

Desarrollo de módulo para el informe SAii que la UM comparte con FIMPES para la acreditación Universitaria

Oliver Alejandro Hernández Rueda De León
Facultad de Ingeniería y Tecnología
Universidad de Morelos

El proyecto que en este escrito se describe fue realizado con la finalidad de desarrollar un módulo que estará alojado dentro del Sistema Académico Virtual de la Universidad de Morelos (UM). Se utilizará para recabar información estadística acerca de la UM que la Dirección de Efectividad Institucional debe reportar a la Federación de Instituciones Mexicanas Particulares de Educación Superior (FIMPES). Se logró que la aplicación coincida fielmente con el formato que tiene como requerimiento la FIMPES para que las instituciones reporten su data estadística que sirve a la FIMPES para desarrollar y sugerir estrategias de innovación y mejora de procesos críticos en el área educativa. Los datos que las coordinaciones suministran en este módulo, servirán además para otros procesos que la universidad realiza para su operación estratégica.

Keywords: bases de datos relacionales, arquitectura de software, patrón arquitectónico, Modelo-Vista-Controlador.

Introducción

La Dirección de Efectividad Institucional responde a requerimientos internos y externos que consisten en la entrega de datos estadísticos que servirán en primera instancia para procesos relacionados con certificaciones y acreditaciones.

Una de las tareas de las que actualmente es responsable la Dirección de Efectividad Institucional, consiste en reunir información relacionada con los estudiantes, mayormente consistente en datos demográficos para generar un reporte que anual y sistemáticamente debe ser entregada a la FIMPES. El sistema que alimentan las instituciones para la FIMPES es llamado Sistema Analítico de Información de Instituciones (SAii).

Por lo anteriormente mencionado se hizo necesario desarrollar este módulo que servirá para agilizar la recolección de los datos y facilitar el llenado de los formatos de manera automática, solicitando el apoyo de los coordinadores en la recolección de una parte de los datos y obteniendo los faltantes mediante la base de datos de manera automatizada.

Antecedentes

La forma tradicional como se ha llevado a cabo esta tarea es de la siguiente manera: la FIMPES hace llegar a la institución el formato en un archivo de Excel, y el personal que labora en efectividad institucional debe llenar los datos con captura manual. Una parte considerable de la información se encuentra alojada en las bases de datos que genera el sistema virtual de la UM, pero las celdas no coinciden con el formato de Excel que envía la FIMPES así que el llenado debe ser con captura manual. Otros datos son conseguidos en las coordinaciones de las facultades de la UM.

Esta forma de llenado de los formatos ha llegado a convertirse en una carga tediosa y agobiante tanto para los coordinadores, como para el personal que labora en efectividad institucional, y para los desarrolladores y custodios del sistema computacional de la institución, pues en ocasiones, el requerimiento de los datos llega en momentos en que la ocupación es extenuante la cantidad de datos es extremadamente abundante y detallada.

El proceso se ha llevado a cabo por años, de la siguiente manera: al llegar la solicitud de los datos por parte del SAii al departamento de efectividad institucional en formato de Excel, el formulario es segmentado para poder proceder a la recolección de estos datos. Una vez que el formato es repartido a cada una de las carreras, se responsabiliza al coordinador de llenar los datos. El tipo de datos requeridos van desde muy generales hasta muy específicos. Los datos que ya están guardados en las bases de datos del sistema académico de la institución, son solicitados por el coordinador a la dirección de sistemas, para posteriormente vaciarlos de manera manual en el formato Excel.

Mas como ya se ha mencionado, no todos los datos están guardados en el sistema académico, por lo cual algunos datos tienen que ser investigados y provistos por parte del coordinador para poder devolver el formato debidamente llenado a la Dirección de Efectividad Institucional. Cuando todos los formatos llenos llegan nuevamente a la dirección, el personal designado se da a la tarea de unir cada uno de estos formatos aislados y lo convierten en un reporte general a nivel institucional para posteriormente entregarlo a la FIMPES.

Al poner a funcionar y al utilizar la aplicación que se describe en este trabajo, se logrará otorgar una solución tecnológica óptima, ya que se respetará y aprovechará el tiempo de los involucrados. La información que se encuentra en las bases de datos que genera el sistema será vertida de manera automática en los archivos de Excel que deben ser entregados a la FIMPES. Así mismo, esta operación se realizará de forma automatizada, con el apoyo de un módulo en el sistema académico que consultará los datos alojados en las bases de datos del sistema y verterá los datos no existentes en nuevas tablas que serán creadas con ese propósito.

Definición del problema

Para determinar el origen del problema, existen varios instrumentos que pueden ser aplicados durante el momento del análisis, el instrumento que se utilizó para los propósitos de este proyecto es el denominado: Instrumento 5, el cual se muestra en la Tabla 1.

El instrumento 5 consiste en una técnica para realizar preguntas iterativas, usadas para explorar las relaciones de causa y efecto subyacentes a un

problema particular. El objetivo principal de la técnica es determinar la causa raíz de un defecto o problema repitiendo la pregunta "¿Por qué?".

Cada respuesta forma la base de la siguiente pregunta. El nombre "5" se deriva de la observación empírica en el número de iteraciones típicamente requeridas para resolver el problema[1].

TABLA 1
ANÁLISIS DE LOS 5 PORQUÉS

¿Por qué? 1	¿Por qué es tardado y tedioso el proceso? Por qué el proceso se realiza a mano
¿Por qué? 2	¿Por qué se realiza el proceso a mano? Por qué la forma en la cual se capturan los datos es mediante un formato Excel
¿Por qué? 3	¿Por qué se realiza la captura de datos mediante un formato Excel? Porque éste es el único medio provisto para capturar estos datos
¿Por qué? 4	¿Por qué es el único medio provisto para capturar datos? Porque no se cuenta con una forma óptima de captura de datos
¿Por qué? 5	¿Por qué no hay otra forma de captura de estos datos? Porque no se ha desarrollado
Conclusión	El proceso es tardado y tedioso debido a la forma en que se capturan los datos a mano en un formato Excel, porque no se cuenta con un método más eficiente de consulta y captura de estos datos por no haber desarrollado anteriormente una aplicación apropiada.

Justificación

El llenado de datos estadísticos en un formato establecido por el organismo FIMPES, requiere de un módulo dentro del sistema académico de la institución que provea al departamento de efectividad y a los coordinadores de las carreras una manera sencilla y viable de recolectar los datos. Dicho módulo es una herramienta muy necesaria, que puede ayudar a reducir y optimizar el tiempo y el esfuerzo que se emplea para la recolección de datos en un archivo de Excel para llenado a mano. Esta herramienta, desarrollada en un ambiente conocido por todas las partes involucradas es muy conveniente, ya que, por su sencillez, el tiempo requerido para aprender a usarla puede ser bastante corto, pues resulta muy fácil de usar y muy sencillo de entender.

Objetivos

El objetivo general de este proyecto es el desarrollo de un módulo computacional, para facilitar la tarea de coleccionar información académica y estadística relacionada con la población estudiantil inscrita en la Universidad de Morelos. Siendo que la información requerida debe reportarse al organismo FIMPES de manera anual y sistemática, y la responsabilidad recae sobre la Dirección de Efectividad Institucional, se considera que el desarrollo de este módulo, podrá impactar de manera favorable el rendimiento de la Dirección de Efectividad Institucional al lograr que esta tarea resulte menos difícil y menos tardada.

Objetivos específicos

1. Desarrollar un módulo computacional hospedado en el sistema académico virtual que utiliza la Universidad de Morelos para el control escolar y el resguardo de datos de la comunidad estudiantil que se inscribe en la institución. Este módulo incluirá una interfaz que permitirá a la Dirección de Efectividad Institucional capturar la data en los formatos establecidos por la FIMPES en su Sistema Analítico de Información de Instituciones (SAII).
2. Desarrollar con las mismas características del módulo, una segunda interfaz que podrá ser

utilizado por los coordinadores, o quien éstos designen para la recolección de datos.

3. Desarrollar e implementar una tercera interfaz, utilizando la arquitectura multicapa para software, con la finalidad de generar los reportes de los planes académicos filtrados por facultad, los cuales permitirán a los coordinadores trabajar de manera más fácil y optimizada.

Limitaciones

1. Limitaciones de tiempo: El desarrollo de este proyecto contó con la limitante de tener que ser realizado en las instalaciones del Departamento de Sistemas de la UM. La razón por la que debió ser desarrollado dentro de las instalaciones, por la sensibilidad del acceso a los datos hospedados en los servidores institucionales, obligó a trabajar en algunas secciones del proyecto únicamente en los tiempos en que era factible para el supervisor que custodia este activo de la institución. De manera que tanto las pruebas en el desarrollo del módulo, como las pruebas de implementación, estuvieron limitadas por los tiempos de disponibilidad del supervisor, y de quienes serán usuarios del módulo.
2. Limitaciones de equipo: Similar a lo ocurrido con el uso del tiempo del supervisor que custodia los sistemas, como de los usuarios que utilizarán el módulo, el desarrollo de la herramienta estuvo sujeto al uso de equipo disponible en las instalaciones del departamento de sistemas. Ya que no está permitido acceder al software del sistema académico desde otros IP, fuera de la institución, o con equipo personal. En ocasiones los equipos destinados para el desarrollo estaban ocupados por otros desarrolladores, y fue necesario esperar a que hubiera disponibilidad de equipo para continuar con el desarrollo de este proyecto.

Delimitaciones

1. El proyecto se ha delimitado a desarrollar un módulo en el Framework de Spring basado en el patrón arquitectónico de software MVC (modelo, vista, controlador) y la arquitectura multicapa, para la gestión del registro y

obtención de datos. Los datos serán obtenidos únicamente de la base de datos del sistema académico institucional con la única finalidad de generar los reportes necesarios para desarrollar planes de mejora institucional. Los desarrollos se llevaron a cabo mediante los lenguajes de programación de JavaScript, HTML5, y bases de datos de Oracle SQL. Aunque ciertamente existen diversos lenguajes de programación que pudieron ser utilizados para desarrollar este módulo, por ejemplo: C-Sharp, C++, y otros más, para los fines de este proyecto se utilizó el lenguaje de Java y HTML5, debido a que es ésta la combinación de lenguajes con la que se desarrolló y se opera el sistema académico de la institución. Esto debió ser así, por la interacción estrecha que existirá entre la base de datos estudiantiles de la institución alimentada por el sistema virtual y el sistema donde se desarrolla este módulo.

2. Se restringe el acceso a estos módulos del sistema solo para los miembros de la Dirección de Efectividad Institucional en la manipulación de datos y la ampliación o añadidura de períodos, y a los coordinadores de carrera, o a quien ellos designen, para hacerse cargo del llenado de información en los formatos provistos.

Hipótesis

Este proyecto consiste en desarrollar aplicaciones tecnológicas que consistirán en los módulos necesarios para generar el reporte denominado SAii. Este reporte es requerido por la Federación de Instituciones Mexicanas Particulares de Educación Superior (FIMPES) a todas las instituciones particulares de educación superior que están afiliadas a este organismo, entre las cuales figura la Universidad de Montemorelos, y anualmente debe entregarse como parte de los compromisos institucionales a fin de mantener la afiliación de la UM al mencionado organismo.

Esto se ha logrado implementando un patrón arquitectónico MVC para la construcción del módulo. Por lo anteriormente expuesto, no se requiere el planteamiento o la comprobación de hipótesis estadísticas.

Marco Teórico

Arquitectura de Software

La arquitectura de software es la estructura o la forma que adopta un sistema(software), esta se elige en etapas tempranas del desarrollo del sistema, ya que sirve como guía para desarrollarlo. Esta es muy importante porque mediante ella, se pueden cumplir los requisitos o se puede fallar en cumplirlos si no está bien definida. La arquitectura de software es la base de donde se parte para desarrollar el sistema, ya que esta misma define todos los elementos dentro del sistema y como deben de estar conectados entre sí y que información tendrá cada elemento [2]. Un ejemplo de arquitectura de software, es la Multicapa, la cual es altamente utilizada gracias a su manera de trabajar, ya que separa los procesos de un software y los ordena, esto hace que se pueda interactuar de manera fácil y entendible con el código.

Arquitectura Multicapa

Una arquitectura multicapa es un conjunto ordenado de elementos, los cuales están contruidos de acuerdo con lo que tenga la capa por debajo, y son la base de lo que se implementa en las capas superiores. Es decir, en un proyecto como este, la primera capa es la que conecta el sistema con la base de datos, y las siguientes capas utilizan esa información y los objetos que se crearon en la primera capa.

Un ejemplo de esto, es el patrón arquitectónico MVC (Modelo, vista, controlador), el cual consta de 3 capas separadas, las cuales interactúan y utilizan los métodos y objetos que se crearon en las capas posteriores, como es el caso del controlador, el cual utiliza los métodos y objetos que pertenecen al modelo. Al tener separados los elementos, es mucho más fácil dar un mantenimiento al sistema en caso de que se encuentren errores o si se requiere hacer una modificación [3].

Modelo-vista-controlador

Modelo-vista-controlador (MVC) es un patrón de arquitectura de software, el cual separa la interfaz de usuario de la gestión de los datos, y consta de tres capas distintas que son, el modelo, la vista y el controlador [4]. La manera en que trabaja un software, el cual está desarrollado a partir de este patrón arquitectónico, es la siguiente:

- **Modelo:** Es la capa que se encarga de la conexión con la base de datos, ya sea para obtener datos de la tabla con la que se está trabajando, como para guardar los datos que han pasado por las otras 2 capas, en alguna tabla predeterminada para guardar esos datos.
- **Controlador:** Es la capa que se encarga de repartir las tareas del sistema, por ejemplo, si el usuario hace una solicitud de datos o si ingresa información al sistema, el controlador se encarga de pedir esos datos al modelo o de mandar esos datos para que el modelo los guarde en la base de datos.
- **Vista:** Es la capa que funciona como la interfaz de usuario, es decir, esta capa muestra la información final al usuario de manera amigable y entendible, y aunque en esta capa también se pueden hacer algunos métodos y funciones, su uso general es como interfaz de usuario.

Se puede ver de manera gráfica como trabaja este patrón arquitectónico en la Figura 1.

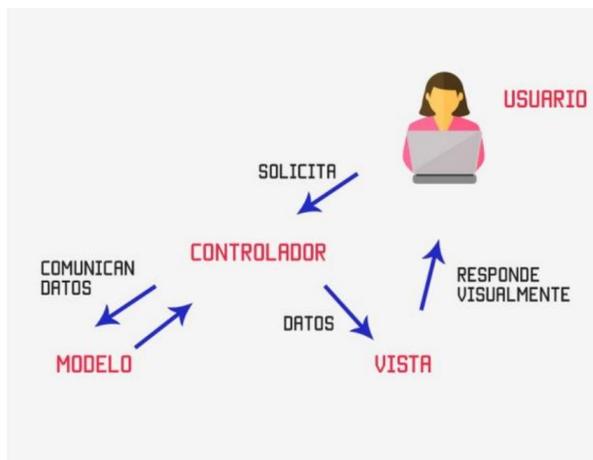


Figura 1. Funcionamiento del MVC. Fuente: Hernández U. 2015

Contenido

Requerimientos del módulo

- Diseñar las tablas necesarias para los nuevos datos en la base de datos.
- Desarrollo de los modelos que gestionan los datos obtenidos de las tablas.
- Desarrollo de los controladores que transportan la información de una capa a otra.
- Diseño de las vistas o la interfaz donde los usuarios podrán guardar y consultar la

información.

Tecnologías utilizadas

Java

Java es un lenguaje de programación, sin el cual muchas aplicaciones y dispositivos de la actualidad no podrían funcionar. Este lenguaje está presente en dispositivos portátiles, centros de datos, consolas de videojuegos, supercomputadoras, dispositivos móviles e internet, esto debido a la rapidez, seguridad y fiabilidad en su funcionamiento. Java ha sido probado, ajustado, ampliado y probado por toda una comunidad de desarrolladores, arquitectos de aplicaciones y entusiastas de Java. Java está diseñado para permitir el desarrollo de aplicaciones portátiles de elevado rendimiento para el más amplio rango de plataformas informáticas posible. Este lenguaje de programación es muy parecido al lenguaje C++, pero reduce el número de líneas de código necesarias para hacer una tarea comparado con este mismo. También este lenguaje tiene una de las comunidades de programadores más grandes del mundo, al rondar en los 9 millones, y al mismo tiempo es una comunidad muy activa, por lo tanto, este lenguaje se actualiza constantemente para implementar las mejores herramientas que su comunidad va desarrollando. El desarrollo de aplicaciones en este lenguaje es de los mejores remunerados en la actualidad [5]. El lenguaje de programación Java es el segundo más utilizado por los programadores según el índice TIOBE, en este año 2020, lo cual indica su gran aceptación y utilidad que tiene este lenguaje de programación [6].

Spring Tool Suite

Spring Tool Suite (STS) es un conjunto de herramientas para crear aplicaciones Spring. Se puede instalar como complemento de una instalación existente de Eclipse JEE o se puede instalar de forma independiente. La versión independiente de STS también se empaqueta con Eclipse EE, por lo que todas las características de Eclipse para el desarrollo de Java EE también están disponibles en STS. Este Framework trabaja con el patrón arquitectónico MVC [7].

Spring JdbcTemplate

Spring JdbcTemplate es un mecanismo poderoso para conectarse a la base de datos y ejecutar consultas SQL. Utiliza internamente la

API de JDBC, pero elimina muchos problemas de estamisma [8].

Oracle SQL Developer

Oracle SQL Developer es un entorno de desarrollo integrado y gratuito que simplifica el desarrollo y la gestión de Oracle Database en implementaciones tradicionales y en la nube. SQL Developer ofrece un completo desarrollo integral para sus aplicaciones PL/SQL, una hoja de trabajo para ejecutar consultas y scripts, una consola DBA para gestionar la base de datos, una interfaz de informe, una solución completa de modelado de datos y una plataforma de migración para mover sus bases de datos de terceros a Oracle [9].

2.3.6 Apache Subversion

Apache Subversion (abreviado frecuentemente como SVN) es una herramienta de control de versiones open source basada en un repositorio cuyo funcionamiento se asemeja enormemente al de un sistema de ficheros. Es software libre bajo una licencia de tipo Apache/BSD. Subversión puede acceder al repositorio a través de redes, lo que le permite ser usado por personas que se encuentran en distintas computadoras. A cierto nivel, la posibilidad de que varias personas puedan modificar y administrar el mismo conjunto de datos desde sus respectivas ubicaciones fomenta la colaboración. Este se integra mediante Spring Tools [10].

Metodología

Descripción del análisis y diseño del Software

Recolección de los requerimientos

La primera reunión que se realizó con el departamento de efectividad, fue para definir en qué plataforma se haría el proyecto, ya que originalmente el proyecto se desarrollaría como una pestaña dentro de la página web del departamento, sin embargo, de esa manera no había acceso a la base de datos del sistema, ya que la página web del departamento de efectividad tiene su propia base de datos.

También se platicó de qué manera se recolectarían los datos y de la interpretación de los indicadores, es decir, que información pide cada indicador de la lista de indicadores que envía FIMPES mediante el SAii.

Análisis de los requerimientos

Teniendo el problema ya definido, se procedió a analizar la manera para gestionar los datos, y se definieron los datos que se podían obtener directo de la base de datos de la Universidad. Viendo que había datos faltantes, surgió un nuevo requerimiento, la creación de una nueva tabla en la cual pudieran ser capturados tales datos en el sistema. Teniendo en cuenta que estos datos son obtenidos tanto de la base de datos del sistema institucional como de los coordinadores de las facultades, se determinó que las interfaces debían estar filtradas entre usuario (Coordinador) y Administrador (Departamento de Efectividad Institucional). En las cuales el Administrador pudiera dar de alta el periodo para la obtención de datos del sistema y posteriormente poder consultarlos, también poder notificar a los usuarios el periodo en el cual se deben llenar estos datos, mientras que el usuario podría llenar la información de los planes correspondientes a su área de coordinación.

Autorización para el desarrollo del software

Una vez que estuvieron definidos los requerimientos y que se entendieron los indicadores, el ingeniero Ery torres, dio la autorización para comenzar a desarrollar el módulo dentro del sistema académico. Para comenzar con el desarrollo del proyecto, se realizó una pequeña capacitación, para aprender a usar el Framework de Spring, incluida una explicación sobre cómo trabaja el patrón MVC mediante Spring y también como se conecta con las bases de datos de la universidad.

El desarrollo del módulo

Para el desarrollo de este módulo, se utilizó otro módulo ya existente que contiene los periodos académicos, esto sirve para que, al momento de entrar al módulo de recolección de datos, se pueda elegir el periodo del cual se requiere obtener los datos que aporta el sistema, e igualmente de esta manera se indica que los datos que registrarán los coordinadores se guardaran dentro del plan, en ese periodo seleccionado, es decir, el dato de la fecha que se guarda en el plan académico que esté llenando el coordinador, será la fecha del periodo académico.

Creación de tablas para los datos no existentes

Creación de tablas – Debido a que muchos datos requeridos, no se encuentran ordenados en la base de datos o simplemente no hay registro de ellos, se creó una nueva tabla en la cual se definieron los campos para guardar estos datos, y /para poder relacionarlo con los datos ya existentes se le asignaron los campos de PLAN_ID, donde se guarda el número del grupo al que pertenece este plan, ya que si se guarda solamente el plan y no el grupo, la información solo se mostraría en ese plan y no en todos los planes del mismo grupo. PLAN_NOMBRE, este campo sirve para identificar el plan y poder hacer el filtro por facultades y que el coordinador sepa que plan está llenando con los datos requeridos, y el campo de la FECHA la cual representa el periodo).

Creación del modelo

El modelo es la primera capa, la cual es la base del patrón MVC, ya que esta es la encargada de gestionar la información, tanto la que puede ser ingresada en las vistas por el usuario, como la que se obtiene mediante la base de datos. Bean – Es la parte del modelo que se encarga de conectar con la base de datos y abstraer su información en un objeto. Después se asigna el valor “-1” a todos los campos y se crean los setters y getters los cuales tienen la función de asignar y de obtener la información de estos campos. Para este módulo se creó un archivo de este tipo.

□ SaiiDatosExtra.java

```

125 public SaiiDatosExtra(){
126     PLAN_ID = "-1";
127     PLAN_NOMBRE = "-";
128     FECHA = aca.util.Fecha.get Hoy();
129
130     DIS_N_IMG = -1;
131     DIS_R_IMG = -1;
132     DIS_TOTAL = -1;
133     REG_ALUM_H = -1;
134     REG_ALUM_M = -1;
135     REG_ALUM_T = -1;
136     CUR_REM_H = -1;
137     CUR_REM_M = -1;
138     CUR_REM_T = -1;
139     RET_SEG_H = -1;
140     RET_SEG_M = -1;
141     RET_SEG_T = -1;
142     DESERT_H = -1;
143     DESERT_M = -1;
144     DESERT_T = -1;
145     DESERT_19 = -1;
146     DESERT_20_24 = -1;
147     DESERT_25_29 = -1;
148     DESERT_30 = -1;
149     DESERT_DIS = -1;
150     TRAB_SIM_H = -1;
151     TRAB_SIM_M = -1;
152     TRAB_SIM_T = -1;
153     MOVIL_NAC_H = -1;
154     MOVIL_NAC_M = -1;
155     MOVIL_NAC_T = -1;
156     MOVIL_NAC_19 = -1;
157     MOVIL_NAC_20_24 = -1;
158     MOVIL_NAC_25_29 = -1;
159     MOVIL_NAC_30 = -1;
160     MOVIL_NAC_DIS = -1;
161     MOVIL_INT_H = -1;
162     MOVIL_INT_M = -1;
163     MOVIL_INT_T = -1;
164     MOVIL_INT_19 = -1;
165     MOVIL_INT_20_24 = -1;
166     MOVIL_INT_25_29 = -1;

```

```

248
249
250 public int getTotal_EDAD() {
251     return TOTAL_EDAD;
252 }
253 public void setTOTAL_EDAD(int total_EDAD) {
254     TOTAL_EDAD = total_EDAD;
255 }
256 public int getTotal_NI_E() {
257     return TOTAL_NI_E;
258 }
259 public void setTOTAL_NI_E(int total_NI_E) {
260     TOTAL_NI_E = total_NI_E;
261 }
262 public int getNIVEL_PREV() {
263     return NIVEL_PREV;
264 }
265 public void setNIVEL_PREV(int nivel_PREV) {
266     NIVEL_PREV = nivel_PREV;
267 }
268 public int getTotal_D_E() {
269     return TOTAL_D_E;
270 }
271 public void setTOTAL_D_E(int total_D_E) {
272     TOTAL_D_E = total_D_E;
273 }
274 public int getTotal_D_A() {
275     return TOTAL_D_A;
276 }
277 public void setTOTAL_D_A(int total_D_A) {
278     TOTAL_D_A = total_D_A;
279 }
280 public int getTotal_D_F() {
281     return TOTAL_D_F;
282 }
283 public void setTOTAL_D_F(int total_D_F) {
284     TOTAL_D_F = total_D_F;
285 }
286 public int getTotal_RI_E() {
287     return TOTAL_RI_E;
288 }
289 public void setTOTAL_RI_E(int total_RI_E) {

```

Fig. 2 Encabezado del objeto SaiiDatosExtra

Mapper- Contiene todas las variables del modelo, haciendo referencia al modelo existente en la base de datos. Recibe el objeto que fue creado en el bean y crea un método de tipo mapper con todas las variables del objeto, es decir, con todos los campos que se extrajeron de la tabla de la base de datos.

El mapper que se creó para este módulo es:

□ SaiiDatosExtraMapper.java

Pruebas y lanzamiento

Pruebas en modo local

Para determinar la eficiencia y el funcionamiento correcto de un software, se necesita hacer diversas pruebas, de esta manera se encuentran errores y también formas de mejorar algunas cosas para optimizar este mismo. El implementar esta aplicación dentro del sistema académico nos da una gran ventaja, ya que el sistema está dividido en dos modos, el primero y el cual utilizamos para las pruebas básicas o para las pruebas sin consecuencias, ya que este modo, no es el sistema que se usan los empleados de la universidad durante el día, solamente es usado por el departamento de desarrollo para programar y lo que se programa dentro de él, si pasa las pruebas, es añadido y subido en el modo real del sistema.

En este modo realizamos varias pruebas sobre todo del guardado de datos y de los filtros de información para los coordinadores, y así se logró encontrar varios errores, por ejemplo, que, al guardar los datos, se creaban varios registros del mismo guardado y el sistema no sabía cuál de estos mostrar en las vistas correspondientes, o que la información estaba separada por planes y se requería que toda la información de todos los planes de una misma carrera, estuviera junta. Sin embargo, también tiene desventajas el realizar pruebas en modo local, ya que son pocas personas las que realizan las pruebas y muchos errores se encuentran después de un uso extendido y masivo de la aplicación.

Pruebas en modo de producción

Después de realizar pruebas en el modo local y mal no encontrar más errores en estas, el ingeniero Ery Torres, el cual es el director del departamento de desarrollo y también el líder del proyecto, decidió que debíamos de subir el proyecto al modo de producción, es decir, al modo real del sistema, y dar acceso a las personas que hacen la recolección de datos, para poder hacer pruebas. de manera masiva, ya que son al menos 20 personas las que tienen esta tarea encomendada. Al realizar estas pruebas, se encontraron muy pocos errores, los cuales eran errores de sintaxis dentro de la programación del controlador, y se corrigieron tan solo con cambiar el nombre de una variable que se estaba usando para guardar 3 datos, en lugar de guardar solamente 1 dato.

Modificaciones

Al terminar de hacer pruebas en el modo real, se decidió hacer algunas modificaciones menores sobre todo en la parte del llenado de datos. La primera modificación fue que se cambió el método con el cual se obtenía el dato del promedio general de calificaciones, ya que el método existente no obtenía un dato preciso. Para esto, se cambió la tabla de la cual se obtenía este promedio, ya que la tabla que se está usando actualmente, no necesita más que una sola conexión con otra tabla, y esto evita que sucedan muchos errores al no tratar de hacer una conexión múltiple como se hacía con la tabla anterior, la cual se debía conectar con otras 3 tablas, para comparar algunos datos.

La segunda modificación que se hizo, fue que en algunos datos, no se entendía muy bien que es lo que se debía de ingresar, y se agregó una pequeña descripción, la cual se puede ver al momento de poner el cursor sobre el campo a llenar de esos datos, y también se decidió cambiar un dato de promedio de los alumnos recién egresados, ya que este dato, se debía llenar como un numero con punto decimal, y al principio del desarrollo de esta parte del módulo, se había programado para guardar un número entero y no un número decimal.

Lanzamiento del módulo

Una vez terminadas las modificaciones, el ingeniero Ery Torres, dio autorización para lanzar el módulo al sistema real, pero esta vez dando los permisos respectivos a los coordinadores y secretarios encargados del registro de datos, y también al departamento de efectividad, para que este pudiera monitorear todo el progreso de los coordinadores. En el momento del lanzamiento de un software siempre existe la posibilidad de que aun queden errores sin corregir, sin embargo, al lanzar este proyecto, el departamento de efectividad, externó su satisfacción con respecto al funcionamiento del módulo, y no se encontraron errores, pero siempre debe de haber una ventana de mejora, es decir, una manera de mejorar aspectos y procesos del software.

Resultados

Al finalizar este proyecto se entregó como resultado un módulo del sistema académico

institucional en el cual se pueden consultar los datos solicitados por el Sistema Analítico de Información de Instituciones, para ser administrado por el departamento de Efectividad Institucional, teniendo como colaboradores en la recolección total de los datos solicitados a los coordinadores de las facultades.

Vistas terminadas

Vista de los reportes

En esta vista llamada “reporte.jsp”, se muestran los diferentes planes académicos ordenados por grupos según el periodo seleccionado, para posteriormente poder consultar los datos de ese grupo, sin importar cuál de los planes de este se seleccione. El contenido de la vista es el siguiente:

- Lista de los periodos académicos
- Id del grupo de planes
- Nombre del plan académico

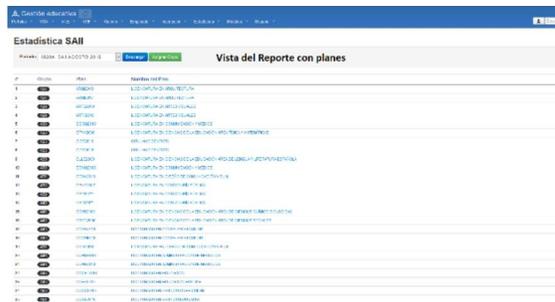


Fig. 6 Vista del reporte de planes

Despliegue de los datos según el plan seleccionado

En esta vista de llamada “plan.jsp”, se despliegan todos los 157 datos solicitados por el SAii, tanto los datos consultados del Sistema académico y los datos ingresados por parte de los coordinadores si es que ya existen, y mostrando los datos que aún no se han ingresado con el indicador “-1”, para poder ser detectado el campo de manera visual fácilmente.

El contenido de esta vista es el siguiente:

- Datos solicitados por el SAii
- Botón de ingreso de datos por parte del Coordinador

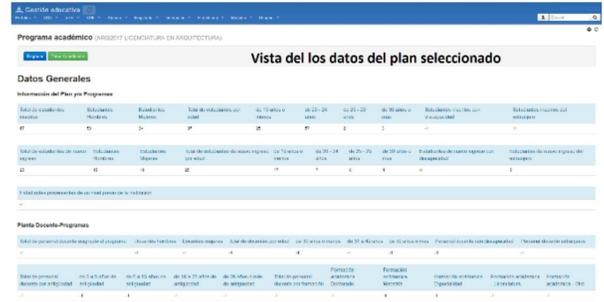


Fig. 7 Vista de los datos del plan académico

Formulario de llenado de datos

En esta vista llamada “datos.jsp”, se encuentra el formulario para el ingreso de datos que no se encuentran existentes en la base de datos, es decir los datos que ingresan los coordinadores, así como también la parte de edición si es que ya se han introducido los datos.

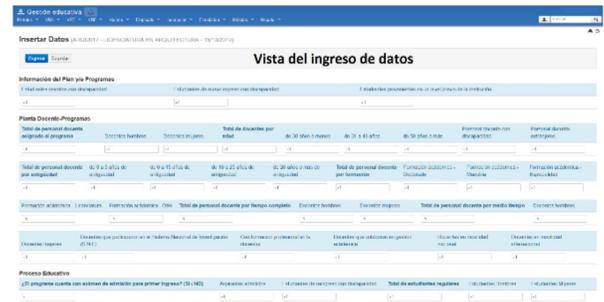


Fig. 8 Vista del ingreso de datos

Conclusiones

Indudablemente en las diversas áreas de las organizaciones sean instituciones o empresas, existen procedimientos que se llevan a cabo de manera rudimentaria (captura manual) que bien podrían verse agilizados y hacerse más precisos por medio de la utilización de diversas aplicaciones tecnológicas computacionales que sirven para facilitar las tareas. En el caso del desarrollo de este módulo, al ser construido dentro de un sistema ya existente, la integración del patrón arquitectónico en el cual está basado este sistema, fue la parte clave del desarrollo. El patrón arquitectónico MVC se ajusta de manera correcta a los requerimientos del proyecto, ya que proporciona una manera de gestionar los datos y una interfaz de interacción del usuario con la información, de una forma efectiva y fácil de entender, ya que este permite la conexión con la base de datos para obtener y guardar registros de

datos y también permite mediante la capa de las vistas, una manera en la que el usuario pueda utilizar este módulo. Al estar distribuido el trabajo en diferentes capas, es más sencillo corregir errores y agregar cosas para mejorar la recolección de datos, ya que todas las capas tienen su función y solo se encargan de esa función y no están todas mezcladas en un solo código que tendría que hacer todo el trabajo por sí solo.

Realizar este proyecto ha aportado valiosos aprendizajes entre los que conviene mencionar que A) El desarrollador de sistemas debe aprender a establecer un equilibrio entre lo que se considera correcto en un desarrollo y lo que desea conseguir el usuario. B) En algunos casos será necesario adecuarse a la capacidad instalada, equipo y software de que dispone el usuario. Ya que se vuelve difícil o imposible realizar desarrollos de vanguardia cuando el cliente o usuario no tiene acceso a la tecnología de punta.

Trabajo futuro

Indiscutiblemente ningún desarrollo tecnológico de aplicaciones es determinante o perfecto. De hecho, todos los desarrollos de aplicaciones en algún momento necesitarán mantenimiento y deberán ser actualizados o sustituidos por alguna innovación tecnológica o por cambios en la necesidad de su utilización.

En el caso de este proyecto no es presumible que haya quedado definitivamente concluido. Tan solo considerando los cambios de paradigma que la pandemia covid-19 ha ocasionado a los procesos educativos, con toda seguridad el organismo FIMPES incluirá en la solicitud de datos insumos que en forma tradicional no habían sido requeridos. Es sabido que la FIMPES ha realizado modificaciones a sus formatos en más de una ocasión. De manera que en la medida que se modifiquen o agreguen campos a los formatos en los que debe capturarse la data para la FIMPES, deberán hacerse los ajustes necesarios a los campos de este módulo.

Es también oportuno mencionar que en la Dirección de Efectividad Institucional el reporte SAIi no es el único reporte de datos que se genera. Hay una cantidad considerable de reportes que se entregan que son llenados a partir de la base de datos del sistema académico de la institución. El

llenado de estos formatos podría agilizarse y hacerse más preciso y, por lo tanto, confiable, por medio del desarrollo de aplicaciones tecnológicas computacionales que permitan una captura automática de los datos.

Teniendo en cuenta lo mencionado anteriormente, el patrón arquitectónico MVC, nos ofrece una oportunidad de mejorar el proyecto conforme lo requiera el departamento de Efectividad Institucional, ya que las capas de este patrón dividen ordenadamente el código, y de esta manera se puede encontrar fácilmente la sección del código que se desea mejorar o cambiar.

Referencias

- [1]” 5 porqués, análisis de la causa raíz de los problemas” <https://www.progressalean.com/5-porques-analisis-de-la-causa-raiz-de-los-problemas/>.
- [2]” Arquitectura de software”<https://dialnet.uniri.es/servlet/articulo?codigo=4786655>.
- [3]” Arquitectura multicapa”
https://www.researchgate.net/publication/235408808_Modelado_de_la_Variabilidad_en_aquituras_Multicapas.
- [4]” Arquitectura MVC”
<https://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html>.
- [5]” Funcionamiento de la arquitectura MVC”
<https://www.fdi.ucm.es/profesor/jpavon/poo/2.14.MVC.pdf>.
- [6]” Conoce el lenguaje JAVA”
<https://www.seas.es/blog/informatica/conoce-el-lenguaje-de-programacion-java/>.
- [7]” Índice TIOBE” <https://www.tiobe.com/tiobe-index/>.
- [8]” Tutorial para principiantes en Spring Tools”
<https://www.codejava.net/frameworks/spring/spring-mvc-beginner-tutorial-with-spring-tool-suite-ide>.
- [9]” Getting Started with Oracle SQL Developer”
https://www.oracle.com/webfolder/technetwork/tutorials/obe/db/11g/r2/prod/appdev/sqldev/sqldev_mngdb/sqldev_mngdb_otn.htm.
- [10]” Uso básico del Sistema de control de subversiones”
<https://ccc.inaoep.mx/~balderas/documents/misc/introduccion-subversion-2011.pdf>.