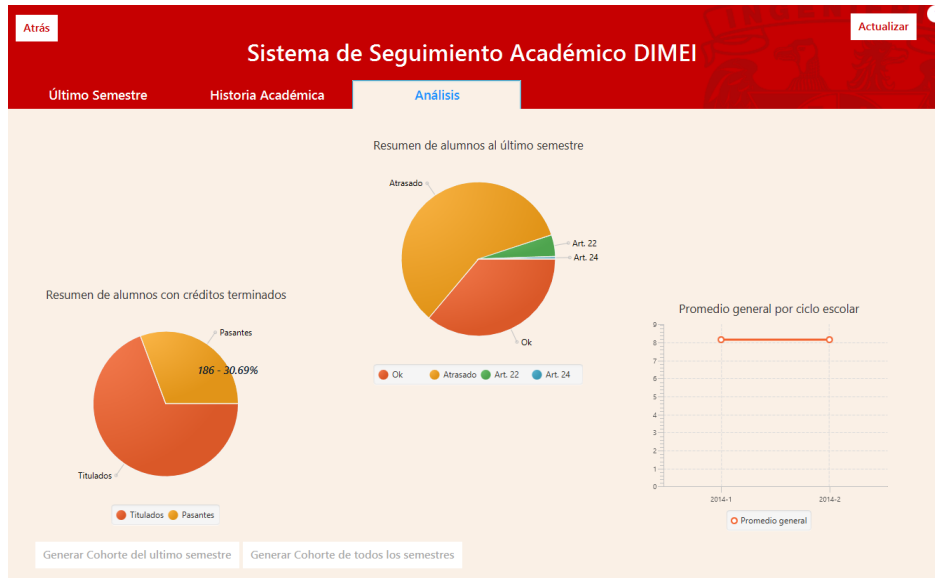


Figura 7.6: Pantalla Principal en sección de Análisis

Para obtener más información sobre los datos en los cuales se basan las gráficas mostradas se requiere dar clic derecho sobre la sección de la gráfica que se quiera revisar, y se mostrará información sobre la cantidad representada así como el porcentaje que simboliza.

En la parte inferior de la sección se encuentran dos botones con funciones muy importantes. Dichos botones generan con la información al último semestre el formato de Cohortes que es requerido por la facultad. Para generarlos basta con dar clic en alguno de los dos, acto seguido se mostrará un asistente para indicar la ruta en la cual se requiere guardar dicho documento. Los documentos de Cohortes serán generados en formato Microsoft Excel 2010 con extensión XLSX.