Propuesta de entrega horario:

**Fecha de entrega:**

21 de diciembre de 2016

**Objetivo propuesto:**

Entrega de programa completamente operativo, con interfaz gráfica para que el cliente pueda interactuar con el programa, añadiendo, modificando y borrando parámetros como pueden ser profesores, asignaturas, cursos además de poder asignar las diferentes restricciones.

Además, se podrá cargar la configuración con todos los datos para la creación del horario desde un fichero tipo Excel.

**Requisitos a cumplir del programa:**

* **Prototipo de motor de reglas para el horario**: Al generar un horario cumple con los demás requisitos
* **Horas Máx/Min semanales de un profesor:** Un profesor no puede impartir más de X e Y configurables horas a la semana al generar su horario, se suman sus horas de clase y tienen que ser mayores que x y menores que y.
* **Horas Máx/Min diarías de un profesor:** Un profesor no puede impartir más de X e Y configurables horas al día al generar su horario, se suman sus horas de clase y tienen que ser mayores que x y menores que y.
* **Horas continuas/dia de un profesor:** Un profesor no debe tener más de X (configurable) horas seguidas de clase al generar su horario, se coge su grupo de clases continuas más largo y nos aseguramos de que es menor o igual que X.
* **Máx de Gaps por día:** Un profesor no debe tener más de X horas libres al día entre su primera y última clase al generar su horario. Hora fin - hora inicio - horas de clase >= X.
* **Máx de Gaps por semana:** Se calcula la cantidad de horas libres entre la hora de inicio y fin de cada día, se suman y tienen que ser menos que X.
* **Máx veces un intervalo horario por semana:** Un profesor puede definir un horario y que dicho horario no se le pueda asignar más de X veces configurable cada semana.
* **Grupos de Alumnos:** Los alumnos deben estar asignados a grupos para cada asignatura, se comprueba intentando asignar un alumno a una clase sin asignarle grupo y no nos lo permite el programa.
* **Grupos de profesores:** Los profesores pueden agruparse.
* **Conjunto de horas preferidas para las sesiones:** Las sesiones tienen un parámetro al ser asignadas a un horario que pesará para poner, por ejemplo, las sesiones de Redes por la mañana, se pueden poner preferencias de tres maneras: Hora de inicio (preferimos empezar a partir de x hora), hora final (preferimos terminar antes de x hora) o intervalo (preferimos que la hora de inicio sea después de X y antes que Y).
* **Min/máx de días entre sesiones:** Para Las asignaturas se puede definir que no pasen más de X días entre una sesión y la siguiente, para comprobarlo introducimos una sesión un día cualquiera y ponemos una sesión más allá de cuantos días hayamos establecido con X, si nos avisa y no nos permite colocar así las sesiones la aplicación, será correcto.
* **Sin solapamiento, aunque sean de distinto curso:** Al generar el horario se intenta colocar las asignaturas en huecos libres para asignaturas pertenecientes a un mismo grado en la medida de lo posible aunque las asignaturas a colocar no sean de un mismo curso.
* **Min/Máx de gap de horas:** Las sesiones no pueden estar separadas por más de X horas de clases configurables.
* **Crear grupos de profesores:** Todas las restricciones que tenga un profesor se le pueden aplicar a determinado grupo de profesores, aunque con un menor peso que las restricciones aplicadas a cada profesor individual (las restricciones individuales sobrescriben a las restricciones de grupo) se comprueba aplicando solo una restricción a un profesor, una como grupo y esta misma como profesor individual para distintas horas, debe coger siempre la restricción individual por encima de la del grupo.
* **Hora para comer:** las horas de clase se organizarán de tal modo que de 14:30 a 15:30 haya una hora libre para comer.
* **Instalación automática:** script para instalar la aplicación y todas sus dependencias en una maquina con un sistema operativo basado en Debian y Ubuntu
* **Interfaz web:** Interfaz web para el control de la aplicación, visualización de horarios generados, carga, modificación, y visualización de datos como profesores, asignaturas, sesiones, etc.

**Tareas**

* **Motor de Reglas:** Implementar todas las restricciones para que puedan ser utilizadas en la generación de horarios: **Clara Calonge.**
* **Interfaz web:** Implementar la interfaz web para el control de la aplicación: **Rafael Murillo, Adolfo Lopez**
* **Prolog –** **Javascript:** Hacer uso de los predicados realizados en Prolog desde la aplicación: **Jose Cara, Rafael Murillo**
* **Importar hoja de cálculo de Excel:** **Jose Cara**
* **Creación del servidor web:** Crear servidor web para poder mostrar la interfaz: **Angel Nuñez-Torron, Jose Cara**