

**ALGORITMO PARA LA PLANIFICACIÓN DE LOS HORARIOS DE CLASES EN
EL PROGRAMA DE INGENIERÍA DE SISTEMAS PARA LA UNIVERSIDAD DE
PAMPLONA, NORTE DE SANTANDER.**

NORBEEY PAEZ AMAYA

1094266505

Programa de Ingeniería de Sistemas

Universidad de Pamplona

Pamplona

2019

**ALGORITMO PARA LA PLANIFICACIÓN DE LOS HORARIOS DE CLASES EN
EL PROGRAMA DE INGENIERÍA DE SISTEMAS PARA LA UNIVERSIDAD DE
PAMPLONA, NORTE DE SANTANDER.**

NORBEEY PAEZ AMAYA

1094266505

Director: ING. YESID MADRID

Asesor: MG. LUIS ALBERTO ESTEBAN

Libro presentado como requisito para optar por el Título de Ingeniero de Sistemas

Programa de Ingeniería de Sistemas

Universidad de Pamplona

Pamplona

2019

Dedicatoria

Este proyecto está dedicado a Dios en primera instancia por darme la vida y la claridad para desarrollarlo, a mis padres por darme el apoyo y su confianza, a la Universidad de Pamplona por permitirme recibir un conocimiento y ayudarme a proyectarlo en aras de ser cada día mejores, al profesor Luis Alberto Esteban por la asesoría constante y correcciones al proyecto cada vez que se requería.

Al profesor y director guía del proyecto Yesid Madrid por dedicarle tanto tiempo a supervisar el avance del proyecto y sus conocimientos ayudaron a darle solución a varios problemas que iban surgiendo conforme avanzaba el desarrollo.

Contenido

1.	Generalidades	12
1.1	Planteamiento del problema.....	12
1.2	Justificación	12
1.3	Objetivo General	12
1.4	Objetivos Específicos.....	12
1.5	Metodología de Trabajo	13
1.5.1	Revisión Bibliográfica.....	13
1.5.2	Condiciones de Uso	13
1.5.3	Estudio del Modelo	14
1.5.4	La Interfaz.....	14
1.5.5	El Algoritmo.....	14
1.5.6	Validación	15
2.	Marco Teórico y Estado del Arte	16
2.1	Planificación de Horarios.....	16
2.1.1	Herramientas Existentes.....	16
2.1.2	Herramientas de Programación.....	22
2.1.3	Plataformas	23
2.1.4	Bases de Datos	30
2.1.5	Estrategias De Programación.....	32
2.2	Estado del Arte	34
3.	Propuesta de Algoritmo	54
3.1	Requerimientos Iniciales	54
3.2	Funcionamiento del Algoritmo.....	55
3.3	Herramienta Desarrollada	57
3.3.1	Requerimientos Técnicos	57
3.3.2	Modelo de datos	57
3.3.3	Funcionalidad	60
3.4	Validación de la Herramienta	65
4.	Conclusiones, Recomendaciones y Trabajos futuros.....	66
4.1	Conclusiones	66
4.2	Recomendaciones	67

4.3	Trabajo a futuro	67
5.	Bibliografía	68
6.	Anexos.....	73
6.1	Manual de Usuario	74

Introducción

En los comienzos de las instituciones de educación superior debido al nacimiento de las diversas facultades y programas que las componen, este problema de asignación de un horario ideal se ha intentado abordar durante largo tiempo y de esto se han generado soluciones parciales, pero aún sigue siendo un problema en las instituciones de educación superior. Por lo tanto, dentro de lo más sencillo de tratar es separar el problema por variables como lo son aulas, materias y docentes. El resultado de este proyecto es un algoritmo desarrollado en Python utilizando el framework Django y el gestor de base de datos de Mysql, manejando restricciones en las tres variables planteadas para facilitar la asignación y tratar de buscar el mejor horario posible, para asignar todas las variables sin inconvenientes en el menor tiempo. Esto fue posible manejando la generación de un reporte descargable del objeto generado en un (1) segundo, y se logró un tiempo de aproximadamente veintiocho punto seis segundos (28.6seg) estadística calculada tomando los tiempos resultado al terminar la ejecución del algoritmo durante 20 veces esto al momento de realizar las inserciones en la base de datos.

En el capítulo de generalidades, se describió la pregunta que nace de la necesidad con la que cuenta el programa de Ingeniería de sistemas de la Universidad de Pamplona, describiendo la implementación de un algoritmo para la planificación de los horarios. Este capítulo, incluye la pregunta del problema y su respectiva solución, también contiene los objetivos que se plantearon para solucionar el problema y la metodología de trabajo.

En el capítulo marco teórico y estado del arte se encuentra toda la información que se encontró en bases de datos y páginas web acerca de los proyectos y artículos

que documentan el uso de algoritmos para la organización del horario de instituciones educativas, en Europa, Latinoamérica y en Colombia.

En el capítulo Propuesta del algoritmo se explica cómo fue el proceso de crear un algoritmo partiendo de las condiciones que fueron detectadas por docentes que antes ya trabajaban el campo de la generación de horarios donde se plantearon tres (3) variables importantes a considerar al momento de la ejecución del algoritmo.

En el capítulo de Herramienta desarrollada se encuentra el análisis de los recursos con los que se cuenta para el desarrollo del proyecto las condiciones y funcionalidades que se debieron tener en cuenta el desarrollo del algoritmo.

En el capítulo de Conclusiones, recomendaciones y trabajos futuros se establecen los resultados obtenidos en la realización del proyecto y se dan recomendaciones para la implementación del proyecto y sus respectivas mejoras.

Glosario

Algoritmo: “Un Algoritmo, se puede definir como una secuencia de instrucciones que representan un modelo de solución para determinado tipo de problemas. O bien como un conjunto de instrucciones que realizadas en orden conducen a obtener la solución de un problema. Por lo tanto, se puede decir que es un conjunto ordenado y finito de pasos que permite solucionar un problema. Los algoritmos son independientes de los lenguajes de programación. En cada problema el algoritmo puede escribirse y luego ejecutarse en un lenguaje de diferente programación. El algoritmo es la infraestructura de cualquier solución, escrita luego en cualquier lenguaje de programación”. (frida, 2009)

Base de datos: “Una base de datos de un Sistema de Información (SI) es la representación integrada de los conjuntos de entidades instancia correspondientes a las diferentes entidades tipo del SI y de sus interrelaciones. Esta representación informática (o conjunto estructurado de datos) debe poder ser utilizada de forma compartida por muchos usuarios de distintos tipos”. (Paré, 2005)

Bases de datos orientadas a objetos: “Una base de datos orientada a objetos es un sistema de gestión de base de datos mediante el cual representa la información en forma de objetos que son utilizados en programación orientada a objetos. La investigación y el desarrollo de la base de datos orientada a objetos fue impulsada en gran medida por su alto rendimiento en almacenamiento de datos y los requerimientos de acceso que tenían entornos de soporte de diseño tales como el CAD (Computer Aided Design) y el CASE (Computer Aided Software Engineering)”. T. (management, 2017)

Framework web: “El concepto framework se emplea en muchos ámbitos del desarrollo de sistemas software, no solo en el ámbito de aplicaciones Web. Se pueden encontrar frameworks para el desarrollo de aplicaciones médicas, de visión por computador, para el desarrollo de juegos, y para cualquier ámbito que

pueda ocurrírseles. En general, con el término framework, nos estamos refiriendo a una estructura software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación. En otras palabras, un framework se puede considerar como una aplicación genérica incompleta y configurable a la que podemos añadirle las últimas piezas para construir una aplicación concreta”. (Gutiérrez., 2014)

Sistema de Información: “Es un conjunto formal de procesos que, operando sobre una colección de datos estructurada según las necesidades de la empresa, recopila, elabora y distribuye la información necesaria para las operaciones de dicha empresa y para actividades de dirección y control, correspondientes para desempeñar su actividad de acuerdo a sus estrategias de negocio. El sistema de información en una empresa debe responder a objetivos de la organización en los niveles operativo, táctico y estratégico. Ejemplo la actividad contable de una empresa como es el suministro de datos sobre las actividades, el suministro de informes y el mantenimiento de registros es un sistema de información contable que responde al objetivo de la empresa”. (Intercom, 2009)

Minería de Datos: “El datamining (minería de datos), es el conjunto de técnicas y tecnologías que permiten explorar grandes bases de datos, de manera automática o semiautomática, con el objetivo de encontrar patrones repetitivos, tendencias o reglas que expliquen el comportamiento de los datos en un determinado contexto. Básicamente, el datamining surge para intentar ayudar a comprender el contenido de un repositorio de datos. Con este fin, hace uso de prácticas estadísticas y, en algunos casos, de algoritmos de búsqueda próximos a la Inteligencia Artificial y a las redes neuronales”. (Sinnexus, 2017)

Heurística: “La heurística, como disciplina científica, y en su sentido amplio puede ser aplicada a cualquier ciencia con la finalidad de elaborar medios, principios, reglas, estrategias como ayuda para lograr encontrar la solución más eficaz y eficiente al problema que analiza el individuo”. (CHARLYRED70, 2016)

Los procedimientos heurísticos, se dividen en:

- “Principios heurístico, establece sugerencias para encontrar la solución idónea al problema.
- Reglas heurística, señalan los medios para resolver el problema.
- Estrategias heurísticas, son aquellas que permiten organizar los materiales o recursos compilados que contribuyen a la búsqueda de la solución del problema”. (CHARLYRED70, 2016)

GUI: “Las **GUI** o interfaces gráficas son un programa informático que realiza la función de interfaz de usuario. Está formado por imágenes y objetos gráficos, que representan la información y acciones que se encuentran en la interfaz. Su objetivo es el de crear un entorno visual fácil de usar para la comunicación entre las funciones y el sistema operativo”. (Jesuita, 2017)

MetaHeurística: “La metaheurística nace con el propósito de evitar tener que comenzar de cero cada vez que se presente un problema cuyo modelo no puede ser tratado por los métodos convencionales, para encontrar una solución óptima. De acuerdo con estos autores, la principal ventaja de una metaheurística bien diseñada, que permite moverse relativamente rápido hacia soluciones, proporcionando una forma muy eficiente de resolver problemas complicados. Entre tanto, la principal desventaja es que no existe garantía de que la mejor solución que se obtenga, sea una solución óptima o incluso que esté cerca de serlo”. (Gómez, 2015)

Variable: “En programación, una variable es un espacio de memoria reservado para almacenar un valor determinado que corresponde a un tipo de dato soportado por el lenguaje de programación en el cual se trabaja, es representada y usada a través de una etiqueta (un nombre simbólico) que le asigna un programador o que ya viene predefinida en el lenguaje. El programador emplea ese nombre de variable para poder usar la información que está contenida en ella. Durante el tiempo de ejecución del programa la variable puede adquirir un valor determinado y puede cambiar durante el curso de ejecución del mismo”. (Alegsa, 2018)

1. Generalidades

En este capítulo se encuentra detallado el planteamiento de problema, la justificación, objetivo general, objetivos específicos y la metodología de trabajo que se siguieron en el desarrollo del proyecto.

1.1 Planteamiento del problema

¿Qué técnicas y herramientas se deben usar para el diseño de un algoritmo capaz de planificar los horarios de clases del programa de Ingeniería de Sistemas para la Universidad de Pamplona, Norte de Santander?

1.2 Justificación

La gestión del proceso de planificación de horarios de los diversos programas de la Universidad de Pamplona contempla condiciones de ineficiencia al momento de tomar decisiones para la asignación óptima de horarios, recursos físicos y docentes para cada materia. Mediante el desarrollo de una aplicación basada en un Framework para aplicaciones web con un motor de bases de datos, se planea diseñar y aplicar un algoritmo que sea capaz de organizar los horarios de las materias y relacionarlos con todas las variables mencionadas anteriormente.

1.3 Objetivo General

Diseñar un algoritmo para planificar los horarios de clases en el programa de Ingeniería de Sistemas de la Universidad de Pamplona mediante la implementación de un prototipo de software web.

1.4 Objetivos Específicos

- Estudiar diferentes algoritmos relacionados con la planificación de horarios.
- Analizar las diferentes alternativas de plataformas, motores y lenguajes aplicados al desarrollo web.
- Implementar un algoritmo para la planificación de horarios de clases de acuerdo a las plataformas, motores y lenguajes seleccionados del estudio

previo.

1.5 Metodología de Trabajo

Describe la revisión bibliográfica, las condiciones de uso, estudio del lenguaje, el algoritmo, la interfaz y la validación del algoritmo.

1.5.1 Revisión Bibliográfica

La revisión bibliográfica que sustenta el desarrollo del algoritmo partió de las consultas previas sobre el tema, se detectó que el problema se encontraba en diversas instituciones de educación superior a nivel mundial que se encontraban en la misma pregunta, ¿cómo diseñar e implementar un algoritmo que permitiese el uso de materias, aulas y docentes?, por ejemplo, en nuestra Universidad:

- Un docente catedra que también es empleado público se le debía asignar un horario que no se le cruzara con sus horas de trabajo.
- Docentes que no son residentes de la ciudad y solicitan no se les asignara clases los fines de semana.
- Aula que son necesarias algunas veces por semana para realizar reuniones de docentes del programa

1.5.2 Condiciones de Uso

El estudio de las condiciones que presentaba crear un algoritmo propio para el programa de Ingeniería de Sistemas de la Universidad de Pamplona parte de la dificultad de usar los recursos sin que se crucen con los horarios de los docentes. Para esto se debieron tomar condiciones tratadas como heurísticas dentro del modelo para mejorar el nivel de complejidad que tendría el algoritmo y así brindar un resultado más óptimo.

1.5.3 Estudio del Modelo

Durante varios meses se planificó la manera de abordar el problema pues fue necesario dividirlo en características específicas para poder generar un modelo que fuera acorde con las necesidades del programa, se formularon reglas básicas que fueron implementadas en el algoritmo para minimizar los recorridos que hiciese en caso de no encontrar una posible solución, con esto se enfatizaba en que solo las variables que cumplieren en el momento de la ejecución con los datos requeridos fueran tenidas en cuenta.

1.5.4 La Interfaz

La interfaz fue creada utilizando el framework de Python llamado Django que brindó facilidades de integración con el código creado, puesto que se definió como un requerimiento el desarrollo del algoritmo fuera en Python y se diseñó una serie de vistas para facilitar el ingreso de los datos requeridos por el algoritmo para iniciar su proceso de asignación.

1.5.5 El Algoritmo

El algoritmo se desarrolló teniendo en cuenta reglas básicas que al momento de iniciar se definen como, por ejemplo:

- El Docente maneja unas restricciones: hay horas en las que el docente no puede dictar una materia.
- Las Materias: las materias tienen grupos que se asignan dependiendo de la cantidad de estudiantes y los grupos a su vez tienen bloques que las componen.
- Los Docentes tienen materias que se les facilita dictar: por lo tanto, se manejó la asignación de prioridades para cada materia entre más alto sea esto querrá decir que esa materia se le asignara primero.
- Las Aulas: estas en teoría no tienen ninguna restricción salvo en los casos que se necesita para realizar algún tipo de evento para esto se manejó unas restricciones que pueden ser llenadas.
- La cantidad de horas: Un docente dependiendo de su tipo de contratación

maneja una cantidad de horas en las que puede dictar clase.

- Los cruces: las materias no debían cruzarse en el mismo salón, ni en la misma hora, ni al mismo docente.
- Las materias del mismo semestre no debían cruzarse, esto se debe realizar para facilitarle a los estudiantes que van en línea puedan ver toda la carga asignada al semestre.
- Se debe asignar todas las materias posibles al aula para que el tiempo que este desocupado sea de intervalos pequeños.

1.5.6 Validación

Se debía generar un reporte para comprobar que la asignación de las tres variables materias, docentes y aulas si se realizara conforme a los parámetros que se establecieron en las condiciones del algoritmo.

2. Marco Teórico y Estado del Arte

Este capítulo comprende la planificación de horarios como termino general, las herramientas existentes, herramientas de programación y el estado del arte tiene en cuenta los proyectos que se han realizado en otras universidades con el mismo objetivo.

2.1 Planificación de Horarios

La planificación de Horarios trata de encontrar la mejor solución posible a la asignación de materias , aulas y docentes, sin que ninguno de estos se crucen en un punto concreto de ahí parte la dificultad pues la cantidad de condiciones y variables hace que el problema se vuelva de difícil solución, un ejemplo claro tomando el artículo “Karen E. Hinton, de la Society for College and University Planning, ha debido asesorar en el área de planificación académica a varias instituciones educación superior en los Estados Unidos. Puntualiza, por ejemplo, que un instituto “no podía decidirse sobre el número de salas de clases que necesitaba para cumplir con su nueva planificación estratégica. En lugar de ello, se basaba en estimaciones de sus jefes de departamento, fundamentadas en su experiencia con los problemas de asignación de horarios”. Luego de que terminaron la asesoría, se dieron cuenta con base al análisis de datos reales, que en el proceso “habían sobrestimado la demanda por un margen de error bastante alto.” (Newman, 2009), el problema parte de los recursos físicos que tenga la institución y cuántos de estos quiere asignar para que se dicten clases.

2.1.1 Herramientas Existentes

- **Agenda del Estudiante:** “Elaborada por estudiantes para estudiantes, y apta sólo para dispositivos con Android, puede organizar los horarios tanto a nivel personal (alumno) como profesional (centro de estudios). Para mejorar la identificación de cada cita, permite añadir una foto e información adicional. Incluye también servicio de notificaciones y la posibilidad de ir

marcando los eventos que van pasando como 'completados'. Su visualización puede ser por día, por semana o por mes". (Mármol, 2018)

- **ascTimetables:** "Sólo es necesario introducir los requisitos y el software de planificación evaluará todas las posibilidades para obtener un horario equilibrado. Y ante errores o cambios, esta herramienta reprogramará automáticamente todas las modificaciones. Entre sus ajustes manuales, es posible personalizar la fuente, el diseño y los logotipos, y exportarlo en formato pdf, Excel o compartirlo online". (Mármol, 2018)
- **Aula 1:** "El software de gestión escolar de Aula1 dispone de un generador de horarios integrado dentro de la plataforma que permite crear varios tipos de horarios. Por ejemplo, los docentes podrán crear distintos horarios según los niveles académicos en los que trabajan. También permite consultar los horarios incompletos en los que queden horas o materias sin asignar y comprobar los errores en las materias por distintos colores". (Mármol, 2018)
- **BEST (Bullet Education Scheduling and Timetabling):** "Este software permite a los centros educativos superar la complejidad y lentitud de realizar la programación de los horarios del curso. Para ello, combina y evalúa varias restricciones y objetivos, de acuerdo con las necesidades de los colegios, optimizando los horarios de profesores, estudiantes y aulas. Además, asigna automáticamente las salas disponibles y próximamente también será capaz de planificar los exámenes de forma automática. Este algoritmo fue desarrollado hace 15 años en Portugal y hoy en día tiene ya cerca de 200 clientes, algunos de ellos en España". (Mármol, 2018)
- **Canva:** "Aunque no es una herramienta específica para crear horarios ni está destinada al entorno educativo, esta herramienta de diseño gráfico

online cuenta con plantillas para ello. Todas ellas pueden personalizarse, introduciendo todas las clases con su horquilla horaria. El documento resultante podrá ser guardado en formato PDF o png para imprimirlo, compartirlo y publicarlo en formato de foto con facilidad. Además, se queda guardado en el perfil del usuario para poder duplicarlo o modificarlo al gusto”. (Mármol, 2018)

- **Class Buddy:** “Además de organizar los horarios de exámenes, reuniones de clase o extraescolares, esta app permite llevar el control de otro tipo de información como, por ejemplo, las notas de los exámenes o los trabajos realizados. Incluye una herramienta de backup, para que nunca se pierda la información, y se puede sincronizar con el calendario de Google. Cuenta con un widget personalizable que ofrece toda la información de un vistazo”. (Mármol, 2018)
- **DocCF:** “Se trata de un software de gestión académica y administrativa. Con él se pueden llevar a cabo cerca de 60 procedimientos distintos y entre ellos está la creación y asignación de horarios para las distintas clases. Ante cualquier problema, cuenta con un sistema de atención al cliente online a través de un web chat, y plataformas como Skype, Facebook Messenger o WhatsApp”. (Mármol, 2018)
- **Doodle:** “Esta herramienta permite crear cualquier tipo de horario ya sea personal o profesional y compartirlo entre los interesados a través de un enlace para decidir entre todos los días y las horas destinadas a cada actividad, con la opción de escoger las direcciones de correo electrónico de forma manual o automáticamente. Los participantes recibirán su horario y marcarán el espacio de tiempo que mejor se ajuste a sus agendas, para lograr entre todo un acuerdo”. (Mármol, 2018)
- **Esemtia de Edebé:** “Entre la gran variedad de herramientas que propone esta plataforma de gestión destinada a los cursos desde Infantil hasta FP

se encuentra una de creación de horarios. Lo hace directamente desde la nube y permite generarlos de forma flexible para cada día”. (Mármol, 2018)

- **FET:** “Es una aplicación de software libre que sirve para generar horarios académicos a través de un algoritmo. Es capaz de definir clases, tiempos, asignaturas, profesores, aulas y crear actividades en un máximo de 20 minutos. Una vez terminado, muestra varias opciones de horarios y genera un documento HTML con el elegido que se puede colgar en la web del centro para compartir con el resto de usuarios”. (Mármol, 2018)
- **Generador de horarios de ARASAAC:** “El Portal Aragonés de la Comunicación Aumentativa y Alternativa cuenta con diversas herramientas de gestión online para crear diferentes recursos, entre ellas un generador de horarios fácil de usar. Para ello, basta con elegir el número de días que debe contener, la hora de inicio y la duración de cada clase, así como el color, el tamaño de la tipografía de las celdas o incluir pictogramas”. (Mármol, 2018)
- **Handy Timetable:** “Con esta herramienta se pueden crear todo tipo de tareas sin ningún tipo de restricción de tiempo (ni de días ni de horas). Incluso se puede aplicar a dos fechas distintas. Permite añadir información adicional a cada nota, la cual se puede visualizar de una vez en la pantalla del dispositivo. Incluye también la opción de seleccionar aquellas que, por su relevancia, se quieren controlar en todo momento. Disponible en Android y iOS”. (Mármol, 2018)
- **Homework:** “Con un máximo de 20 horas al día, esta herramienta hace posible un seguimiento detallado tanto de las tareas a realizar como de sus horarios. Permite la inclusión de información adicional, como el nombre del profesor o la asignatura a impartir, y la opción de editarla si es necesario realizar algún cambio de última hora. Incluye un widget de escritorio”. (Mármol, 2018)

- **Horarios escolares Custombit:** “Con esta herramienta se generan los horarios de clase de profesores tanto de colegios como de universidades de manera automática. Tienen en cuenta la cantidad de horas semanales que le corresponden a cada asignatura y la disponibilidad de cada docente para no incurrir en una incompatibilidad de horarios”. (Mármol, 2018)
- **KronoWin:** “Partiendo de las asignaturas, los grupos, los profesores y las aulas, genera un horario escolar óptimo para cada centro. Además, incluye la posibilidad de cambiar la estructura de los horarios, replicar las líneas de asignaturas, enviarlas a cada departamento y filtrar cada horario por asignaturas, entre otras. Contiene cabeceras de impresión modificables como el logotipo del colegio y ofrece actualizaciones constantes para conseguir mayor comodidad en el trabajo”. (Mármol, 2018)
- **Peñalara GHC:** “El programa Peñalara GHC permite llevar a cabo la gestión completa de los horarios escolares, teniendo en cuenta los requisitos académicos, pedagógicos y organizativos del centro, así como sus recursos humanos y materiales. Válido para todo tipo de niveles e instituciones educativas, organiza de forma objetiva y mediante la elección de un perfil de usuario los horarios semanales. Gracias a su potente motor de cálculo es capaz de resolver cualquier problema de planificación de profesores, espacios, sesiones lectivas... Es posible acceder a los archivos del software a través de Internet, exportar los horarios a gestores de calendarios (como Office 365 o Google Calendar) o llevar a cabo la captación de desideratas de forma remota. En breve, estará disponible en Windows Store de Microsoft”. (Mármol, 2018)
- **SG Horarios:** “Este generador de horarios permite tanto la edición manual como automática de horarios, así como la gestión de aulas y de guardias, y la construcción de un calendario de reuniones para las evaluaciones. Entre

otras opciones es posible distribuir las actividades a lo largo de la semana y establecer las asignaturas opcionales en el inicio y fin de un bloque de clases, seguidas de los idiomas extranjeros. Además, añade el inicio y duración del almuerzo de los grupos y profesores, impide o restringe la colocación de actividades en tiempos lectivos y asigna los días libres de cada profesor”. (Mármol, 2018)

- **Timetable:** “Esta app, exclusiva de Android, facilita la organización de tareas de colegios y universidades. En cada nota nueva se puede incluir información de interés como el nombre de la asignatura, el aula en la que se imparte y el profesor. Incluye un servicio de notificaciones, las cuales podemos revisar en cualquiera de sus dos modos de visualización: lista y cuadrícula. Se sincroniza con todos los dispositivos a la vez”. (Mármol, 2018)
- **Untis:** “A través de un algoritmo de optimización, confecciona automáticamente los horarios del centro a partir de los criterios pedagógicos introducidos. El programa propone al usuario hasta diez alternativas diferentes, permitiendo incluso hacer cambios manuales. Una vez finalizado, realiza un diagnóstico para verificar que no existen entradas contradictorias en los datos o que no se incumplen los criterios definidos. Además, incorpora otras funcionalidades como la identificación de clases irregulares en el proceso de planificación, valores contractuales de horas en que los profesores deben impartir clases, horario óptimo de guardias, planificación de sustituciones o estadísticas, entre otras. Este generador de horarios está completamente integrado en la plataforma de gestión Alexia, ambos servicios de Educaria”. (Mármol, 2018)
- **Weeras Platform:** “Trabaja desde la nube y se adapta a cualquier necesidad educativa para facilitar la gestión integral de escuelas, academias y proyectos de carácter educativo. Los docentes podrán acceder

a cualquier recurso digital de aprendizaje, evaluar el progreso de los alumnos y promover la comunicación entre el profesorado y los familiares”. (Mármol, 2018)

2.1.2 Herramientas de Programación

En este inciso se describe las diferentes herramientas computacionales tenidas en cuenta al momento de abordar el modelo.

2.1.2.1 Lenguajes

Lenguaje JavaScript: “Se utiliza principalmente del lado del cliente, aunque se puede utilizar del lado del servidor. Actualmente y gracias a tecnologías como AJAX es utilizado para enviar y recibir información del servidor. Como principales ventajas, tenemos que destacar que es un lenguaje de scripting seguro y fiable, cuyos scripts tienen capacidades limitadas, debido a la seguridad. Como desventajas, podríamos mencionar que el código debe descargarse por completo y es visible por cualquier usuario”. (piensasolutions, 2017)

LENGUAJE PHP: “Es un lenguaje enfocado en la creación de webs dinámicas. Sus scripts son interpretados por el servidor y genera código HTML. Requiere Apache o IIS con librerías de PHP. Hereda su sintaxis de C, Java y Perl. Como principales ventajas, hemos de decir que es un lenguaje fácil de aprender y muy rápido. Soporta la orientación a objetos y utiliza un lenguaje multiplataforma. Además, puede conectarse con una gran cantidad de base de datos: MySQL, PostgreSQL, Oracle, MS SQL Server... No necesita que se definan los tipos de variables. Uno de sus aspectos más llamativos es que está diseñado con el fin de ser un lenguaje muy seguro para escribir CGI, más que Perl o C. Es el lenguaje base que utilizan la mayoría de CMS o gestores de contenidos más extendidos como WordPress, PrestaShop, Drupal o Joomla!”. (piensasolutions, 2017)

LENGUAJE PYTHON: “Considerado por muchos el lenguaje más limpio a la hora de programar. El código, al igual que JavaScript, es interpretado y no compilado.

Algo curioso en este lenguaje es que permite a los programadores elegir un estilo de programación concreto (objetos, estructurado, funcional...), debido a que es un lenguaje de programación multiplataforma. Entre las ventajas de Python, se destacan que es libre y de fuente abierta, de propósito general. Cuenta con muchas funciones y librerías, es multiplataforma y fácil de programar.” (piensasolutions, 2017)

LENGUAJE RUBY: “Como el anterior, es un lenguaje interpretado y está orientado a objetos. Hereda su sintaxis de Python y Perl. El lenguaje puede cargar librerías de extensiones dinámicamente si el sistema operativo lo permite. Además, es un lenguaje portátil. Otra ventaja que ofrece es que cualquiera puede encontrar en su página web gran cantidad de información y tutoriales”. (piensasolutions, 2017)

2.1.3 Plataformas

Enumera las plataformas más populares que se consultaron para el desarrollo del proyecto cada una fue organizada con su respectivo lenguaje.

2.1.3.1 Plataformas en Ruby

Ruby on Rails: “listado-ruby-frameworks-rails Si ya estás familiarizado con Ruby es probable que ya hayas escuchado sobre esta framework o incluso ya la estés utilizando. Ruby on Rails es una framework de código abierto. Es una de las frameworks más populares para Ruby. El objetivo principal de Rails es favorecer la convención antes de la configuración. Como muchas otras frameworks, se especializa en disminuir la repetición de código. Requiere una configuración mínima, de modo que puedes concentrarte en terminar tu proyecto”. (blog.aulaformativa, 2016)

SINATRA: “es un lenguaje específico de dominio (DSL) para crear aplicaciones web en Ruby de manera sencilla. Sinatra es en sí, una librería, debido a su popularidad. De hecho, algunos desarrolladores la consideran una framework

completa y ha servido como base para la creación de otras frameworks. Con Sinatra se puede estructurar una aplicación según el problema al que se quiera dar solución.” (blog.aulaformativa, 2016)

HOBBIT: “Esta micro-framework ha sido desarrollada teniendo como inspiración el DSL de Sinatra. Se caracteriza por ser rápida y requerir una configuración mínima. El objetivo por el cual se ha creado esta framework es para evitar que el usuario cree líneas de código innecesarias. Asimismo, al usar Hobbit se puede entender el uso de Rack y sus extensiones”. (blog.aulaformativa, 2016)

RAMAZE: “Es una framework que te permite simplificar los procesos más sencillos y ofrecer soluciones para procesos más complejos. Ha sido creado para que se pueda programar en base al modelo MVC. Ramaze ya lleva un buen tiempo en línea, de modo que ahora existe una gran comunidad que conoce sobre la framework y puede ofrecer soporte.” (blog.aulaformativa, 2016)

NANCY: “Es una micro-framework inspirada en Sinatra y sólo a este hecho se debe su nombre. Incluso ponen como subtítulo “La hija pequeña de Sinatra” en la página oficial a modo de reconocimiento la influencia de Sinatra en esta framework. Nancy es una framework rápida que comparte ciertas funciones que también tiene Sinatra”. (blog.aulaformativa, 2016)

PADRINO: “Es una framework que ha sido desarrollada en base a la librería web de Sinatra. Puede ser empleada para crear frameworks. El objetivo final de Padrino es extender la funcionalidad de Sinatra, de modo que los desarrolladores puedan crear aplicaciones más complejas teniendo un entorno ya conocido (que es Sinatra)”. (blog.aulaformativa, 2016)

Crepe: “Es una framework ligera que ha sido diseñada para agilizar el proceso de creación de APIs. Posee un DSL intuitivo que ha sido inspirado en RSpec para hacer que la creación de APIs sea mucho más sencilla”. (blog.aulaformativa, 2016)

NYNY: “Es una framework que posee las características básicas. Debido a que cada aplicación creada en NYNY entra dentro de la categoría de “middleware”, puede ser usada junto con Sinatra, Rails o cualquier otra aplicación que haya sido basada en Rack. Posee la misma interfaz para extensiones que Sinatra, aunque no todas estas extensiones funcionan correctamente en NYNY. Sin embargo, si alguna vez se desea emplear alguna extensión que haya sido específicamente desarrollada para esta framework, es improbable que presente problemas al correr en Sinatra”. (blog.aulaformativa, 2016)

RUBY GRAPE: “Es otra micro-framework desarrollada para complementar otras frameworks. Puedes usar Grape para crear APIs de manera más sencilla, de hecho esta es la razón principal por la cual se ha desarrollado esta framework en primer lugar: para reemplazar esta funcionalidad que poseen frameworks más grandes y populares como Rails”. (blog.aulaformativa, 2016)

CELLULOID: “Es una framework que les permite a los desarrolladores crear programas de objetos concurrentes de forma tan sencilla como crear programas secuenciales de objetos regulares. Celluloid puede ser usado tanto por programadores principiantes como expertos para poder crear programas de manera más sencilla”. (blog.aulaformativa, 2016)

HANAMI: “Es una completa framework para Ruby que posee un gran número de características. A pesar de ello, es una framework rápida que consume una cantidad menor de memoria que otras frameworks. Hanami posee un diseño minimalista para optimizar tu nivel de concentración”. (blog.aulaformativa, 2016)

SCORCHED: “Es otra framework ligera para Ruby basada en DSL, al igual que Sinatra. Con Scorched se pueden crear sitios web y aplicaciones sencillas. Su requisito es haber usado otras framework que posean como base el DSL”. (blog.aulaformativa, 2016)

CUBA: “Es una micro-framework para desarrollo web inspirada en Rum. Con Cuba se pueden crear aplicaciones basadas en Rack. También se pueden realizar pruebas y organizar el código. Uno de los principales objetivos de Cuba es que se puedan crear aplicaciones web con la menos cantidad de archivos posibles. Sin embargo, no sólo se limita a aplicaciones; también se pueden crear páginas de destino con Cuba”. (blog.aulaformativa, 2016)

CAMPING: “Es una framework desarrollada bajo el modelo MVC. Es sumamente ligera (4 KB). La idea detrás de la creación de Camping es que se pueda crear la estructura de una aplicación y mantenerla en un solo archivo. Como Camping se organiza bajo el modelo MVC, esa misma aplicación puede ser trasladada a Rails para añadir los toques finales o realizar pruebas. Camping se considera a sí misma como una framework desarrollada para experimentar y aprender Ruby”. (blog.aulaformativa, 2016)

2.1.3.2 Plataformas en PHP

Symfony 4: “Es uno de los frameworks más utilizados y recomendados por las empresas digitales y el preferido de muchos desarrolladores web. Symfony está compuesto por un amplio número de componentes que se pueden reutilizar, además de contar con una comunidad activa que siempre expone nuevos códigos para el desarrollo de posibles mejoras en las actualizaciones”. (TÉBAR, 2018)

Laravel: “El objetivo de Laravel es el de ser un framework que permita el uso de una sintaxis refinada y expresiva para crear código de forma sencilla, evitando el “código espagueti” y permitiendo multitud de funcionalidades. Aprovecha todo lo

bueno de otros frameworks y utiliza las características de las últimas versiones de PHP”. (García, 2015)

CakePHP: “Es un marco de desarrollo [framework] rápido para PHP, libre, de código abierto. Se trata de una estructura que sirve de base a los programadores para que éstos puedan crear aplicaciones Web”. (Cake, 2012)

CodeIgniter: “Es un framework para el desarrollo de aplicaciones en php, que utiliza el MVC. Esto permite a los programadores o desarrolladores Web mejorar su forma de trabajar, además de dar una mayor velocidad a la hora de crear páginas Webs”. (PINEDA, 2016)

2.1.3.3 Plataformas en JavaScript

Angular: “Es un framework de desarrollo para JavaScript creado por Google. La finalidad de Angular es facilitarnos el desarrollo de aplicaciones web SPA y además darnos herramientas para trabajar con los elementos de una web de una manera más sencilla y óptima”. (Robles, 2017).

ReactJS: “React (también llamada React.js o ReactJS) es una biblioteca JavaScript de código abierto para crear interfaces de usuario con el objetivo de animar al desarrollo de aplicaciones en una sola página. Intenta ayudar a los desarrolladores a construir aplicaciones que usan datos que cambian todo el tiempo. Su objetivo es ser sencillo, declarativo y fácil de combinar”. (campusMVP, 2017)

Vue.js: Vue 2.0 también se presentó en 2016. La idea era tomar prestado lo mejor de Angular, React e incluso EmberJS y convertirlo en una nuevo framework. Está demostrado en numerosas pruebas que es más rápido y ligero en comparación con Angular e incluso con React. Al igual que Angular, Vue.js ofrece enlace a datos en dos direcciones, posibilidad de renderizado en lado del servidor (también ReactJS). Del mismo modo, al igual que ReactJS, utiliza un DOM virtual,

proporciona componentes reactivos y componibles, y se centra mucho en su núcleo, dejando cosas como el enrutado o el estado global en componentes de terceros. En este sentido es menos completo que Angular, pero más que ReactJS. Incluso ofrece soporte para componentes JSX de ReactJS de ser necesario, pudiendo combinar ambas”. (campusMVP, 2017)

Ember.js: “Ya en 2015 Ember era considerado por muchos como el mejor framework JavaScript para aplicaciones web, dejando a React y AngularJS atrás. Hoy en día puede presumir de una gran comunidad online, actualizaciones periódicas y un uso de buenas prácticas de JavaScript que garantizan la mejor experiencia ya por defecto. Ember proporciona también enlace de datos en dos direcciones, manteniendo tanto la vista como el modelo sincronizados todo el tiempo, y tiene integrada la popular sintaxis de Handlebars para crear plantillas. Al igual que Angular ofrece una arquitectura clara para que no sea necesario estar tomando decisiones, lo cual facilita el desarrollo siguiendo buenas prácticas. Está más pensado para crear aplicaciones grandes desde cero que para pequeñas pruebas de concepto, componentes o incorporarlo a aplicaciones existentes.” (campusMVP, 2017)

Aurelia: “Aurelia es como si fuese una variante Angular más elegante y sencilla de entender. Utiliza también TypeScript, por lo que la escritura de código es menos propensa a errores y más sencilla que con JavaScript. Una de las cosas que más les gusta a sus seguidores es la forma de definir componentes, sin el uso de sintaxis rebuscadas, como sí ocurre con Angular y (sobre todo) React. Los componentes escriben en JavaScript y HTML estándar (en realidad en TypeScript y HTML, que luego se generan a JavaScript) y se parecen mucho a las plantillas que se han usado tradicionalmente en desarrollo web. Es muy rápido y ligero, a pesar de no usar un DOM virtual, que no tiene nada que envidiar en rendimiento a React. No depende de componentes de terceros.” (campusMVP, 2017)

2.1.3.4 Plataformas en PYTHON

Pyramid: “Nació de una fusión entre Pylons 1.0 y repoze.bfg. Este framework viene con "pilas incluidas", pero no hace ninguna suposición acerca de los componentes de tu sitio web. La comunidad Pyramid está creciendo rápidamente y cada día son más los desarrolladores que se suman al uso del framework. La documentación es excelente y permite a los desarrolladores avanzar sin tener que contar con el apoyo de la comunidad. Pyramid se esfuerza por ser minimalista, rápido y fiable. (programacion.net, 2018)

Bottle: “Es un microframework muy simple que proporciona un mínimo de herramientas al desarrollador (enrutamiento, plantillas y una pequeña abstracción sobre WSGI). Bottle se puede ejecutar en Python 3”. (programacion.net, 2018)

Django: “Es, con diferencia, el mayor framework web basado en Python. Se apoya en una comunidad grande y activa. Cuenta con una potente interfaz de administración, así como otras muchas características que deberás ir descubriendo tú mismo. Django ofrece formularios model-based, tiene su propio lenguaje de plantillas y cuenta con una excelente documentación que está disponible de manera gratuita”. (programacion.net, 2018)

Flask: “Es un microframework que se creó originalmente como una broma del April Fools Day (como el día de los inocentes en EEUU) que derivó en un framework en solo un único archivo. Se esfuerza por ser simple y pequeña; todo el framework consiste en un puñado de módulos. No hay un esqueleto o una estructura de la cual partir, todo se empieza con una página en blanco. Flask no proporciona grandes funcionalidades, pero, hay extensiones Flask disponibles para agregar ORM, validación de formularios, manejo de carga”. (programacion.net, 2018)

2.1.4 Bases de Datos

MySQL: “Es un sistema de gestión de bases de datos relacional desarrollado bajo licencia dual: Licencia pública general/Licencia comercial por Oracle Corporation y está considerada como la base de datos de código abierto más popular del mundo, y una de las más populares en general junto a Oracle y Microsoft SQL Server, sobre todo para entornos de desarrollo web. MySQL fue inicialmente desarrollado por MySQL AB (empresa fundada por David Axmark, Allan Larsson y Michael Widenius). MySQL AB fue adquirida por Sun Microsystems en 2008, y ésta a su vez fue comprada por Oracle Corporation en 2010, la cual ya era dueña desde 2005 de Innobase Oy, empresa finlandesa desarrolladora del motor InnoDB para MySQL”. (oracle, 2015)

PostgreSQL: “Es un potente sistema de base de datos relacional de objetos de código abierto que utiliza y amplía el lenguaje SQL combinado con muchas características que almacenan y escalan de forma segura las cargas de trabajo de datos más complicadas. Los orígenes de PostgreSQL se remontan a 1986 como parte del proyecto POSTGRES en la Universidad de California en Berkeley y tiene más de 30 años de desarrollo activo en la plataforma central. PostgreSQL se ha ganado una sólida reputación por su arquitectura probada, confiabilidad, integridad de datos, conjunto de características sólidas, extensibilidad y la dedicación de la comunidad de código abierto detrás del software para ofrecer constantemente soluciones innovadoras y de alto rendimiento. PostgreSQL se ejecuta en todos los sistemas operativos principales, ha sido compatible con ACID desde 2001, y tiene complementos poderosos como el popular extensor de base de datos geoespacial PostGIS. No es sorprendente que PostgreSQL se haya convertido en la base de datos relacional de código abierto elegida por muchas personas y organizaciones”. (Group, 2019)

MSSQL: “Cuando se trata de los sistemas que elige para administrar los datos, se recomienda que el rendimiento y la seguridad no sean un impedimento para la administración del negocio. Microsoft, como líder del sector en cuanto a sistemas de administración de bases de datos operativas (ODBMS), mejora constantemente su oferta para ayudarlo a aprovechar al máximo el mundo de los datos que está en constante expansión”. (Microsoft, 2019)

SQLite: “Es una biblioteca en lenguaje C que implementa un motor de base de datos SQL pequeño, rápido, autónomo, de alta confiabilidad y con todas las funciones. SQLite es el motor de base de datos más utilizado en el mundo. SQLite está integrado en todos los teléfonos móviles y en la mayoría de las computadoras, y se incluye en innumerables aplicaciones que las personas usan todos los días. El formato de archivo SQLite es estable, multiplataforma y compatible con versiones anteriores, y los desarrolladores se comprometen a mantenerlo así durante al menos el año 2050. Los archivos de base de datos SQLite se usan comúnmente como contenedores para transferir contenido rico entre sistemas y como un formato de archivo a largo plazo para datos. Hay más de 1 billón de bases de datos SQLite en uso activo”. (sqlite, 2019)

Microsoft Acces: “Es un sistema manejador de base de datos. Una base de datos permite manipular información organizada y optimizada para reducir el espacio que ocupan los datos y permiten la consulta ágil y eficiente de la información. En el caso de Access, es un manejador de bases de datos relacional. La información se organiza en tablas. Entre las tablas se establecen relaciones que permiten unir tablas. Es posible crear consultas sencillas o complejas para ver información. Con Microsoft Access se puede crear una base de datos, modificar su estructura, actualizar la información presente. Además, puede hacer consultas muy eficientes que permiten acceder a la información que cumple con determinadas condiciones”. (misapuntessistemas, 2012)

2.1.5 Estrategias De Programación

Fuerza bruta: “Los algoritmos de fuerza bruta resuelven el problema con la estrategia más obvia de solución, que no siempre es la mejor según el número de operaciones que se requiere”. (Aleman, 2014)

Divide and conquer (divide y reinarás): “Esta metodología divide las instancias del problema a resolver en instancias cada vez más pequeñas, usualmente en forma recursiva, hasta llegar a una instancia en que el problema es resoluble en forma trivial o con unas pocas instrucciones. Los algoritmos de búsqueda binaria son un ejemplo de la metodología divide and conquer”. (Aleman, 2014)

Programación dinámica: “Cuando un problema presenta una sub-estructura óptima o lo que es lo mismo, cuando la solución óptima de un problema se obtiene a partir de las soluciones óptimas de sus sub-problemas, se encuentra la solución resolviendo primero los sub-problemas más sencillos y luego utilizando esas sub-soluciones para resolver problemas incrementalmente difíciles. Por ejemplo, si se tiene una serie de puntos (definidos por coordenadas x , y) que delimitan una región, y se necesita saber si otro punto se encuentra dentro o fuera de esa región, una forma de resolver el problema consiste en comenzar formando cuadrados con puntos contiguos, para luego formar figuras cada vez más grandes y, por cada figura, determinar si el punto está dentro o fuera de ella. En cada paso se aprovecha la información de los pasos anteriores, hasta que, al completar todos los puntos, se obtiene el resultado del problema”. (Aleman, 2014)

Programación lineal: “Para resolver un problema utilizando programación lineal, se plantea una serie de inecuaciones y luego se busca maximizar (o minimizar) las variables, respetando las inecuaciones. Muchos problemas pueden plantearse en la forma de un conjunto de inecuaciones, para luego resolverlos utilizando un algoritmo genérico, como, por ejemplo, el denominado método Simplex”. (Aleman, 2014)

Búsqueda y enumeración: “Muchos problemas (como, por ejemplo, un juego de ajedrez) pueden modelarse con grafos y resolverse a partir de un algoritmo de exploración del grafo. Tal algoritmo especificará las reglas para moverse en el grafo en busca de la solución al problema. Esta categoría incluye también algoritmos de backtracking (vuelta atrás), los cuales van ensayando distintos caminos con posibles soluciones y vuelven atrás cuando no las encuentran. Por ejemplo, para encontrar la salida en un laberinto, cada vez que se llega al final de un camino se vuelve atrás hasta la última bifurcación, para probar con las distintas alternativas de esa bifurcación”. (Aleman, 2014)

Algoritmos heurísticos: “el propósito de estos algoritmos no es necesariamente encontrar una solución final al problema, sino encontrar una solución aproximada cuando el tiempo o los recursos necesarios para encontrar la solución perfecta son excesivos. Muchos sistemas antivirus utilizan métodos heurísticos para detectar conductas de programas que podrían estar actuando en forma maliciosa”. (Aleman, 2014)

Algoritmos voraces: “seleccionan la opción de solución (solución local) que tenga un costo menor en la etapa de solución en la que se encuentran, sin considerar si esa opción es parte de una solución óptima para el problema completo (solución global)”. (Aleman, 2014)

2.2 Estado del Arte

Título: algoritmo para gestión de horarios de la facultad de ingeniería de la universidad de cuenca.

Autores: Sonia Edith Barreto Barros y Luis Alberto López Villavicencio.

Lugar: Cuenca-Ecuador

Link: <http://dspace.ucuenca.edu.ec/bitstream/123456789/4538/1/Tesis.pdf>

“El presente trabajo de investigación está enfocado en el tema timetabling o generación de horarios de clases. Para esta tesis se revisaron diferentes métodos que han sido utilizados en la resolución de esta problemática dentro del área de investigación de operaciones e inteligencia artificial. Se ha tomado como base el método BINGO para ser mejorado y adaptado a las necesidades de la Facultad de Ingeniería de la Universidad de Cuenca, trabajos que entregaron como resultado un nuevo método al que se lo ha denominado SOLU. El método propuesto considera la disponibilidad de los profesores, los requerimientos de la facultad, así como la disponibilidad de aulas y laboratorios. Una vez implementado el método, éste fue sometido a diferentes pruebas verificando su correcto funcionamiento y comportamiento, estudio que incluyó una prueba con datos reales. Los resultados obtenidos de las pruebas evidenciaron el cumplimiento de cada una de las restricciones, evitando de esta manera cruces de profesores, materias, paralelos y aulas o laboratorios. Finalmente, en la creación del algoritmo se logró reducir el tiempo empleado para la asignación de horarios, brindando así un mejor servicio a la facultad.” (Villavicencio, 2013), Esta tesis se revisó por la similitud del uso de algoritmos no-polinomiales que plantea para la organización del horario de la facultad de ingeniería.

Título: Combinación entre Algoritmos Genéticos y Aleatorios para la Programación de Horarios de Clases basado en Ritmos Cognitivos.

Autor: Omar D. Castrillón.

Lugar: Manizales-Colombia.

Link:https://scielo.conicyt.cl/scielo.php?pid=S071807642014000400008&script=sci_arttex

“Se ha diseñado un método basado en algoritmos evolutivos (genéticos y aleatorios) para programar los horarios de clases en una universidad. Esta metodología considera los ritmos cognitivos de los estudiantes que indican que es mejor enseñar algunas asignaturas en intervalos específicos de tiempo. Primero se describen las diferentes técnicas empleadas para desarrollar este problema. Luego se propone una nueva metodología basada en ritmos cognitivos y algoritmos evolutivos, para resolver todas las restricciones duras y blandas del problema. Finalmente se comparan diferentes metodologías para establecer cuál es la más eficiente. Se concluye que los algoritmos evolutivos son más eficientes que otras técnicas empleadas (19.5%) en la programación de horarios universitarios.” (Castrillón, 2014), Este artículo facilita la selección del algoritmo que sea el más adecuado para que sea el encargado de organizar los horarios.

Título: planificación de horarios del personal de cirugía de un hospital Del estado Aplicando algoritmos genéticos (time tablingProblem)

Autores: Gissella María Bejarano Nicho.

Lugar: Lima-Perú

Link: <http://tesis.pucp.edu.pe/repositorio/handle/123456789/551>

“El presente documento describe un proyecto de fin de carrera en Ciencias de la Computación. Este proyecto intenta dar solución al problema de generación de horarios del personal de un hospital, para ser más exactos del servicio de Cirugía y Radioterapia de un hospital del estado. La solución se construye con el uso de un algoritmo genético. Se ha tomado como caso de estudio el servicio de un

hospital del país, el cual se visitó en repetidas ocasiones para el respectivo levantamiento de información y adaptación de un algoritmo que cumpla con sus restricciones y requerimientos. Para facilitar la búsqueda de esta solución se aplicará los operadores de casamiento y mutación, especialmente pensados para la estructura del cromosoma o individuo. La calidad de las soluciones generadas por el algoritmo se medirá en base a la cantidad de preferencias cumplidas (condiciones débiles, dado que las restricciones y requerimientos son condiciones obligatorias que se cumplen en cada solución. Para determinar los valores de los parámetros del algoritmo se realizaron varias corridas con diferentes combinaciones de valores y se eligió la que optimizaba la función objetivo de la solución. Se estima que la duración del proyecto será de un año.” (Nicho, 2010), Aunque el proyecto no se basa en un horario para una institución educativa , si trata el problema de la asignación de horarios.

Título: Algoritmos Evolutivos Aplicados a la Generación De Horarios para el Colegio Aplicación de la Una - Puno

Autores: Santos Hañari Mamani

Lugar: Puno-Perú

Link: <http://repositorio.unap.edu.pe/handle/UNAP/3008>

“El presente trabajo, tiene como objetivo encontrar una solución basado en algoritmos evolutivos, capaz de resolver de manera automatizada la generación de horarios con una base científica y metodológica, contribuyendo de esta forma a minimizar el tiempo y así reemplazar los papeles por sistemas automatizados que puedan ahorrar tiempo y recursos. Para la generación horaria al inicio de cada periodo académico es necesario asignar y coordinar los recursos, así como también organizar y distribuir los horarios de los profesores teniendo en cuenta condiciones particulares y otros factores como la disponibilidad de los profesores. Los algoritmos evolutivos son métodos adaptativos que pueden usarse para resolver problemas de búsqueda y optimización, son capaces de ir creando soluciones para problemas del mundo real. La evolución de dichas soluciones

hacia valores óptimos del problema depende de una tasa de mutación de 0.005 y de un elitismo de 7 mejores individuos. Para la implementación del Algoritmo Evolutivo en el problema planteado, se siguieron ciertos pasos como es la selección de los mejores individuos por el método de la ruleta, se aplicó una autoadaptación del operador de mutación a través de cuatro diferentes operadores de mutación por intercambio, por inversión, por inserción y por mezcla. En el proceso el algoritmo evolutivo solo tardó unos 35 minutos para encontrar la mejor solución al problema y generar el horario escolar.” (MAMANI, 2016), este proyecto utiliza un algoritmo inteligente pero fue descartado puesto que su tiempo de ejecución está en más o menos 35 minutos aunque se basa en el uso de un algoritmo genético.

Título: Elaboración De Una Solución Meta heurística Usando Un Algoritmo Genético Que Permita Elaborar La Distribución De Los Horarios Académicos

Autores: Ana Nataly Angeles Diaz

Lugar: Lima-Perú

Link: <http://tesis.pucp.edu.pe/repositorio/handle/123456789/6057>

“El presente documento describe un proyecto de fin de carrera en Ciencias de la Computación. Este proyecto intenta dar solución al problema de generación de horarios académicos en instituciones de nivel superior. La solución se construye con el uso de un algoritmo genético a partir de una población inicial generada por un algoritmo Grasp fase construcción. Se ha tomado como caso de estudio a la facultad de Ciencia e Ingeniería de la Pontificia Universidad Católica del Perú, en la cual se contó con el apoyo del encargado de realizar el horario de la especialidad de ingeniería informática para el respectivo levantamiento de información, con lo cual se consiguió la adaptación de un algoritmo que cumpla con sus restricciones y requerimientos. Para facilitar la búsqueda de esta solución se aplicarán los operadores de selección, casamiento, mutación y elitismo. La calidad de las soluciones, generadas por el algoritmo, se medirá en base a la cantidad de restricciones cumplidas. Para determinar los valores de los parámetros de los algoritmos se realizaron varias ejecuciones con diferentes combinaciones de

valores y se optó por la que optimizaba la función objetivo de la solución. Se estima que la duración del proyecto será de un año”. (Diaz, 2015), esta es solo una propuesta y en su desarrollo plantea una función para la solución de la asignación de horarios.

Título: Un Sistema De Generación De Horarios Para La Enseñanza De Pregrado En Universidades Peruanas Mediante Algoritmos Genéticos

Autores: Sherly Patricia Blaz Aristo

Lugar: Lima-Perú

Link: <https://core.ac.uk/download/pdf/54236369.pdf>

“Este proyecto intenta dar solución al problema de generación de horarios para los cursos universitarios de pregrado en las Universidades Peruanas, en la cual cada encargado de la elaboración de horarios debe realizar una laboriosa tarea de asignación de salones y docentes en un periodo específico para los dictados de clases de las asignaturas que se imparten en un ciclo académico. La presente investigación es un esfuerzo para facilitar dicha labor manual, ya que el procedimiento actual en muchas Universidades del Perú requiere muchos recursos y un gran esfuerzo de parte de los responsables, además de generar horarios no factibles y no aprovechar de manera óptima los recursos de la institución. En este trabajo se propone un sistema inteligente de generación de horarios basado en algoritmos genéticos, el cual fue adaptado para poder cumplir con los requisitos específicos impuestos por cada Facultad de las diferentes Universidades del Perú y así satisfacer a los usuarios involucrados. En nuestras pruebas se consideró como caso de estudio la Facultad de Ingeniería de Sistemas de la Universidad Nacional Mayor de San Marcos, obteniendo como resultado un horario con cero violaciones de restricciones obligatorias y reduciendo en lo más posible las violaciones de las restricciones blandas y así superando la generación de horarios elaborado de forma manual que actualmente se lleva a cabo.” (Aristo, 2016), este proyecto se basa en el uso de un algoritmo genético aunque su uso es teórico no se realiza una solución práctica.

Título: Revisión de Algoritmos Genéticos Aplicados al Problema de la Programación de Cursos Universitarios

Autores: Mireya Flores Pichardo

Lugar: Cuernavaca Morelos, MÉXICO

Link:<http://riaa.uaem.mx/xmlui/bitstream/handle/20.500.12055/93/progmat312011Revision.pdf?sequence=1&isAllowed=y>

“La programación de horarios académicos es un problema particular que se encuentra dentro del problema general de asignación de recursos. Este problema de horarios, se conoce en la comunidad científica como Problema de Programación de Horarios Universitarios. Los problemas de programación de horarios consisten en generar horarios para tareas definidas, buscando cumplir de la mejor manera con condiciones o requerimientos específicos. Este problema ha sido tratado con diferentes métodos, por ejemplo, Colonia de Hormigas, Búsqueda Tabú, Coloreo de grafos y Algoritmos Genéticos. En éste trabajo se hace una revisión de algunos algoritmos evolutivos que han abordado el problema de horarios académicos aplicando diferentes modelos.” (Pichardo, 2011), este proyecto usa el planteamiento de un algoritmo genético pero solo como referencia para su estudio en la aplicación a un horario universitario.

Título: Programación de Horarios Escolares basados en Ritmos Cognitivos usando un Algoritmo Genético de Clasificación No-dominada, NSGA-II

Autores: Víctor F. Suárez, Álvaro Guerrero y Omar D. Castrillón*

Lugar: Manizales-Colombia

Link:https://scielo.conicyt.cl/scielo.php?pid=S0718-642013000100012&script=sci_arttext

“Se plantea la solución del problema de programación óptima de horarios escolares de una escuela pública colombiana. Esto se hace considerando no solo la asignación adecuada de salones y docentes sino también los ritmos cognitivos que presentan los estudiantes como el factor más importante en el proceso de optimización. Se propone como método de solución el Algoritmo Genético de

Clasificación No-dominada, NSGA-II. Los resultados muestran una mayor eficiencia del algoritmo en comparación con otros aplicados al mismo problema y evaluados en la misma forma. A nivel experimental, la metodología evidencia que los grupos programados mediante la orientación descrita presentan una reducción en los niveles de mortalidad académica en comparación con una programación horaria que no consideran los ritmos cognitivos de los estudiantes”. (Víctor F. Suárez, 2012) Esta propuesta solo se basa en la teoría pero es una referencia valida en su aplicación a los horarios Universitarios y a revisar el problema de la deserción estudiantil

Título: Generación de Horarios Académicos en INACAP Utilizando Algoritmos Genéticos

Autores: Jorge Andrés Ahumada Ahumada

Lugar: Santiago de Chile

Link: <http://repositorio.uchile.cl/bitstream/handle/2250/131197/Generaci%F3n-de-horarios-acad%E9micos-en-INACAP-utilizando-algoritmos-gen%E9ticos.pdf?sequence=1>

“En todas las instituciones de educación, el proceso de creación de un horario académico es un desafío que se debe sortear semestre a semestre. Este proceso no es simple, ya que está sujeto a restricciones físicas, reglamentarias y legales entre otras. El objetivo de esta investigación es ofrecer una alternativa de solución para el problema de la asignación de horarios y salas en INACAP, una de las instituciones de educación de mayor envergadura en cuanto a cantidad de alumnos y a infraestructura. En este trabajo se revisan algunos de los métodos más utilizados para resolver estos problemas de tipo timetabling, un problema considerado de tipo NP-completo, entre los cuales se encuentra la programación lineal y los algoritmos genéticos. Los algoritmos genéticos, como varios otros métodos, se basan en procesos que se encuentran presentes en la naturaleza. Son técnicas de optimización que emulan de cierta forma los conceptos de la evolución como la supervivencia y la reproducción postulados por Charles Darwin.

En este documento se muestra el diseño y la implementación de un algoritmo genético que logra resolver el problema planteado en tiempos muy razonables y con una muy buena calidad de las soluciones reflejada en el bajo porcentaje de choques horarios. Se concluye que los algoritmos genéticos son una alternativa muy rápida y confiable para los problemas de optimización” (AHUMADA, 2014), este proyecto usa un algoritmo genético y trata de realizar una asignación basándose en los problemas de timetabling que incluyen a los estudiantes.

Título: Diseño de Horarios para Sincronizar Líneas de Buses en Múltiples Periodos de Planificación

Autores: O.J Ibarra-Rojas, Y.A. Rios-Solis

Lugar: Santiago de Chile

Link: <http://www.ingenieriadetransporte.org/index.php/sochitran/article/view/155/92>

“Presentamos el problema de Diseño de Horarios con Sincronización en Múltiples Periodos para determinar el tiempo de salida de un conjunto de viajes, maximizando el número de sincronizaciones que permiten una transferencia de pasajeros con tiempos de espera adecuados y evitan congestión de autobuses en algunos nodos de la red de transporte. Debe cumplirse regularidad de servicios en cada periodo de planificación y transiciones suaves entre diferentes periodos. Diseñamos una formulación y una Búsqueda de Vecindario Variable basada en operadores que usan propagación de restricciones. Obtenemos soluciones de calidad para instancias grandes y mostramos el beneficio de implementar el enfoque multi-periodo.” (O.J Ibarra-Rojas, 2014), este proyecto trata de organizar los horarios en una terminal de transportes no se parecen pero manejan un problema parecido que son las tablas de llegadas entre otras.

Título: Aplicación de Programación Lineal para la Asignación de Horarios en una Institución Educativa Mexicana

Autores: Adriana Canseco González, Diana Sánchez Partida, Catya Zúñiga Alcaraz, Elías Olivares-Benítez

Lugar: Guadalajara, México

Link: <http://revistas.ubiobio.cl/index.php/RI/article/view/2779>

“Un problema recurrente en cada inicio de periodo escolar en las Instituciones Educativas (IEs) es la calendarización de horarios de los profesores para un conjunto de grupos y cursos. En este trabajo se pretende optimizar la asignación de cursos a grupos en un conjunto de periodos de tiempo con determinados requerimientos. Se aborda este problema dentro de un plantel de educación técnica media, en el cual se analizan tres grupos de la especialidad de Laboratorista químico ofertada por la institución. Se diseñó un modelo de programación matemática que fue resuelto mediante la técnica de ramificación y acotamiento disponible en un software comercial de optimización llamado LINGO® 10. Este caso real fue implementado en el ciclo 2013 – 2014 y validado con éxito al resolver en 4 segundos las actividades que realizarán los empleados dentro de una franja horaria de entre 5 a 7 horas. El enfoque planteado contribuye a agilizar la toma de decisiones del centro escolar al inicio del ciclo académico.” (Adriana Canseco-González, 2016), Este proyecto se basa en el desarrollo de un modelo de programación lineal aplicado a un software ya existente dio como resultado la asignación de sus horarios en 4 segundos, sin embargo se pudo evidenciar en la lectura que las variables tenidas en cuenta fueron docentes y aulas.

Título: Programación de Horarios de Clases y Asignación de Salas para la Facultad de Ingeniería de la Universidad Diego Portales Mediante un Enfoque de Programación Entera

Autores: Rodrigo Hernández, Jaime Miranda P., Pablo A. Rey

Lugar: Santiago de Chile

Link: www.dii.uchile.cl/~ris%20/RISXXII/horariosUDP_RISVersion%20FINAL.pdf

“Un aspecto importante en la gestión académica de las universidades es la generación de horarios y la asignación de salas de clase para los distintos cursos que realizan. En este artículo se presenta un modelo de programación entera el cual decide simultáneamente los horarios de los cursos y la asignación de salas. Las variables utilizadas están asociadas a la definición del horario del curso para una semana por medio de un patrón horario. Una particularidad del modelo es que tanto las condiciones sobre capacidad y tipo de salas de clase, así como las combinaciones de bloques horarios para un curso, son manejadas implícitamente mediante las variables de decisión. Se reportan los resultados de la comparación de la programación obtenida con el modelo propuesto y la programación que efectivamente se utilizó. El modelo propuesto entrega de manera rápida y eficiente los horarios y asignaciones de sala de clase satisfaciendo todos los requerimientos obligatorios y condiciones deseables para la Facultad en un tiempo menor a los 5 minutos” (Rodrigo Hernandez, 2008), aunque utiliza un algoritmo basado en patrones obtenidos de modelos previamente asignados para poder tener una referencia al momento de seleccionar el horario adecuado para asignar salas de clase y cursos.

Título: Modelo matemático para la programación de horarios y asignación de aulas en la Universidad de las Fuerzas Armadas ESPE

Autores: Hidalgo Zurita, Martha Ximena Vásconez Espinoza, Cumandá del Rocío

Lugar: Sangolqui, Ecuador

Link: <http://repositorio.espe.edu.ec/handle/21000/15054>

“La generación de horarios y asignación de aulas en Instituciones Educativas se presenta al inicio de cada período académico y conlleva un trabajo complejo puesto que intervienen muchas restricciones y limitaciones que deben ser consideradas de acuerdo a la situación particular de cada Institución. La Universidad de las Fuerzas Armadas - ESPE no es ajena a este tipo de problemas, actualmente esta tarea se realiza en forma manual en cada uno de los Departamentos y los encargados, generalmente, son el planificador y los coordinadores del área, lo que genera inconvenientes pues no suele existir una adecuada coordinación entre ellos, ocasionando, que puedan existir cruces de horarios y aulas, esto repercute de manera significativa en el aprovechamiento de los recursos físicos y humanos de la Institución, influyendo negativamente en el desempeño de estudiantes y docentes. Por lo expuesto, este trabajo de investigación propone una solución al problema de programación de horarios y la distribución adecuada de las aulas de clase, para los diferentes grupos de cada una de las carreras de pregrado de la Universidad de las Fuerzas Armadas - ESPE, matriz, modalidad presencial, para lo cual se desarrolla una heurística que lleva a una solución técnica factible. Se propone una metodología, basada en técnicas como, programación lineal entera, búsqueda tabú y gestión del conocimiento, que cumple con las necesidades y requerimientos de la Institución.” (Martha Ximena Hidalgo Zurita, 2018), este proyecto presenta una solución novedosa integrando 3 técnicas de programación para buscar el horario ideal.

Título: Aplicación de búsquedas Tabú y Grasp para la resolución de un problema de calendarización de horarios de clase en universidades

Autores: Marco Vinicio Añazco Maldonado, Howard Lizardo Chávez Antón

Lugar: Quito, Ecuador

Link: <http://bibdigital.epn.edu.ec/handle/15000/16682>

“Esta investigación tiene como finalidad encontrar una solución adecuada al problema de programación de horarios de clases en un tiempo razonable, en la Carrera de Ingeniería en Desarrollo de Negocios Bilingüe (Ingeniería de Emprendedores) de la Facultad de Especialidades Empresariales de la Universidad Católica de Santiago de Guayaquil (UCSG), mediante las metaheurísticas BúsquedaTabú y GRASP. Originalmente, la estructura de horarios de clases, se realiza de forma manual por parte de la coordinación de la carrera, trabajo que toma aproximadamente dos semanas hasta encontrar un horario que satisfaga las condiciones requeridas para la ejecución del mismo. Debido a los tiempos excesivos de estructuración de horarios y el esfuerzo adicional del recurso humano, se decide plantear algoritmos metaheurísticos que permitan encontrar una solución tomando en consideración todos los requerimientos de la carrera. En este trabajo de tesis, encontramos soluciones mediante la Programación Lineal Entera y las metaheurísticas Tabú Search y GRASP. Además, se realiza una comparación de los resultados obtenidos en cada caso.” (Marco Vinicio Añazco Maldonado, 2016), En este proyecto podemos apreciar el uso de varias técnicas de programación y una comparación que se hace para verificar el mejor horario encontrado por cada técnica.

Título: Diseño e implementación de un algoritmo matemático basado en optimización para la generación de horarios de clases en la Universidad Politécnica Salesiana

Autores: Pedro Javier Cornejo Reyes

Lugar: Cuenca, Ecuador

Link: <https://dspace.ups.edu.ec/handle/123456789/16121>

“La presente investigación refleja el resultado de la aplicación de técnicas de programación lineal entera y algoritmos genéticos para el desarrollo e implementación de horarios de clases para la carrera de Ingeniería de Sistemas de la Universidad Politécnica Salesiana. En el presente trabajo se ha realizado el modelamiento e implementación de un modelo matemático de horarios de clases mediante el levantamiento y aplicación de requerimientos por parte de la institución antes mencionada. Esto último permitió formular restricciones, pues fueron indispensables para delimitación del modelo.

Las conclusiones y resultados de este trabajo pueden ser utilizadas para procesos futuros en los cuales se requiera mejorar e implementar el modelo para asignación de horarios en las distintas carreras de la Universidad Politecnica Salesiana” (Reyes, 2018) , Este proyecto plante al uso de un modelo matemática , además queda como base para futuros proyectos que desearan aplicarlo.

Título: Sistema de asignación de horarios académicos para la Pontificia Universidad Católica Del Ecuador Sede Esmeraldas (PUCESE).

Autores: Sergio Wladimir Paredes Salazar,

Lugar: Esmeraldas, Ecuador

Link:

repositorio.pucese.edu.ec/bitstream/123456789/612/1/PAREDES%20SALAZAR%20SERGIO%20WLADIMIR.pdf

“La generación de horarios académicos es un proceso metódico que se lleva a cabo en la mayoría de institutos educacionales del mundo, ya sean centros universitarios o de estudios básicos. Para este proceso se pueden emplear diferentes métodos de desarrollo, como lo es el caso de herramientas tecnológicas que se han estandarizado por todo el mundo o como lo es el caso de herramientas manuales.

Desde la primera creación de horarios en el mundo, los centros educativos han optimizado muchos recursos, como es el caso del tiempo, pero, sin embargo, siempre suelen traer problemas durante su desarrollo, como, por ejemplo; cruce de horas de docentes, problemas para asignar aulas; por lo que conlleva a una reestructuración de los mismos. La propuesta que se presenta en este proyecto es un sistema estandarizado que asigne horarios de clases, aulas y laboratorios en la PUCESE, cuyos principales beneficiarios son los directores de escuela y el director de currículo.” (Salazar, 2016), En esta propuesta se pudo interpretar que se utiliza una herramienta tecnológica que ha sido estandarizado para el uso en la creación de horarios.

Título: Sistema de gestión de horarios académicos para la Universidad Central del Ecuador

Autores: René Alfonso Carrillo Flores, Hernán Oswaldo Campoverde Ramos

Lugar: Quito, Ecuador

Link: <http://www.dspace.uce.edu.ec/handle/25000/5171>

“El presente proyecto de tesis que aquí se presenta, tiene la finalidad de automatizar los procesos que se lleven a cabo dentro de la Facultad de Ciencias Físicas y Matemática para la generación de sus horarios académicos y los docentes puedan impartir sus cátedras en las horas correspondientes, permitiendo agilizar el proceso de generación de horarios y que estén disponibles en cualquier momento que se lo requiera tanto para el docente como para el alumnado por medio de este sistema web. El proyecto se basa en la construcción de un aplicativo web en Java, bajo la especificación JEE6, Servidor Glassfish 4.1, Base de Datos PostgreSQL, se especifica un esquema completo del sistema en cada una de sus etapas de construcción, se ha realizado las correspondientes pruebas necesarias previas a la puesta en producción del aplicativo.” (René Alfonso Carrillo Flores, 2015), Aunque este proyecto no revela algoritmos usados en el desarrollo de su aplicativo, si se revela que lenguaje, motor y plataforma web.

Título: Sistema experto para la generación automática de horarios académicos usando bases del conocimiento

Autores: Jacqueline Alexandra Mena Fonseca, Edison Fabián De la Cruz Cando

Lugar: Sangolqui, Ecuador

Link: <http://repositorio.espe.edu.ec/handle/21000/11211>

“En la actualidad la labor educativa que se desarrolla durante la planificación de cada nuevo periodo académico ha requerido que para esta actividad se asigne más recursos humanos, económicos. Motivo por el cual es de primordial importancia adecuar los medios disponibles y buscar alternativas, con el sentido

de optimizar los recursos y agilizar estas actividades para de esta forma obtener mejores resultados en tiempos óptimos para una toma oportuna de decisiones con la información obtenida. Hoy en día los diferentes centros educativos, es muy común que coordinadores, directores realicen los horarios académicos usando herramientas las cuales no brindan las facilidades que se requieren para esta actividad. Por lo que un director o coordinador, necesita de medios que le permitan organizar los horarios académicos de una manera rápida y eficiente, sin requerir con alto número de recursos para realizar esta actividad. Actualmente existen varias alternativas que si bien es cierto organizan los horarios académicos.” (Jacqueline Alexandra Mena Fonseca, 2015) este proyecto utiliza la minería de datos como base para la realización de la asignación de los horarios.

Título: Un sistema experto basado en minería de datos y programación entera lineal para soporte en la asignación de materias y diseño de horarios en educación superior

Autores: Daniel Calle López, Javier Cornejo Reyes, Fernando Pesántez Avilés, Mónica Rodas Tobar, César Vásquez Vásquez, Vladimir Robles Bykbaev

Lugar: Quito, Ecuador

Link: scielo.senescyt.gob.ec/scielo.php?script=sci_arttext&pid=S139065422018000100102&lng=en&nrm=iso&tlng=en

“En general, varias de las organizaciones tienden a manejar la asignación de cargas horarias de sus empleados de manera tradicional (imposición de una jornada de trabajo de 8). Todo esto a fin de mantener cierto control sobre variables como asistencia y ausentismo del personal e incluso en algunos casos considerando aquello como sinónimo de eficacia laboral. Sin embargo, la nueva tendencia organizacional ha roto ciertos paradigmas con orientación mecanicista, pretendiendo crear un nuevo horizonte hacia la construcción de organizaciones orgánicas y dinámicas. Por lo expuesto, en este artículo se describe una propuesta de sistema experto basado en programación entera lineal y minería de datos para abordar el problema de asignación de materias y el diseño de horarios

(considerado un problema NP-completo). Los resultados preliminares son alentadores, ya que han permitido realizar la asignación de materias en una base de 133.000 registros de docentes y generar horarios con un mínimo de coste computacional.” (Daniel Calle López, 2018), en este algoritmo utilizado para la universidad de Quito también se valida la tasa de asistencia y el ausentismo de los docentes.

Título: Análisis y propuesta de un procedimiento para la gestión de horarios del curso académico en la Facultad de ADE de la UPV

Autores: Luis Alejandro López Balboa

Lugar: Valencia, España

Link: <https://riunet.upv.es/handle/10251/86742>

“El objetivo principal de este proyecto es cambiar nuestro sistema de gestión de horarios hacia uno más actualizado y homogéneo en la asignación de los horarios entre las distintas asignaturas del curso académico, que reduzca los tiempos de elaboración y sean adaptables a los alumnos y a los profesores. Este trabajo se realiza debido a que nuestro sistema para elaborar los horarios para los cursos académicos es bastante complejo y supone un gasto de tiempo excesivo, además de que los tiempos establecidos para las diferentes asignaturas (Teoría de Aula, Práctica de Aula, Seminarios, Prácticas, etc.) no son homogéneos en comparación a otras Escuelas de la Universidad Politécnica de Valencia (UPV) que tienen unos horarios más uniformes. Por ejemplo, en algunas Escuelas todas las asignaturas de 6 créditos se dividen entre 3 de teoría y 3 de prácticas. Esto les permite crear “casillas” de 90 minutos para todas las asignaturas. Como propuesta por parte de la Vicedecana-Jefa de Estudios de la Facultad de Administración y Dirección de Empresas (ADE), María de Miguel Molina, nos disponemos a mejorar este sistema de elaboración de horarios de la Facultad de ADE para los cursos académicos futuros. Se analizará el procedimiento actual y, en base a la información recopilada de diferentes fuentes, se propondrán las mejoras necesarias para diseñar un nuevo método para la gestión de horarios, a ser posible con soporte

informático en función del presupuesto necesario. Realizaremos cuestionarios a diferentes Escuelas de la UPV para que nos expliquen cómo funcionan sus sistemas o qué tipo de herramientas utilizan, así como aconsejarnos sobre cómo deberíamos de enfocar el nuestro.” (Luis Alejandro López Balboa, 2017), este proyecto se basa en un procedimiento a tener en cuenta para la generación de un horario.

Título: Implantación de un programa de planificación de horarios en la Escuela de Ingenierías Industriales

Autores: Miguel Fernández Petra

Lugar: Valladolid, España

Link: <http://uvadoc.uva.es/handle/10324/20790>

“La planificación de horarios se realiza de manera anual en la Escuela de Ingenierías Industriales de la Universidad de Valladolid. En este proceso de planificación se tienen que tener en cuenta diferentes restricciones, aunque la más importante es que el horario generado no contenga incompatibilidades para los grupos de alumnos, profesores o aulas. Este proceso se realiza actualmente de manera manual, por lo que conlleva una cantidad de tiempo y trabajo asociada a él muy elevada, ya que hay que ir realizando iteraciones para conseguir cumplir con todos los requisitos que tiene la Escuela. El objetivo de este trabajo es, mediante la búsqueda, estudio, selección y adaptación de un programa de gestión de horarios, reducir el tiempo empleado en esta planificación, consiguiendo además simplificar la misma y facilitar la posibilidad de optimizar la utilización de los recursos disponibles.” (Petra, 2016), este proyecto parte de aplicaciones ya existentes e intenta adaptar la que se ha seleccionado para el uso de la escuela de ingenierías industriales.

Título: Acompañamiento Al Desarrollo Del Software Para El Registro De Horarios
En Los Diferentes Programas De La Universidad De Los Llanos

Autores: Edgar Leonardo Velásquez Villar

Lugar: Villavicencio, Colombia

Link:

<http://repositorio.unillanos.edu.co/jspui/bitstream/001/1105/1/RUNILLANOS%20SIS%200290P%20ACOMPA%20C3%291AMIENTO%20AL%20DESARROLLO%20DEL%20SOFTWARE%20PARA%20EL%20REGISTRO%20DE%20HORARIOS%20EN%20LOS%20DIFERENTES%20PROGRAMAS%20DE%20LA%20UNIVERSIDAD%20DE%20LOS%20LLANOS%20.pdf>

“Se presentó un software para la creación de los horarios que, mediante la integración informática, brindaba mejoras al proceso de creación, planificación y divulgación de los horarios; una solución tecnológica a medida de la problemática existente en la Universidad de los Llanos. El sistema para el registro de horarios en los diferentes programas de la Universidad de los Llanos, fue una iniciativa que brindó una solución integral a las necesidades de planear, coordinar y divulgar los horarios, mejorando los servicios prestados a los estudiantes; la Universidad de los Llanos contaba con un sistema que le permitía crear y publicar los horarios de todas las facultades, pero su ejecución era compleja y centralizada en la Oficina de Admisiones, el gran volumen de trabajo que allí converge ocasionaba retrasos y que la información no fuera publicada a tiempo. La mayoría de los programas construían los horarios utilizando herramientas de oficina y luego delegaban la digitalización de la información a la Oficina de Admisiones, trabajo sumamente dispendioso ya que se debían revisar todos los horarios antes de que fueran publicados en la página web de la Universidad de los Llanos. La información de los horarios es de vital importancia para poder iniciar el proceso de inscripción de materias cada periodo académico. Este software fue presentado como reemplazo total a la forma en la que se creaban los horarios para los diferentes programas de la Universidad de los Llanos.” (VILLAR, 2017), esta es una metodología de acompañamiento para el software que tiene la universidad de los llanos.

Título: Aplicación de algoritmos meta - heurísticos a la asignación de salas y salones, programa Ingeniería de Sistemas jornada diurna Universidad Libre.

Autores: Walter Enrique Rodríguez Rodríguez

Lugar: Bogotá, Colombia

Link: <https://repository.unilibre.edu.co/handle/10901/8799>

“Acorde a los planes, visión y el compromiso de la universidad de prestar educación de alta calidad, la automatización del proceso de asignación de salas y salones reduce las inconsistencias ayudando al buen comienzo y desarrollo de los planes académicos de cada semestre en la institución, así como obtener información veraz acerca de la suficiencia o insuficiencia de la capacidad instalada y de los posibles planes de adquisición de nuevos equipos y creación de nuevas salas para atender la creciente demanda por parte de los alumnos a la institución. Además de aliviar la carga que obtiene el encargado de planear y organizar las salas y salones de su programa, el automatizar esta tarea permite que se concentre en otras actividades y que los estudiantes se sientan más a gusto con la institución reduciendo los niveles de estrés que se puedan presentar debido a las inconsistencias de esta tarea. La aplicación de un algoritmo meta-heurístico es un aporte metodológico que puede servir como base para futuras investigaciones, implementaciones o como material de estudio para los estudiantes de la universidad que deseen conocer y observar técnicas de inteligencia artificial aplicadas y desarrolladas en la institución.” (Rodríguez, 2010), este proyecto se basa en la creación de una metodología a seguir para la creación de un algoritmo para la asignación de horarios.

3. Propuesta de Algoritmo

Este capítulo enumera los pasos que se tuvieron en cuenta hasta el desarrollo de la aplicación teniendo los requerimientos iniciales como las metas establecidas para determinar que se ha desarrollado una aplicación que cumpla con estas condiciones.

3.1 Requerimientos Iniciales

Utilizando como punto de partida el proceso de horarios utilizado para la universidad de pamplona y la experiencia de docentes del programa de sistemas

- Se deben permitir definir restricciones para las aulas, los docentes y materias. En las aulas una restricción define las horas o días en las que no se pueda utilizar por lo tanto no se asignaran grupos en ese momento, en los docentes las restricciones implican según su tipo de contrato si es planta se deberá establecer su horario de trabajo, si es ocasional se le asignara su carga completa , si su contrato es catedra servidor público las restricciones implican las horas en las que este docente no pueda asistir a clases ejemplo: si solo se le pueden asignar un grupos de 6-8 de la mañana o de 6-10 de la noche.
- Se deben identificar los cruces de las materias si pertenecen al mismo semestre no se deben cruzar, esto para garantizar que los estudiantes que van en línea puedan tomar su carga académica completa correspondiente al semestre
- Las aulas deben usarse el mayor tiempo posible y están organizadas por tipos de aula, los cuales serán usados para determinar la asignación de un bloque de clases a un grupo perteneciente a una materia en particular, este proceso es realizado por cada grupo que sea registrado.
- Para cada docente se le define un número máximo de horas a programar y un listado de materias que puede orientar, cada vez que se asigne un grupo se deben descontar las horas que ya se les asignen a los docentes

- Si existe más de un grupo de la misma materia y solo un docente, se le debe asignar en diferentes horas para evitar su cruce. También se debe tener en cuenta el caso contrario que existan más de un grupo, pero más de 1 docente que la puedan dictar deberá asignarse de ser posible a la misma hora, pero en aula diferente.
- Se debe generar un reporte para poder revisar que si se han asignado todas las materias con sus respectivas aulas y docentes junto con la hora y el día en que se asignaron.

3.2 Funcionamiento del Algoritmo

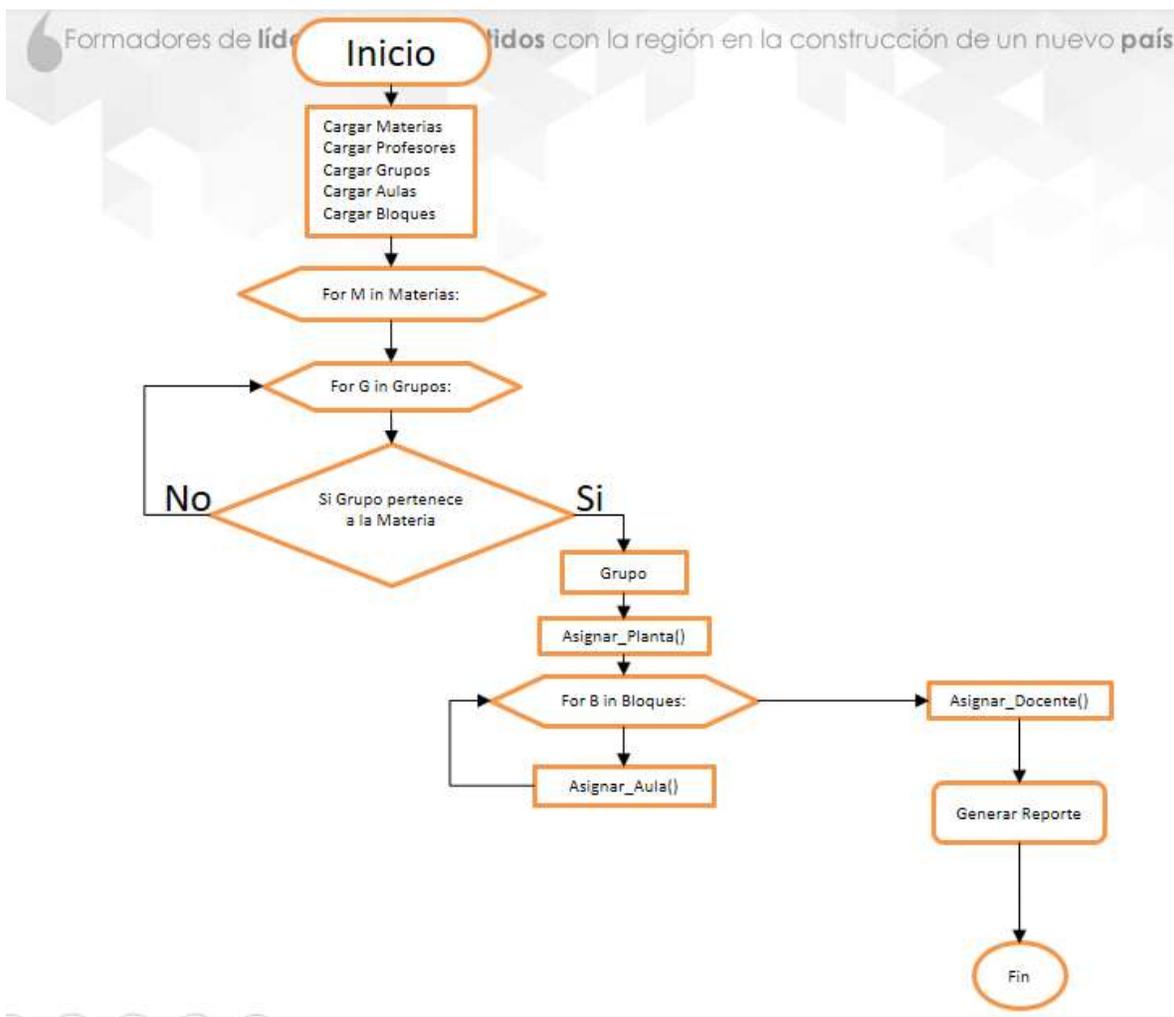


Grafico01 diagrama de flujo del funcionamiento que tiene el algoritmo

Los objetos que se requieren (tipos de aulas, aulas, materias, grupos, bloques, docentes y las materias que estos dictan). Como inicio del algoritmo, la función `asignar_Aula()` recibe los grupos, bloques que los componen, entre otras. Esta función se encarga de pedir los docentes que dictan las materias con sus restricciones respectivas y evalúa la disponibilidad de un horario al bloque de cada grupo por materia, por medio de esta función se obtiene la hora de inicio, la hora fin, y el día que corresponde asignados a los bloques. Luego resta asignar los docentes al grupo, la primera función es `asignar_planta()`, la cual asigna los docentes que tienen una carga académica definida, y así el algoritmo tendrá en cuenta las restricciones del mismo en la asignación del horario. El algoritmo tiene en consideración las prioridades que se le asignan a los docentes con relación a la materia, y contempla un orden en la asignación. Por último, las materias que siguen sin docente se procesan por medio de la función `asignar_docente()`, la cual se encarga de evaluar la disponibilidad de los docentes restantes en la base de datos con relación a los horarios establecidos anteriormente.

Una vez finalizada la ejecución del algoritmo, se genera el horario en un objeto **data** que contiene los grupos con docentes y los bloques que los componen con sus respectivos días y horas. La salida de todo el proceso se puede visualizar en la herramienta web, la cual genera las consultas en la base de datos, u obtener por medio de un archivo CSV que se genera del objeto **data**.

3.3 Herramienta Desarrollada

En este inciso se abordan las principales características de la herramienta planteada como solución al problema abordado.

3.3.1 Requerimientos Técnicos

En el presente trabajo después de estudiar diferentes alternativas se decidió usar las siguientes tecnologías

- Virtual enviroment (es un generador de entornos virtuales)
- Python 3.6 o superior
- Mysql el actual.
- Django versión 2.1 o superior
- Numpy librería

3.3.2 Modelo de datos

Especifica el modelo entidad relación que se definió para el desarrollo de la aplicación teniendo en cuenta los requerimientos iniciales del algoritmo.

3.3.2.1 Modelo de datos comprimido

El siguiente grafico muestra las tablas y sus relaciones sin mostrar los campos que lo componen esto se hace para generar una idea de cómo se estableció la dirección de la cardinalidad.

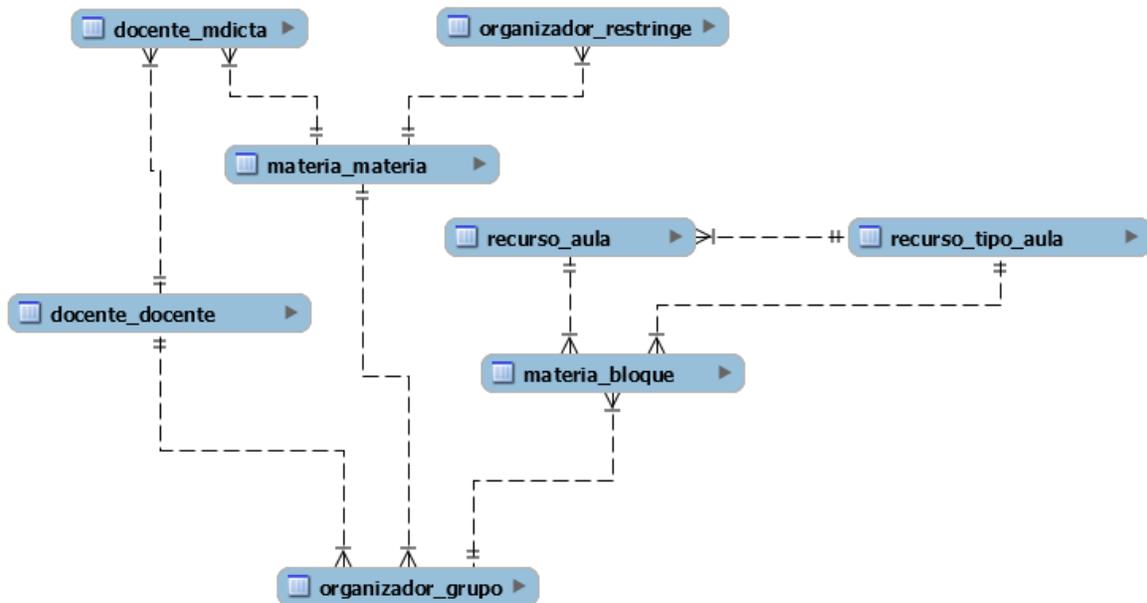


Grafico02: Modelo comprimido base de datos

3.3.2.2 Modelo de Datos A detalle

Este modelo se muestran las tablas con detalle, se observan los campos que las componen y su tipo.

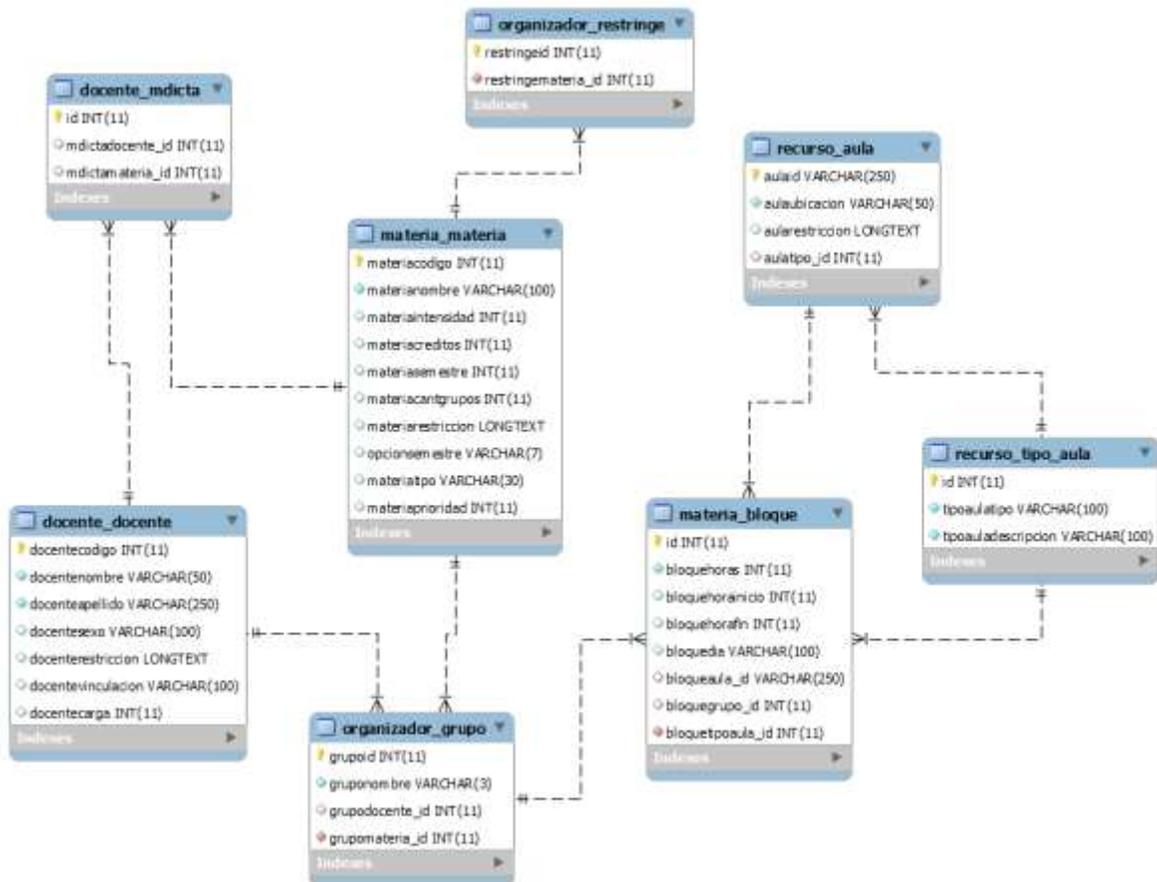


Grafico03: Modelo entidad Relación diseñado y utilizado para el proyecto

3.3.3 Funcionalidad

La interfaz utiliza las tres variables más importantes que son el corazón del proyecto Materias, Aulas y Docentes. A continuación, se detallará cada funcionalidad que se creó para facilitar el proceso de inserción y ordenamiento de los datos.

3.3.3.1 Gestión Docente

Permitió listar y registrar los docentes que requiere el programa y contiene los campos necesarios como nombres, apellidos, carga entre otros. De igual manera permite editar los datos del docente, eliminarlos y también agregarles las materias que el docente puede dictar, y recibe unas restricciones que se crearon con el fin de que se pueda seleccionar los días en los que el docente no puede asistir al aula con el fin de no asignarle un horario en esas horas.

3.3.3.2 Gestión Materia

Contiene clases visuales que permiten listar las materias que se encuentran registradas en la base de datos, añadir una materia nueva editar una materia existente o en su defecto eliminar una materia, pero también incluyo un espacio para las restricciones de la materia específicamente para saber los días de disponibilidad para dictar la materia si ese fuera el caso y permite registrar los bloques que pertenecen a la materia pues cada uno cuenta con 1 o 2 bloques dependiendo de su tipo

3.3.3.3 Gestión Aula

Contiene clases visuales que permiten listar las aulas y los tipos de aulas que hay registrados en la base de datos así mismo permite registrar un aula nueva o un tipo de aula nuevo, también tiene un espacio para las restricciones que era útil para definir qué día o que horas el aula no se iba a usar.

3.3.3.4 Generador de Horario

Contiene clases no visuales que se encargan de sustraer las variables de la base de datos y mediante sus funciones seleccionar un grupo, asignarle una materia, un aula, un horario y buscar un docente que cumpla con el horario y que pueda dictar la materia que se necesita, por ultimo genera un archivo Excel donde muestra el código del grupo, el código de la materia, el nombre de la materia, la hora de inicio, la hora de fin, el día, el aula, el semestre al que pertenece la materia y evitar los cruces entre los grupos de una misma materia.

El recorrido del objeto empieza en la clase **main()** donde se hacen las consultas para llenar las listas antes de iniciar las diferentes recorridas las clases que se usaron fueron las siguientes:

Clase Horario

```
class Horario:
    def __init__(self):#es la matriz general
        #self.restricciones = restricciones
        #import ipdb;ipdb.set_trace()
        #print(restricciones+ "hola")
        self.dias = {'Lunes': 0, 'Martes': 1, 'Miercoles': 2, '
        self.horas = {'600': 0, '700': 1, '800': 2, '900': 3, '
                        '1700': 11, '1800': 12, '1900': 13, '2000': 14,
```

Img03: crea la matriz inicial.

La clase horario se encarga de crear la matriz de restricciones de 6*16, contiene las funciones asignar, restringir, desocupar y las funciones que se encargan de buscar las aulas y asignar los docentes.

Clase Materias

```
class Materias:
    #carga los atributos de la consulta al modelo materia
    def __init__(self, materia):
        self.codigo = materia.materiacodigo
        self.nombre = materia.materianombre
        self.intensidad = materia.materiaintensidad
        self.creditos = materia.materiacreditos
        self.semestre = materia.materiasemestre
        restriccion = materia.materiarestriccion
        lista = ast.literal_eval(restriccion)
        values = list(lista.values())
        restriccion = np.array(list(map(int, values))).reshape(6,16).T
        self.restricciones = Horario()
        self.restricciones.Horario = restriccion
        self.grupos = materia.materiacantgrupos
        self.opcionsemestre = materia.opcionsemestre
```

Img04: Constructor de la clase materias

Esta clase recibe los datos de la consulta a la base de datos de la tabla materias incluyendo el campo de las restricciones que es de tipo string.

Clase Profesor

```
class Profesor:
    #carga los atributos de la consulta al modelo profesor
    def __init__(self, docente):
        self.codigo = docente['codigo']
        self.nombre = docente['nombre']
        self.apellido = docente['apellido']
        self.sexo = docente['sexo']
        restriccion = docente['restriccion']
        lista = ast.literal_eval(restriccion)
        values = list(lista.values())
        restriccion = np.array(list(map(int, values))).reshape(6,16).T
        self.restricciones = Horario()
        self.restricciones.Horario = restriccion
        self.vinculacion = docente['vinculacion']
        self.carga = docente['carga']
        self.materias = docente['materias']
```

Img05: Constructor de la clase profesor

Esta clase es el constructor y recibe los datos de la consulta a la base de datos en la tabla docente y crea una matriz con el campo restricciones que es de tipo String.

Clase Aulas

```
class Aulas:
    #carga los atributos de la consulta al modelo de aula
    def __init__(self, aula):
        self.codigo = aula.aulaid
        self.ubicacion = aula.aulaubicacion
        restriccion = aula.aularestriccion
        lista = ast.literal_eval(restriccion)
        values = list(lista.values())
        restriccion = np.array(list(map(int, values))).reshape(6,16).T
        self.restricciones = Horario()
        self.restricciones.horario = restriccion
        self.tipo = aula.aulatipo
```

Img06: Constructor de la clase aulas

Este es el constructor de la clase aulas se encarga de recibir los datos de la consulta a las tablas Aula al igual que las clases anteriores recibe un campo de restricciones que es un String y debe ser convertido a una matriz.

Clase Grupoa

```
class Grupoa:
    #este objeto se forma de la consulta de l
    #linea se hace el ciclo for y se llena la
    def __init__(self, grupo):
        self.codigo = grupo.grupoid
        self.materia = ""
        self.profesor = grupo.grupodocente
        self.bloques = []#'bloques': {'numb
```

Img07: Constructor de la clase grupo

Esta clase recibe los datos iniciales que son enviados desde la clase main y se forman de una serie de una consulta a la tabla materia, bloque, docente y las restricciones que recibe son las que tiene la materia, la clase grupo se compone de dos funciones, **asignar aula ()** y **asignar profesor ()**.

Función Bloquea

```
class Bloquea:
    def __init__(self, bloque):
        self.codigo = bloque.id
        self.horas = bloque.bloquehoras
        self.tipoaula = bloque.bloquetipoaula
        self.aula = ""
        self.horainicio = ""
        self.horafin = ""
        self.dia = ""
        self.grupo = bloque.bloquegrupo
```

Img08: constructor de la clase Bloquea

Esta función se crea a partir de una consulta a los grupos solicitando los bloques que tiene relacionados

Clase Horario

La clase horario está compuesta por:

- **Función init ():** esta función se encarga de crear la matriz genérica de días y horas.
- **Función asignar ():** esta función se encarga de guardar las restricciones en cada espacio de la matriz y le asigna el numero dos (2) para identificar que fue asignada por el algoritmo.
- **Función restringir ():** esta función revisa la matriz obtenida de la base de datos si el usuario asigna la restricción aparecerá el número uno (1).
- **Función desocupar ():** esta función revisa la matriz obtenida de la base de datos si no encuentra ninguna restricción entonces aparecerá el número cero (0).

Función main ()

Es la función principal de todo el algoritmo pues es la encargada de organizar los objetos y enviarlos a cada clase para llenar las variables, realiza la construcción del objeto grupos que mediante un ciclo repetitivo se encarga de enviar a los métodos asignar aula y asignar profesor después los recibe nuevamente y se

envían a un archivo de tipo **CSV** donde se descarga después por decisión del usuario.

3.4 Validación de la Herramienta

Para iniciar la validación del funcionamiento de la herramienta se realizó unas pruebas usando el plan completo de estudios de 56 materias, 20 docentes y 15 aulas incluyendo laboratorios especializados de informática, laboratorios propios de las materias y aulas teóricas, y se pudo establecer que ninguna de las materias se cruzó además de que se garantizó el máximo uso de las aulas el cual era uno de los requerimientos. Se hicieron pruebas incluyéndole restricciones de horas en las que una materia o un docente o un aula no se encontraba disponible se encontró que no se asignaron en esas horas, ese nos indica que se cumplió la restricción establecida a esas horas.

4. Conclusiones, Recomendaciones y Trabajos futuros

A través de este capítulo se mencionan algunas conclusiones relevantes que se tuvieron en cuenta en el transcurso del trabajo, adicionalmente se abordan nuevos escenarios de trabajo para mejorar, lo que implicaría una siguiente versión de la herramienta y se generan nuevas ideas de trabajos a futuro que se pueden derivar de este proyecto.

4.1 Conclusiones

- El uso de reglas y condiciones mejoró el rendimiento del algoritmo, pues al solo tener que comprobar algunas reglas básicas permite mejorar el tiempo de ejecución.
- El uso de Python dada su capacidad de trabajo y velocidad de operación, brindó ventajas al momento de la programación.
- El uso de Django permitió agilizar la conexión desde la creación de una plataforma web hacia el algoritmo creado.
- Al realizar la asignación completa el tiempo total utilizado para hacer todas las operaciones fue alrededor de (30) segundos, característica que demuestra la eficacia del algoritmo.
- Tal como se visualizó en el estado del arte, las herramientas utilizadas en la asignación de horarios no permiten la interacción entre código inteligente y la interfaz. Se comprobó que Django, gracias a su dependencia de Python permite la interacción anteriormente mencionada.
- Fue posible desarrollar una interfaz para facilitar la interacción del usuario con el algoritmo.
- La representación del algoritmo se realizó usando matrices para facilitar el uso de técnicas inteligentes en trabajos posteriores.

4.2 Recomendaciones

- Tener en cuenta los prerrequisitos de las materias.
- Definir un indicador para dar un orden a la programación de los grupos
- Utilizar en los servidores y probar su funcionamiento en un entorno de programación en paralelo.

4.3 Trabajo a futuro

- Aplicar técnicas inteligentes y comparar el rendimiento obtenido.
- Llevar a un entorno de producción más grande incluyendo más materias, recursos y aulas.
- Tener en cuenta la variable de estudiantes, como posible indicador en la asignación de nuevas reglas.
- La aplicación fue diseñada para un solo programa, pero se puede generalizar para varios programas académicos donde se podrían compartir docentes, aulas.
- Tener en cuenta los docentes que comparten servicios con otros programas.

5. Bibliografía

- Adriana Canseco-González, D. S.-P.-A.-B. (2016). Aplicación de Programación Lineal para la Asignación de Horarios en una Institución Educativa Mexicana. *Ingeniería Industrial*, 115.
- AHUMADA, J. A. (2014). *repositorio uchile*. Obtenido de repositorio uchile: <http://repositorio.uchile.cl/bitstream/handle/2250/131197/Generaci%F3n-de-horarios-acad%E9micos-en-INACAP-utilizando-algoritmos-gen%E9ticos.pdf?sequence=1>
- Alegsa, L. (17 de 05 de 2018). *alegsa.com*. Obtenido de *alegsa.com*: <http://www.alegsa.com.ar/Dic/variable.php>
- Aleman, R. (25 de 05 de 2014). *ventics*. Obtenido de *ventics*: <http://www.ventics.com/clases-de-algoritmos/>
- Aristo, S. P. (2016). *UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS*. Obtenido de *UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS*: <https://core.ac.uk/download/pdf/54236369.pdf>
- blog.aulaformativa. (26 de Enero de 2016). *blog.aulaformativa*. Obtenido de *blog.aulaformativa*: <https://blog.aulaformativa.com/listados-ruby-frameworks/>
- Cake. (2012). *book.cakephp*. Obtenido de *book.cakephp*: <https://book.cakephp.org/1.3/es/The-Manual/Beginning-With-CakePHP/What-is-CakePHP-Why-Use-it.html>
- campusMVP. (6 de junio de 2017). *campusmvp*. Obtenido de *campusmvp*: <https://www.campusmvp.es/recursos/post/los-5-mejores-frameworks-de-javascript-en-2017.aspx>
- CHARLYRED70. (20 de 10 de 2016). *programacionparatodosite*. Obtenido de *programacionparatodosite*: <https://programacionparatodosite.wordpress.com/2016/10/20/que-es-heuristica/>
- Daniel Calle López, J. C. (Marzo de 2018). *scielo.senescyt.gob.ec*. Obtenido de *scielo.senescyt.gob.ec*:

- http://scielo.senescyt.gob.ec/scielo.php?script=sci_arttext&pid=S1390-65422018000100102&lng=en&nrm=iso&tlng=en
- Díaz, A. N. (mayo de 2015). *Tesis PUCP*. Obtenido de Tesis PUCP: <http://tesis.pucp.edu.pe/repositorio/handle/123456789/6057>
- García, J. M. (11 de 12 de 2015). *arsys*. Obtenido de arsys: <https://www.arsys.es/blog/programacion/que-es-laravel/>
- Gómez, M. M. (2015). Las metaheurísticas: tendencias actuales y su aplicabilidad en la ergonomía. *Actualidad y Nuevas Tendencias*, 108-120.
- Group, T. P. (2019). *postgresql.org*. Obtenido de postgresql.org: <https://www.postgresql.org/about/>
- Jacqueline Alexandra Mena Fonseca, E. F. (2015). *Universidad de las Fuerzas Armadas ESPE*. Obtenido de Universidad de las Fuerzas Armadas ESPE: <http://repositorio.espe.edu.ec/handle/21000/11211>
- Jesuita, E. (26 de 09 de 2017). *uoc.fje.edu*. Obtenido de uoc.fje.edu: <https://fp.uoc.fje.edu/blog/que-son-las-interfaces-graficas-o-gui/>
- Luis Alejandro López Balboa. (2017). *riunet.upv.es*. Obtenido de riunet.upv.es: <https://riunet.upv.es/handle/10251/86742>
- MAMANI, S. H. (2016). *Tesis Una- Puno*. Obtenido de Tesis Una- Puno: <http://repositorio.unap.edu.pe/handle/UNAP/3008>
- management, D. (16 de junio de 2017). *kyocera*. Obtenido de kyocera: <https://smarterworkspaces.kyocera.es/blog/conceptos-base-datos-orientada-objetos/>
- Marco Vinicio Añazco Maldonado, H. L. (2016). *digdigital.epn.edu*. Obtenido de digdigital.epn.edu: <http://bibdigital.epn.edu.ec/handle/15000/16682>
- Mármol, P. (16 de Noviembre de 2018). *educaciontrespuntocero*. Obtenido de educaciontrespuntocero: <https://www.educaciontrespuntocero.com/recursos/herramientas-elaborar-horarios/34971.html>
- Martha Ximena Hidalgo Zurita, C. d. (2018). *repositorios.espe.edu*. Obtenido de repositorios.espe.edu: <http://repositorio.espe.edu.ec/handle/21000/15054>

Microsoft. (2019). *microsoft.com*. Obtenido de microsoft.com:
<https://info.microsoft.com/gartner-odbms-magic-quadrant-register.es-la.1.html?ls=website>

misapuntessistemas. (Enero de 2012). *Mis apuntes*. Obtenido de Mis apuntes:
http://misapuntes.info/MisDocumentosWindows/Para_que_sirve_Microsoft_Access.pdf

Newman, A. K. (24 de abril de 2009). *Chronicle of Higher Education*.
chronicle.com.

Nicho, G. M. (junio de 2010). *Tesis pucp*. Obtenido de Tesis pucp:
http://tesis.pucp.edu.pe/repositorio/bitstream/handle/123456789/551/BEJARANO_NICHO_GISSELLA_MAR%C3%8DA_PLANIFICACI%C3%93N_HORARIOS_PERSONAL_CIRUG%C3%8DA.pdf?sequence=1&isAllowed=y

O.J Ibarra-Rojas, Y. R.-S. (2014). *ingenieriadetransporte.org*. *Ingeniería de Transporte*, 92. Obtenido de *ingenieriadetransporte.org*:
<http://www.ingenieriadetransporte.org/index.php/sochitran/article/view/155>

oracle. (2015). *oracle.com*. Obtenido de oracle.com:
<https://www.oracle.com/mysql/>

Pérez Serrano, G. (2012). *cvc.cervantes.es*. Obtenido de *cvc.cervantes.es*:
https://cvc.cervantes.es/ensenanza/biblioteca_ele/diccio_ele/diccionario/metodologiacualitativa.htm

Petra, M. F. (2016). *uvadoc.uva.es*. Obtenido de *uvadoc.uva.es*:
<http://uvadoc.uva.es/handle/10324/20790>

Pichardo, M. F. (28 de Febrero de 2011). *Universidad Autónoma del Estado de Morelos*. Obtenido de Universidad Autónoma del Estado de Morelos:
<http://riaa.uaem.mx/xmlui/bitstream/handle/20.500.12055/93/progmat312011Revision.pdf?sequence=1&isAllowed=y>

piensasolutions. (19 de Octubre de 2017). *piensasolutions.com*. Obtenido de *piensasolutions.com*:
<https://www.piensasolutions.com/blog/principales-lenguajes-programacion-web/>

PINEDA, J. M. (03 de 11 de 2016). *coriaweb*. Obtenido de *coriaweb*:
<https://www.coriaweb.hosting/codeigniter-cuales-algunas-ventajas/>

programacion.net. (2018). *programacion.net*. Obtenido de programacion.net:
https://programacion.net/articulo/los_4_frameworks_web_mas_populares_para_python_1069

René Alfonso Carrillo Flores, H. O. (2015). *dspace.uce.edu.ec*. Obtenido de dspace.uce.edu.ec: <http://www.dspace.uce.edu.ec/handle/25000/5171>

Reyes, P. J. (2018). *dspace.ups.edu*. Obtenido de dspace.ups.edu:
<https://dspace.ups.edu.ec/handle/123456789/16121>

Robles, V. (05 de 08 de 2017). *victorroblesweb*. Obtenido de victorroblesweb:
<https://victorroblesweb.es/2017/08/05/que-es-angular-y-para-que-sirve/>

Rodrigo Hernandez, J. M. (2008). Programacion de Horarios de Clases y Asignacion de Salas para la Facultad de Ingenieria de la Universidad Diego Portales Mediante un Enfoque de Programacion Entera. *Revista Ingenieria de Sistemas*, 121-143.

Rodríguez, W. E. (2010). *repository.unilibre.edu.co*. Obtenido de repository.unilibre.edu.co:
<https://repository.unilibre.edu.co/handle/10901/8799>

Salazar, S. W. (15 de Febrero de 2016). *repositorio.pucese.edu*. Obtenido de repositorio.pucese.edu:
<https://repositorio.pucese.edu.ec/bitstream/123456789/612/1/PAREDES%20SALAZAR%20SERGIO%20WLADIMIR.pdf>

Sampieri, H. (s.f.). *Metodologia de la investigacion 5° Edicion*.

Sinnexus. (2017). *Sinnexus*. Obtenido de Sinnexus:
https://www.sinnexus.com/business_intelligence/datamining.aspx

sqlite. (08 de febrero de 2019). *sqlite.org*. Obtenido de sqlite.org:
<https://www.sqlite.org/index.html>

TÉBAR, E. (27 de Septiembre de 2018). *wearemarketing*. Obtenido de wearemarketing: <https://www.wearemarketing.com/es/blog/frameworks-en-el-desarrollo-web-las-mejores-practicas-para-tu-negocio-online.html>

tiposde.org. (2017). *tiposde.org*. Obtenido de tiposde.org:
<https://www.tiposde.org/ciencias-exactas/843-tipos-de-algoritmos/>

Víctor F. Suárez, Á. G. (septiembre de 2012). *Universidad Nacional de Colombia*.
Obtenido de Universidad Nacional de Colombia:
https://scielo.conicyt.cl/scielo.php?pid=S0718-07642013000100012&script=sci_arttext

VILLAR, E. L. (2017). *repositorio.unillanos.edu.co*. Obtenido de *repositorio.unillanos.edu.co*:
[http://repositorio.unillanos.edu.co/jspui/bitstream/001/1105/1/RUNILLANOS%20SIS%20290P%20ACOMPA%C3%91AMIENTO%20AL%20DESARROLLO%20DEL%20SOFTWARE%20PARA%20EL%20REGISTRO%20DE%20HORARIOS%20EN%20LOS%20DIFERENTES%20PROGRAMAS%20DE%20LA%20UNIVERSIDAD%20DE%20LOS%](http://repositorio.unillanos.edu.co/jspui/bitstream/001/1105/1/RUNILLANOS%20SIS%20290P%20ACOMPA%C3%91AMIENTO%20AL%20DESARROLLO%20DEL%20SOFTWARE%20PARA%20EL%20REGISTRO%20DE%20HORARIOS%20EN%20LOS%20DIFERENTES%20PROGRAMAS%20DE%20LA%20UNIVERSIDAD%20DE%20LOS%20)

6. Anexos

En este capítulo se encuentran descritos los procedimientos necesarios para el apropiamiento de la interfaz por medio de la plataforma web.

6.1 Manual de Usuario

**PROYECTO ALGORITMO PARA LA PLANIFICACIÓN DE HORARIOS EN EL
PROGRAMA DE INGENIERÍA DE SISTEMAS DE LA UNIVERSIDAD DE
PAMPLONA
MANUAL DE USUARIO PARA EL USO ADECUADO DEL ALGORITMO DE
ORGANIZACIÓN DE HORARIOS**

Norbey Paez Amaya | Ingeniería de Sistemas Universidad de Pamplona | 2019

Introducción

Este manual se creó como apoyo para instruir al usuario en el correcto uso de la herramienta desarrollada, para así garantizar el mejor resultado a la ejecución del algoritmo de asignación, los pasos importantes están basados en las tres (3) variables contempladas en el planteamiento del problema y así obtener la mejor solución posible, así como el manejo de restricciones puede facilitarle a usted la asignación de horarios deseados.

1. Abrir el navegador y teclear la ip del servidor o el nombre adjuntando el puerto como aparece en la imagen:



No es seguro | 192.168.0.111:8000

Img01: ip del servidor

2. presione enter para acceder a la página index que después de unos segundos aparecerá donde se aprecian un total de 5 botones **Docente**, **Aulas**, **Materia**, **Generar Horario** y **Descargar Horario Generado**, como se puede apreciar en la siguiente imagen:



Img02: index.html

3. Lo primero que se debe a hacer es ir al botón **Aulas** y le daremos clic esto mostrara la página que lista las aulas que se compone de 3 botones Añadir Nueva Aula Tipo de Aula y un botón Atrás que nos devuelve a nuestra página anterior como se puede apreciar en la imagen:



Img03: Aula_listar.html

- ahora debe dar clic en el botón tipo de aula te llevara a listar todos los tipos de aula que están registrados como se muestra en la imagen:



Img04: Tipo_aulalistar.html

- De clic en el botón Nuevo para agregar un nuevo tipo de aula también si lo desea puede darle atrás y volverá al listar aulas como se muestra en la imagen:



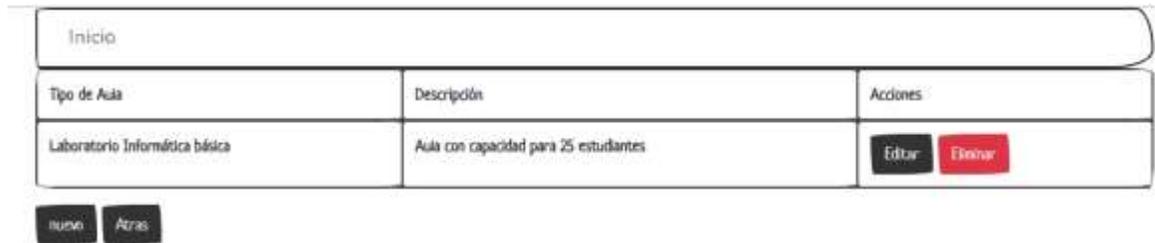
Img05: botones nuevo y atrás

- A continuación, después de dar clic en el botón nuevo le aparecerá un formulario como el que se muestra en el ejemplo en la imagen:

The image shows a form for registering a new classroom type. It includes a 'Inicio' button, a 'Tipo:' label, a text input field containing 'Laboratorio Informática básica', a 'Descripción:' label, a text input field containing 'Aula con capacidad para 25 estudiantes', and 'Guardar' and 'Atras' buttons.

Img06: registrar tipo aula

7. Después de guardar se encontrará con el listar aula que ahora muestra el nuevo tipo de aula que usted ha creado como se puede apreciar en la imagen:

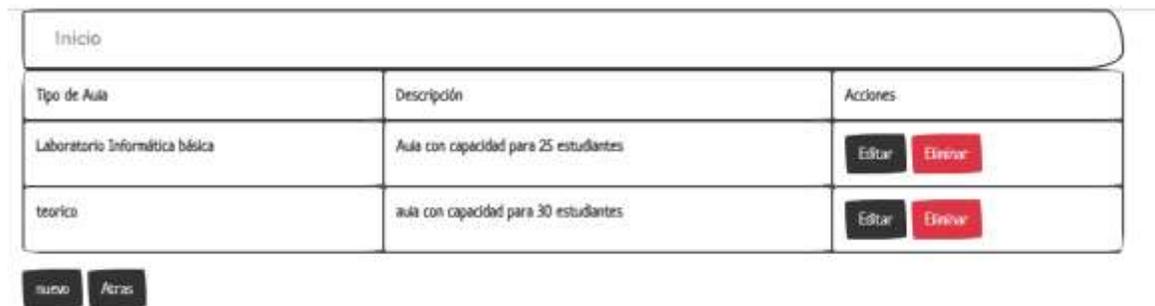


Inicio		
Tipo de Aula	Descripción	Acciones
Laboratorio Informática básica	Aula con capacidad para 25 estudiantes	Editar Eliminar

[Nuevo](#) [Atrás](#)

Img07.1: listar Tipo aula

Para un óptimo funcionamiento se pueden agregar más tipos de aulas como lo muestra esta otra imagen:

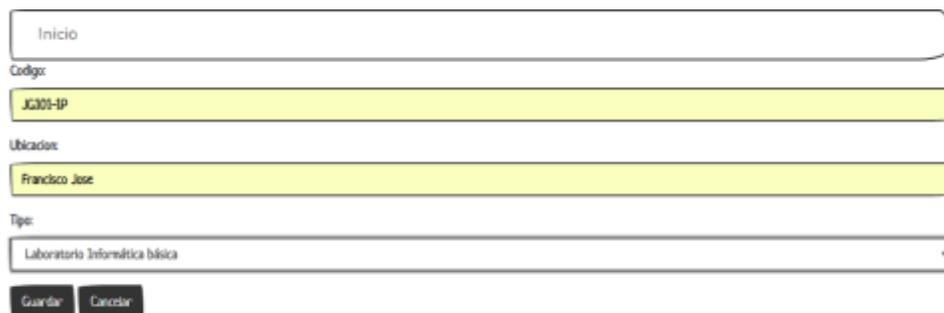


Inicio		
Tipo de Aula	Descripción	Acciones
Laboratorio Informática básica	Aula con capacidad para 25 estudiantes	Editar Eliminar
teorico	aula con capacidad para 30 estudiantes	Editar Eliminar

[Nuevo](#) [Atrás](#)

Img07.2: listar varios tipos de aulas

8. Al finalizar la operación debe dar clic al botón Atrás y volver a la página de listar Aula allí ahora seleccione el botón nuevo y agregue una nueva aula como se muestra en la imagen:



Inicio

Código:
JC001-IP

Ubicación:
Francisco Jose

Tipo:
Laboratorio Informática básica

[Guardar](#) [Cancelar](#)

Img08: Registrar aula

9. Justo después de guarda ya aparecerá una nueva aula en nuestro listado, debe agregar más aulas ejemplo aula teórica, laboratorio especializado dependiendo de los requisitos de las materias como se muestra en la imagen:

Inicio		
Codigo	Ubicacion	Acciones
CD10SP	Camilo Daza	<input type="button" value="Editar"/> <input type="button" value="Eliminar"/> <input type="button" value="Ver Restricciones"/>
JG101-1P	Francisco Jose	<input type="button" value="Editar"/> <input type="button" value="Eliminar"/> <input type="button" value="Ver Restricciones"/>

Img09: listar Aula

10. Después de dar clic en Ver restricciones aparecerá una ventana como se muestra en la imagen donde podrá seleccionar los días y las horas donde el aula no se usar al ejecutar el algoritmo estas restricciones serán tenidas en cuenta

Inicio						
Por favor seleccione las horas en las que el aula no se pueda usar						
Hora / Día	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado
6:00 - 6:59	<input type="checkbox"/>					
7:00 - 7:59	<input type="checkbox"/>					
8:00 - 8:59	<input type="checkbox"/>					
9:00 - 9:59	<input type="checkbox"/>					
10:00 - 10:59	<input type="checkbox"/>					
11:00 - 11:59	<input type="checkbox"/>					
12:00 - 12:59	<input type="checkbox"/>					
13:00 - 13:59	<input type="checkbox"/>					

Img10: Restricciones de aula

11. Ahora debe registrar una materia así que de clic en el botón atrás como se ve en la imagen:



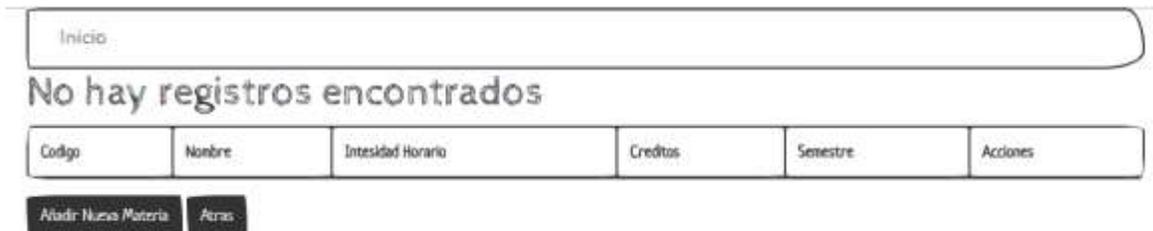
Img11.1: botones de aula

Luego en la página index debe dar clic al botón materia:



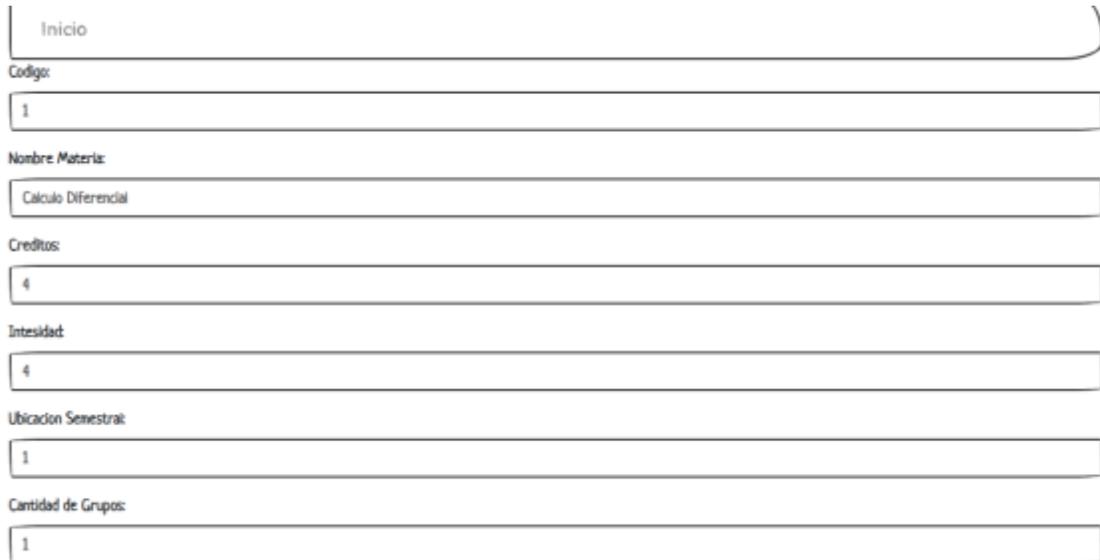
Img11.2: botones index

12. Al dar clic en el botón materia le llevará a la página listar materia, se observará tal como se muestra en la siguiente imagen:



Img12: listar materia

13. Ahora debe registrar una materia para esto debe darle clic al botón Añadir Nueva Materia esto le llevara al formulario de registrar materia como se aprecia en la imagen:



Formulario de registro de materia con los siguientes campos:

- Inicio: []
- Código: [1]
- Nombre Materia: [Calculo Diferencial]
- Creditos: [4]
- Intesidad: [4]
- Ubicación Semestral: [1]
- Cantidad de Grupos: [1]

Img13.1: registrar materia

En el campo cantidad de grupos puede escoger la cantidad de grupos que desea de la materia o si usted prefiere no asignar grupos de esa materia, el algoritmo no lo tendrá en cuenta al momento de ejecutarse, también si usted desea que se tenga en cuenta las materias del mismo semestre sin que se crucen los horarios con otras de otros semestres o que solo se asigne como se muestra en la imagen:

Opción Semestre:

- Desea que no se cruce con materias del mismo Semestre?
- Solo asigna



Guarda Atras

Img13.2: opción semestre

- Por ultimo debe escoger el tipo de materia puede seleccionar entre los tres diferentes (básica, socio humanística y profesional que fusiona profundización y profesional) como se observa en la siguiente imagen:

Opción Semestre:

- Desea que no se cruce con materias del mismo Semestre?
- Solo asigna

Tipo de Materia:

Img13.3: Tipo de materia

Aparecerá de la siguiente manera:

157005	CÁLCULO DIFERENCIAL	1	<input type="button" value="Editar"/>	<input type="button" value="Eliminar"/>	<input type="button" value="Restricción"/>
--------	---------------------	---	---------------------------------------	---	--

Img13.4: listar materias

14. Si se desea se le pueden agregar restricciones a las materias, debe dar clic al botón Restricción esto le llevara a las restricciones donde podrá escoger los días y las horas donde la materia no se dictará al hacer esto, el algoritmo tendrá en cuenta las restricciones que se le asignen y no asignara la materia en donde se encuentre una restricción como aparece en la imagen:

Inicio						
Por favor seleccione las horas en las que la materia no se pueda dictar						
Hora Día	Lunes	Martes	Miercoles	Jueves	Viernes	Sabado
6:00 - 6:59	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
7:00 - 7:59	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
8:00 - 8:59	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
9:00 - 9:59	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
10:00 - 10:59	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
11:00 - 11:59	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Img14: Materia restricción

15. Una vez termine de asignar las restricciones a la materia guarde esto le llevara al listado de las materias ahí nuevamente debe dar clic al botón atrás que lo llevara al index:

Inicio					
Código	Nombre	Intensidad Horario	Créditos	Semestre	Acciones
1	Cálculo Diferencial	4	4	1	Editar Eliminar Bloque Restricción

[Añadir Nueva Materia](#) [Atras](#)

Img15: materia listar

16. Ahora se deben crear los docentes así que en la página index debe dar clic al botón docente como se ve en la imagen:

Inicio							
Docente	Aulas	Materia	Grupos	Generar Horario	Descargar Horario Generado	Limpiar Datos Generados	Restringir Día

Img16: index

17. Después de dar clic en docente esto le llevara a listar los docentes que están registrados en la base de datos como es su primera entrada esto aparece vacío como se ve en la imagen:

Inicio				
No hay registros encontrados				
Código	Nombres	Apellidos	Sexo	Acciones

[Añadir Nuevo Docente](#) [Atras](#)

Img17: Docente listar

18. Debe dar clic en el botón Añadir Nuevo Docente esto le llevara al formulario rellenar los datos del docente entre los campos se encuentran Código (Cedula del Docente), Nombre, Apellidos, Sexo, Tipo de Contratación (ocasional, planta, catedra, catedra publica) y en la Carga Actual la que le corresponda a su tipo de contratación si tiene alguna descarga debe hacer la resta y el resultante colocarlo en la Carga Actual:

Inicio

Código:

1

Nombre:

pepito

Apellido:

perez

Sexo:

masculino

Tipo Contratación:

ocasional

Carga Actual:

2

Guardar Atras

Img18: Formulario registrar docente

19. Después de guardar podrá observar el nuevo docente en el listado y ahora cuenta con los botones Materias, Restricción y Grupos, debe dar clic a Materias para asignarle materias que el docente puede dictar como se puede ver en la imagen:

100220	Pepito	Perez	Ocasional	Editar	Eliminar	Restricción	Materias	Grupos
--------	--------	-------	-----------	--------	----------	-------------	----------	--------

Img19: Docente listar

20. Una vez de clic en Materias le listará todas las materias que puede asignarle al docente, pero en este momento estará vacío como se aprecia en la imagen:

Inicio

No hay registros encontrados

Codigo Materia	Materia	Acciones
----------------	---------	----------

Añadir Materia Atras

Img20: Materias que dicta el docente

21. Para que el docente sea tenido en cuenta por el algoritmo debe agregarle materias que el docente pueda dictar junto con la prioridad de asignarle esa materia (Baja, Media y Alta), esto depende de cómo desee que sea considerado al asignar los grupos y debe darle clic al botón guardar como aparece en la imagen:

Inicio

Materia:

CÁLCULO DIFERENCIAL

Prioridad de Asignacion: Baja

Guardar Atras

Img21: nueva materia a dictar

22. Después de darle clic al botón guardar se listarán automáticamente las materias que le ha agregado al docente como se muestra en la imagen:

157005	CÁLCULO DIFERENCIAL	Baja	Eliminar	Cambiar Prioridad
--------	---------------------	------	----------	-------------------

Añadir Materia Atras

Img22: listar materias que dicta el docente

23. Debe darle clic al botón atrás después de terminar de agregar materias al docente esto lo llevara al listado de los docentes:

100220	Pepito	Perez	Ocasional	Editar	Eliminar	Restricción	Materias	Grupos
--------	--------	-------	-----------	--------	----------	-------------	----------	--------

Img23: Docente listar

24. Ahora si lo desea puede agregar restricciones al docente como el día y la hora donde no puede asistir, aunque por su tipo de vinculación tiene unas restricciones que han sido preseleccionadas esto será tenido en cuenta por el algoritmo al momento de asignar el horario de la materia, debe seleccionar las opciones que desea o dejar solamente vacío y darle al botón guardar como se puede ver en la siguiente imagen:

Inicio						
Por favor seleccione las horas en las que no pueda asistir						
Hora Día	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado
6:00 - 6:59	<input type="checkbox"/>					
7:00 - 7:59	<input type="checkbox"/>					
8:00 - 8:59	<input type="checkbox"/>					
9:00 - 9:59	<input type="checkbox"/>					
10:00 - 10:59	<input type="checkbox"/>					
11:00 - 11:59	<input type="checkbox"/>					
12:00 - 12:59	<input type="checkbox"/>					
13:00 - 13:59	<input type="checkbox"/>					

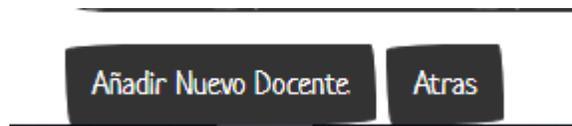
Img24: Restricción Docente

25. Luego de darle clic al botón guardar esto le direccionara al listado de los docentes que tiene registrados como la siguiente imagen:

Codigo	Nombres	Apellidos	Vinculación	Acciones
100220	Pepito	Perez	Ocasional	Editar Eliminar Restricción Materias Grupos

Img25.1: Listar Docente

Luego solo debe ir al final del listado de docentes y dale clic al botón atrás esto le direccionara a la página index.html:



Img25.2: botón atrás

26. Devuelta en la página index debe ahora dar clic al botón Grupos esto nos listara todos los grupos que se crearon automáticamente, teniendo en cuenta la cantidad de grupos que fueron colocados al momento de insertar una materia, la página listar grupo tendrá la siguiente apariencia:

Inicio			
Grupo	Materia	Docente	Acciones
A	ÁLGEBRA LINEAL	Sin Asignar	Eliminar Editar Bloques Horario
A	ANÁLISIS DE ALGORITMOS	Sin Asignar	Eliminar Editar Bloques Horario
A	BASES DE DATOS II	Sin Asignar	Eliminar Editar Bloques Horario
A	BIOLOGÍA GENERAL	Sin Asignar	Eliminar Editar Bloques Horario
A	CÁLCULO DIFERENCIAL	Sin Asignar	Eliminar Editar Bloques Horario

Img26: Listar grupos

27. Debe darle clic al botón Bloques y agregar los bloques que tiene la materia esto dependerá de los créditos ejemplo: si la materia es de 4 créditos esto significa que está compuesta de dos bloques de dos horas cada uno cuando usted crea el bloque tiene la opción de escoger como se muestra en la siguiente imagen sin embargo debe aparecerle vacío así que deberá dar clic en el botón Nuevo Bloque y agrega el bloque de las horas que pertenezcan:

Inicio				
Grupo	Materia	Numero de Horas Dia	Tipo de aula	Acciones
A	CÁLCULO DIFERENCIAL	2	Teorica	Eliminar
A	CÁLCULO DIFERENCIAL	2	Teorica	Eliminar

Nuevo Bloque Atras

Img27.1: Listar bloques

Inicio

Horas:

4

Tipo de Aula: Teorica

Guardar Atras

Img27.2: Nuevo bloque

28. Después de terminar de agregar los bloques que requiere el grupo este proceso se deberá repetir por cada grupo de la materia hasta que todos los grupos tengan los bloques correspondientes una vez terminado debe darle al botón atrás para ir al listado de los grupos:

Inicio			
Grupo	Materia	Docente	Acciones
A	ÁLGEBRA LINEAL	Sin Asignar	Eliminar Editar Bloques Horario
A	ANÁLISIS DE ALGORITMOS	Sin Asignar	Eliminar Editar Bloques Horario
A	BASES DE DATOS II	Sin Asignar	Eliminar Editar Bloques Horario
A	BIOLOGÍA GENERAL	Sin Asignar	Eliminar Editar Bloques Horario
A	CÁLCULO DIFERENCIAL	Sin Asignar	Eliminar Editar Bloques Horario

Img28: Listar Grupo

29. En el botón Horario puede verificar los campos que aún están sin asignar y una vez ejecute el algoritmo estos campos se llenaran con esa información:

Inicio				
Grupo	Aula	Dia	Hora Inicio	Hora Fin
A	Sin Asignar	Sin Asignar	Sin Asignar	Sin Asignar
A	Sin Asignar	Sin Asignar	Sin Asignar	Sin Asignar

[Atras](#)

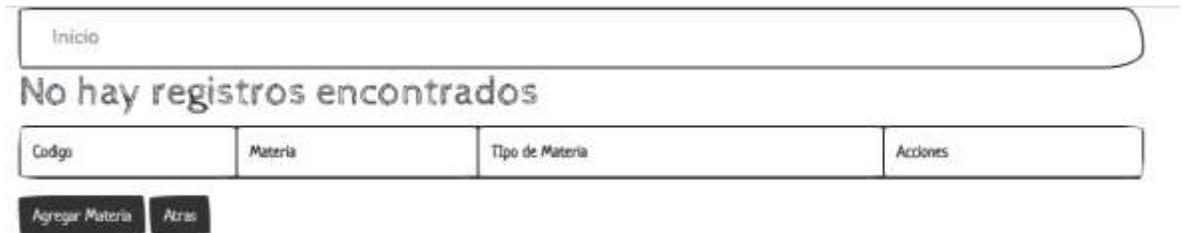
Img29: Horario del Bloque

30. Al finalizar de agregar los bloques debe darle clic al botón atrás para volver al botón listar grupos y allí nuevamente darle clic al botón atrás para ir a la página index:

Inicio							
Docente	Aulas	Materia	Grupos	Generar Horario	Descargar Horario Generado	Imprimir Datos Generados	Restringir Día

Img30: Pagina index

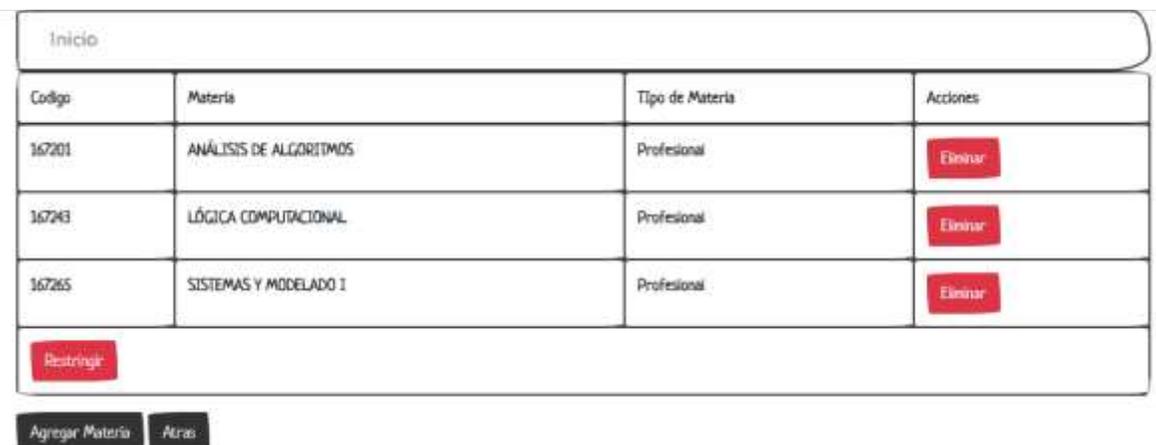
31. Si desea que una o más materias no se dicten un día en específico, entonces debe dar clic al botón Restringir Día esto nos llevara a la página que lista las materias a restringir en este caso está vacía como se muestra en la imagen:



Img31: Listar materias a restringir

32. Puede añadir las materias que desea no se dicten en un día en específico este proceso se hará en dos pasos:

a) Debe escoger las materias que desea restringir como se ve a continuación:



Img32.a: Materias a restringir.

- b) Una vez terminado ahora debe darle clic al botón Restringir esto le mostrara una tabla de checkbox donde se podrán escoger los días y las horas donde no se van a dictar esas materias y una vez terminado solo deberá guardar:

Inicio

Por favor seleccione las horas en las que no pueda asistir

Hora Dia	Lunes	Martes	Miercoles	Jueves	Viernes	Sabado
6:00 - 6:59	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
7:00 - 7:59	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
8:00 - 8:59	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
9:00 - 9:59	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
10:00 - 10:59	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
11:00 - 11:59	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
12:00 - 12:59	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
13:00 - 13:59	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
14:00 - 14:59	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
15:00 - 15:59	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
16:00 - 16:59	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
17:00 - 17:59	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
18:00 - 18:59	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
19:00 - 19:59	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
20:00 - 20:59	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
21:00 - 21:59	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Guardar Atras

Img32.b: Tabla de selección días y horas

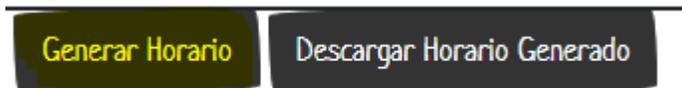
33. Al finalizar solo debe dar guardar y esto le direccionara a la página que lista las materias restringidas, solo debe dar atrás para volver a la página index por ultimo solo resta ejecutar el algoritmo.

34. En la página index teniendo todos los datos listos ahora debe buscar los botones Generar Horario y Descargar Horario Generado que se encuentran a su derecha en la página index:



Img34.1: Index

35. Luego de ubicar los botones debe presionar el Botón Generar Horario. Primero esto ejecutará el código del algoritmo espere mientras carga:



Img35.1: Botones de Horario

Y luego darle clic en el botón Descargar Horario Generado el cual generará un archivo CSV que se descargará automáticamente en su navegador:



Img35.2: Botones de Horario

a) Después de darle clic al botón Descargar Horario Generado puede buscar el archivo Horario.csv:



Img31: archivo descargado

- b) Ahora solo restar abrir el archivo y organizar la vista de un CSV eso deberá arrojar algo parecido a la imagen:

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	
CODIGOMAT	MATERIA	GRUPO	CAPA	TOTALMATE	TIPOGRUPO	Dia	NOMBREHO	BI_HORAINI	BI_HORAFIN	NOMBREHO	BF_HORAINI	BF_HORAFIN	RECURSO	Profesor	
167250	PROGRAMA A					Martes		800					909	CD101P	Orlando
167250	PROGRAMA A					Jueves		800					1009	FJ103-1P	Orlando
167250	PROGRAMA B					Martes		1000					1159	CD101P	Luis Alberto
167250	PROGRAMA B					Jueves		1400					1609	FJ103-1P	Luis Alberto
167251	PROGRAMA A					Martes		1400					1509	CD101P	Mauricio
167251	PROGRAMA A					Viernes		800					1009	FJ103-1P	Mauricio
167251	PROGRAMA B					Martes		1600					1759	CD101P	Edgar Alexis
167251	PROGRAMA B					Viernes		1400					1609	FJ103-1P	Edgar Alexis
167231	ESTRUCTUR					Martes		800					709	CD101P	Sergio
167231	ESTRUCTUR					Jueves		1100					1309	FJ103-1P	Sergio
167201	ANÁLISIS DE A					Martes		1200					1359	CD101P	Omar
167201	ANÁLISIS DE A					Jueves		1700					1909	FJ103-1P	Omar
167238	INTRODUCC					Martes		800					909	FJ103-1P	Armando
167238	INTRODUCC					Jueves		800					1009	FJ103-1P	Armando
167232	FUNDAMEN					Martes		1800					1909	CD101P	Edgar Alexis
167232	FUNDAMEN					Viernes		1100					1359	FJ103-1P	Edgar Alexis
167232	FUNDAMEN					Martes		2000					2159	CD101P	Yesid
167232	FUNDAMEN					Viernes		1700					1909	FJ103-1P	Yesid

Img32.2: vista generado

36. Puede revisar en el botón grupos y notara que se han asignado docentes a los grupos:

Inicio			
Grupo	Materia	Docente	Acciones
A	ÁLGEBRA LINEAL	Diga Parada	Eliminar Editar Bloques Horario
A	ANÁLISIS DE ALGORITMOS	Omar Portilla	Eliminar Editar Bloques Horario
A	BASES DE DATOS II	Sergio Peñaloza	Eliminar Editar Bloques Horario
A	BIOLOGÍA GENERAL	Pepito Perez	Eliminar Editar Bloques Horario
A	CÁLCULO DIFERENCIAL	Diga Parada	Eliminar Editar Bloques Horario

Img36: Listar grupos con docente

a) Al dar clic en horario ya debe notar que se ha asignado aula, día y hora al bloque que compone el grupo:

Inicio				
Grupo	Aula	Día	Hora Inicio	Hora Fin
A	CD101P	Miércoles	1400	1559
A	CD101P	Viernes	800	959

Atras

img36.a: bloques con horas

b) Ahora si va selecciona el botón docente y luego el botón Grupos puede observar que el docente ya tiene grupos asignados:

Inicio			
Grupo	Materia	Docente	Acciones
A	BIOLOGÍA GENERAL	Pepito Perez	Horario
A	LABORATORIO BIOLOGÍA GENERAL	Pepito Perez	Horario
A	LABORATORIO DE QUÍMICA GENERAL	Pepito Perez	Horario
A	QUÍMICA GENERAL	Pepito Perez	Horario
A	MATEMÁTICAS ESPECIALES	Pepito Perez	Horario

Volver

Img36.b: Listar grupos asignados al docente

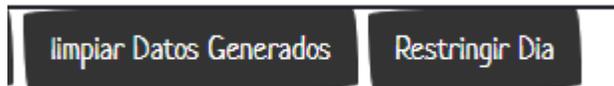
c) Y de esta manera puede observar al darle clic al botón horario los bloques como quedaron asignados incluyendo el día, hora de inicio y hora de fin:

Inicio				
Grupo	Aula	Día	Hora Inicio	Hora Fin
A	CD101P	Miércoles	1200	1359
A	CD101P	Viernes	600	759

Atras

Img36.c: Listar bloques con horario

37. En la página index existe un botón llamado limpiar Datos al darle clic todos los cambios que se han hecho al momento de correr el algoritmo volverán a su estado inicial generó:



Img37.1: Botón limpiar Datos Generados

Inicio			
Grupo	Materia	Docente	Acciones
A	ÁLGEBRA LINEAL	Sin Asignar	Eliminar Editar Bloques Horario
A	ANÁLISIS DE ALGORITMOS	Sin Asignar	Eliminar Editar Bloques Horario
A	BASES DE DATOS II	Sin Asignar	Eliminar Editar Bloques Horario
A	BIOLOGÍA GENERAL	Sin Asignar	Eliminar Editar Bloques Horario
A	CÁLCULO DIFERENCIAL	Sin Asignar	Eliminar Editar Bloques Horario

Img37.2: grupos sin docente

Inicio				
Grupo	Aula	Dia	Hora Inicio	Hora Fin
A	Sin Asignar	Sin Asignar	Sin Asignar	Sin Asignar
A	Sin Asignar	Sin Asignar	Sin Asignar	Sin Asignar

Atras

Img37.3: Bloques sin Aulas, días y horas

Inicio			
No hay registros encontrados			
Grupo	Materia	Docente	Acciones
Volver			

Img37.4: Grupos asignados a los docentes