**3. Fase de Implementación y Pruebas**

En este capítulo se muestra la implementación del proyecto, describiendo las herramientas de desarrollo y mostrando los resultados de las pruebas realizadas del producto final.

**3.1 Implementación del proyecto**

**3.1.1 Lenguaje de Programación y Motor de Base de Datos**

**3.1.1.1 JAVA**

En (java.com) se encuentra la descripción más clara de lo que es Java:

“Java es un lenguaje de programación y una plataforma informática comercializada por primera vez en 1995 por Sun Microsystems. Hay muchas aplicaciones y sitios web que no funcionarán a menos que tenga Java instalado y cada día se crean más. Java es rápido, seguro y fiable. Desde portátiles hasta centros de datos, desde consolas para juegos hasta súper computadoras, desde teléfonos móviles hasta Internet, Java está en todas partes.”

Java nos ofrece una gran cantidad y variedad en la documentación para el desarrollo y codificación de algoritmos genéticos, tanto que hasta incluye librerías dedicadas a este tipo de algoritmos.

Desarrollos como el de (Naupri) fueron programados en Java obteniendo resultados óptimos, por lo cual, teniendo en cuenta esto y las características de fiabilidad, portabilidad, eficiencia, entre otras, que posee Java, este proyecto se realizó utilizando este lenguaje de programación.

**3.1.1.2 PostgreSQL**

(postgresql.org) lo describe de manera general así:

**“**PostgreSQL es un potente sistema de base de datos, de código abierto objeto-relacional. Cuenta con más de 15 años de desarrollo activo y una arquitectura probada que se ha ganado una sólida reputación por su fiabilidad, integridad de datos y corrección. Se ejecuta en todos los sistemas operativos, incluyendo Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64), y Windows.”

Es altamente escalable tanto en la enorme cantidad de datos que puede manejar y en el número de usuarios simultáneos que puede acomodar. Hay sistemas activos de PostgreSQL en entornos de producción que manejan más de 4 terabytes de datos. Algunos de los límites generales de PostgreSQL se describen a continuación.

* Tamaño máximo de base de datos: Ilimitado
* Máximo Tamaño de la tabla: 32 TB
* Máximo tamaño de fila: 1.6 TB
* Tamaño máximo de Campo: 1 GB
* Número máximo de filas por tabla: Ilimitado
* Columnas máximas por tabla: 250 - 1600 dependiendo de los tipos de columna
* Los índices máximos por tabla: Ilimitado

Las anteriores características y muchas otras ventajas que ofrece PostgreSQL lo hace la herramienta perfecta para el manejo de la cantidad de información que se puede manejar resolviendo el problema del School Timentabling.

**3.1.2 Patrón MVC en Java Swing**

Para el desarrollo del proyecto se implementó el patrón MVC (Modelo Vista-Controlador) que nos permite una mayor organización en la estructura del proyecto y el flujo de datos del software.

Según (Bascón) MVC es un patrón de diseño que considera dividir una aplicación en tres módulos claramente identificables y con funcionalidad bien definida: El Modelo, las Vistas y el Controlador.

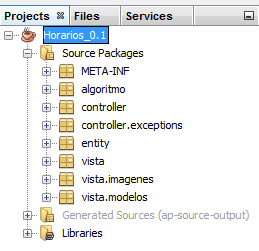


Fuente: Bascón Pantoja

La implementación de Swing del patrón de diseño MVC presenta muchas ventajas:

* Permite la creación de interfaces de usuario de una manera sencilla y rápida, permitiendo el manejo del patrón MVC pero ocultando los detalles de su implementación.
* El mecanismo de eventos de Java se adapta perfectamente al mecanismo de notificaciones de MVC.
* Al estar los modelos separados de la vista, las posibilidades de extensión de la librería y de personalización de componentes ya existentes son enormes.
* Permite al usuario crear sus propias estructuras de datos y adaptar la interfaz de usuario a ellas y no a la inversa, como sucede con librerías ya implementadas.

**3.1.3 Estructura de Paquetes del Proyecto**

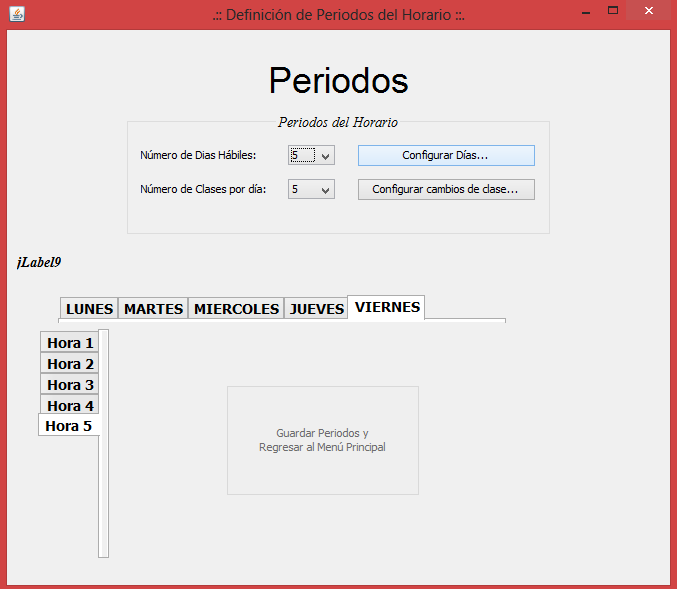


**3.2 Pruebas del Sistema**

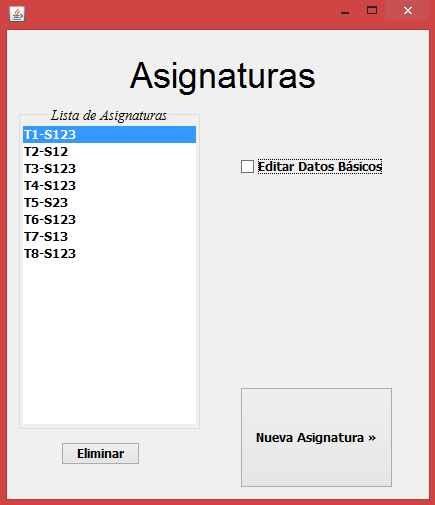
**3.2.1 Caso de Ejemplo**

A continuación se muestra paso a paso la implementación del caso de ejemplo con el cual se realizaron las pruebas del sistema:

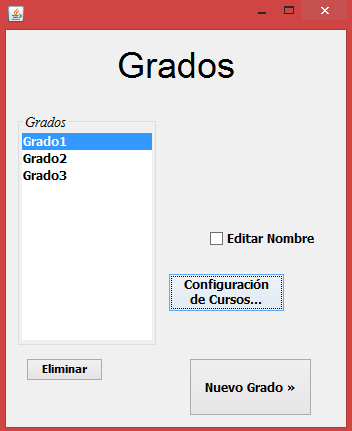
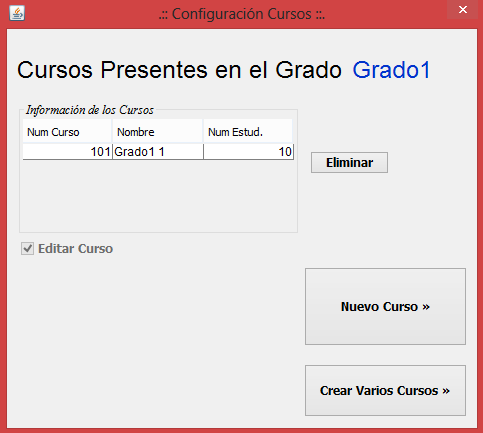
* **Configuración de periodos:** 5 días a la semana con 5 horas diferentes de clase por día.



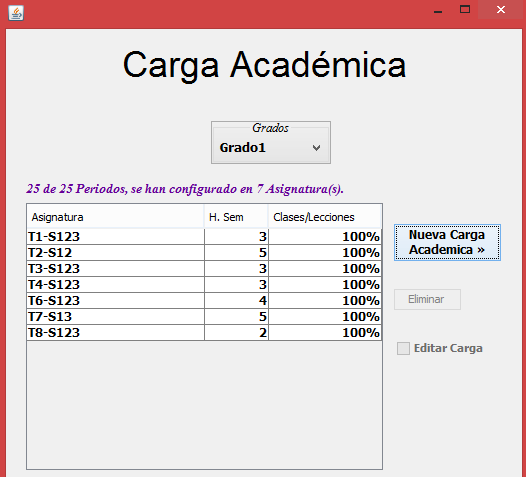
* **Configuración de Asignaturas:** En total 8 asignaturas.



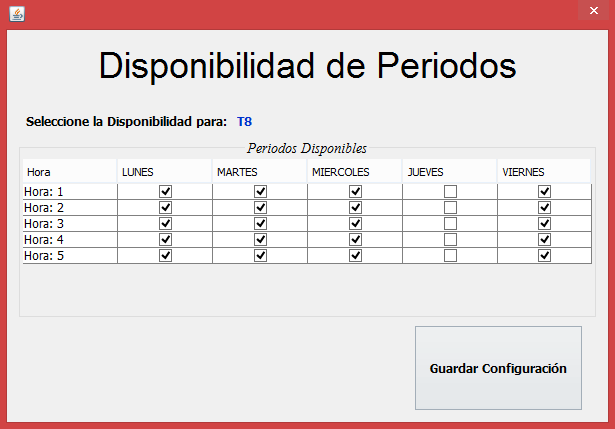
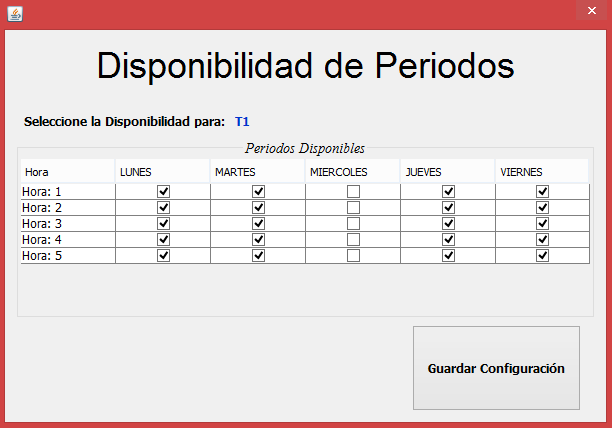
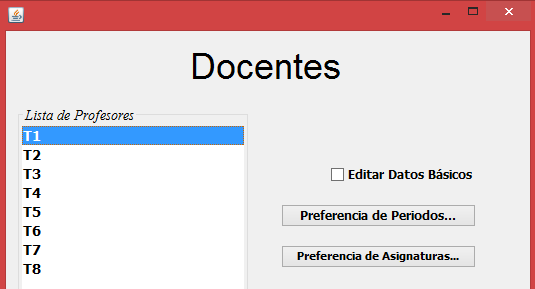
* **Configuración de cursos:** 3 cursos, 1 en cada grado (Grado1 – Curso 101, Grado 2 – Curso 201, Grado 3 – Curso 301). Cada curso cuenta con un total de 10 estudiantes.

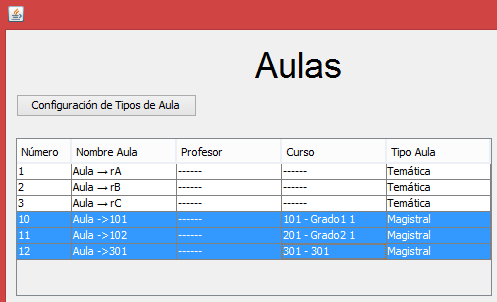
* **Configuración de carga académica:** Se asignaron cierta cantidad de horas a la semana en las cuales debe impartirse cada materia, estas varían entre 2, 3, 4 y 5 horas a la semana.



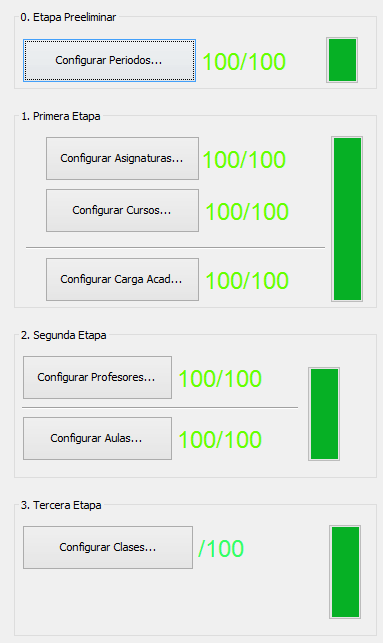
* **Configuración de Profesores:** En total son 8 profesores. A cada profesor se le asignó una configuración de preferencia de periodos y de asignaturas, de tal manera cada uno tiene 1 día libre a la semana, y así, al generar el horario final podamos ver que la solución se ajusta a los parámetros asignados.



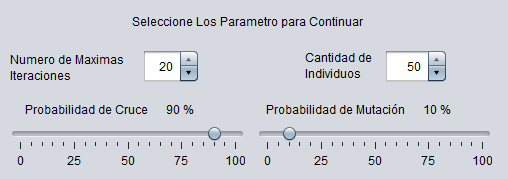
* **Configuración de Aulas:** A cada curso se le asignó 1 aula.



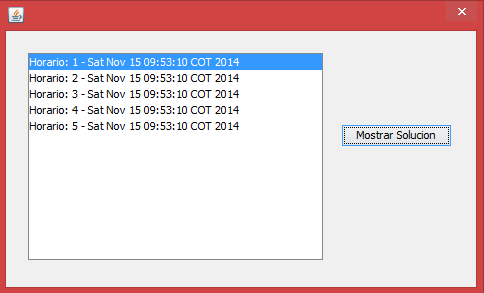
* Finalmente, obtuvimos que la configuración estaba completa, y se procede a la generación de la solución.



* **Parámetros para la ejecución del algoritmo:** Se variaron los parámetros de probabilidad de cruce y mutación, de tal manera que se puedan observar los cambios en cuanto a la cantidad y la calidad de soluciones que se generan para el ejemplo.



* **Generación de la solución:** Una vez configurados todos los parámetros requeridos para generación del horario para el ejemplo. Ya con el horario generado, se pueden visualizar las diferentes soluciones obtenidas y los diferentes reportes de cada uno de ellos.

****

****

**3.2.2 Prueba Piloto – Colegio María Mercedes Carranza**