



*Comprometidos con el desarrollo regional*

**Desarrollo de un Software para la generación y gestión de horarios  
del personal operativo del Hospital San Carlos (Saldaña), que  
precisan de turnos de trabajo rotatorios**

**Cristhian Camilo Rojas Villanueva**

**Código: 2320142081**

**Edwin Fabián Mora Garzón**

**Código: 2320111094**

**Facultad de Ingeniería**

**Ingeniería industrial**

**Ibagué, 2018**



Desarrollo de un Software para la generación y gestión de horarios del personal operativo del Hospital San Carlos (Saldaña).

---



**Desarrollo de un Software para la generación y gestión de horarios del personal operativo del Hospital San Carlos (Saldaña), que precisan de turnos de trabajo rotatorios**

Trabajo de grado que se presenta como requisito parcial para optar al título de:

**Ingeniero Industrial**

**Cristhian Camilo Rojas Villanueva**

**Código: 2320142081**

**Edwin Fabián Mora Garzón**

**Código: 2320111094**

**Director:**

**Ing. Mec. Juan Camilo Cática Buendía**

**Facultad de Ingeniería**

**Ingeniería Industrial**

**Ibagué, 2018**



### **DEDICATORIA**

*Este trabajo de grado va dedicado a mi padres y a mi hija, quienes fueron mi fuerza y apoyo en este largo camino de formación profesional, a quienes les debo mis valores, perseverancia y principios los cuales me ayudaron a culminar esta etapa.*

*A todas las personas que hicieron posible el desarrollo de este proyecto.*



## Tabla de Contenido

<b>Resumen .....</b>	<b>6</b>
<b>Introducción .....</b>	<b>8</b>
<b>Planteamiento del problema.....</b>	<b>9</b>
<b>Justificación .....</b>	<b>10</b>
<b>Objetivos.....</b>	<b>11</b>
<b>Alcances y limitaciones .....</b>	<b>12</b>
<b>Marco de Referencia .....</b>	<b>13</b>
o Antecedentes	13
o Turnos de trabajo rotativo	14
o Código Sustantivo del Trabajo	15
o Modelado de Software	16
o Diagramas UML	16
o MVC	16
▪ Modelo.....	16
▪ Vista .....	17
▪ Controlador.....	17
o Programación de Turnos Rotativos	17
▪ Cíclico.....	17
▪ No-Cíclico .....	19
o Lenguajes de Programación	21
o Bases de Datos	21
▪ Estructuradas .....	21
▪ No Estructuradas .....	22
<b>Metodología .....</b>	<b>24</b>
<b>Análisis del Hospital .....</b>	<b>25</b>
o Estructura Organizacional	25
o Parámetros legales (Lo Referente al CST)	26
o Ciclos de Trabajo y el CST	28
o Parámetros institucionales	30
o Modelo operativo antiguo del Hospital	31
<b>Diseño del Software .....</b>	<b>35</b>
o Modelado	35
o Arquitectura de Solución	38
▪ Modelo.....	38
▪ Vista .....	38
▪ Controlador.....	38
o UML	39



○ Alcances y limitaciones del software	43
○ Algoritmo de Turnos Rotativos	43
<b>Desarrollo del Software</b> .....	<b>46</b>
○ Selección del Lenguaje	46
○ Base de Datos	46
▪ Diseño .....	46
▪ Selección .....	47
▪ Ciclo de la Información .....	48
○ Interfaz Gráfica	48
<b>Módulo de Exportación</b> .....	<b>51</b>
<b>Conclusiones</b> .....	<b>52</b>
<b>Bibliografía</b> .....	<b>54</b>



## Resumen

En la actualidad las empresas de distintos sectores requieren de personal para el desarrollo de las actividades de mantenimiento, producción y gestión en las diferentes áreas de la organización; muchas de estas empresas requieren de personal contratado bajo el concepto de turnos rotativos, en el cual se desempeñan labores referentes a la naturaleza del cargo de manera continua sin ningún tipo de interrupción. Bajo dichos conceptos los descansos, las vacaciones y ocasiones donde parte del personal abandona parcialmente las actividades de su cargo son compensadas con personal adicional el cual permite establecer continuidad en el cargo y una operación adecuada para el personal, por tal motivo para las empresas donde se requiere de dicho concepto de trabajo precisan de un sistema capaz de establecer y mantener activamente la asignación de turno correctamente.

El trabajo desarrollado se establece como medida de mejoramiento de la situación establecida anteriormente, donde el manejo y gestión de personal puede llegar a ser controlada y operada mediante un software que permita desarrollar los horarios de trabajo del personal operativo para el caso del hospital San Carlos Saldaña, respetando la carga operativa por medio del cumplimiento de los estándares empresariales y las normativas de ley referentes a los turnos de trabajos rotativos y el Código Sustantivo del Trabajo.

**Palabras clave:** No estructurada, turnos rotativos, algoritmo

## ABSTRACT

Nowadays, companies from several sectors require personnel for the development of maintenance, production, and management activities in different areas of the organization. Many of these companies require personnel hired under the concept of rotating shifts, in which they work continuously on tasks related to the nature of the job. Under these concepts, breaks, holidays, and occasions where part of the personnel stop doing certain activities of their cargo are compensated with additional personnel; which allows to establish the continuity of the task and adequate operation of the staff. For this reason, the companies, which require that precise work concept, need a system capable of actively establishing and maintaining the shift assignment correctly.



This work is developed as a procedure of improvement for the situation established above. Where the administration and management of personnel can be controlled and operated through a software that allows them to develop the working schedules of the operative personnel in the case of San Carlos Saldaña hospital. respecting the workload by complying with the business protocols and regulations related to rotating work shifts and the Substantive Labor Code.

**Keywords: Unstructured, Rotating shifts, Algorithm**



## Introducción

El presente documento detalla el proceso de diseño y elaboración de un software de gestión de horarios de trabajo para turnos rotativos para el hospital San Carlos ubicado en el municipio de Saldaña en el departamento del Tolima. En la actualidad el hospital cuenta con un grupo de empleados que desarrollan turnos de trabajo rotativo, sin embargo, no cuentan con un modelo que permita desarrollar ajustes adecuados sobre la carga horaria repartida entre el personal, por esta razón se precisa de un modelo que permita equilibrar la carga horaria entre el personal sin el incumplimiento a las normas definidas en el Código Sustantivo del Trabajo (en adelante CST).

Para dar solución al problema planteado se diseñó el software de tal manera que implementará el conjunto de reglas definidas en el conjunto de artículos asociados al tiempo de trabajo de los empleados y el modelo de contratación definidos en el CST. Por esta razón el proyecto se plantea como solución general a los problemas relacionados al incumplimiento de los acuerdos pautados durante la contratación de personal que desarrolla turnos de trabajo rotativos.

Uno de las razones más importante por las cuales se plantea la solución a través de un software es debido a la dificultad expresada por el personal del hospital para seguir una regla por parte general que permita equilibrar la carga laboral entre los empleados sin incumplir con ninguno de los artículos del CST, por ello se diseñó un software que permite la construcción de unidades representativas del hospital y de las empresas en general para el control de los turnos a través del tiempo, así como el control de incumplimiento de las normas, conteo en horas semanales y mensuales desarrolladas por los empleados, definición de grupos de trabajo y la implementación de un algoritmo que permite la generación de turnos con posibilidad de asignación de permisos o días de descanso.

Como medida general de integración con plataformas ofimáticas convencionales se desarrolló un módulo de integración con Excel, el cual permite exportar los horarios desarrollados para su almacenamiento y visualización en diferentes entornos.

El presente documento se encuentra dividido en cinco secciones, la primera de ellas es el marco de referencia, donde se analizan los antecedentes, el Código Sustantivo del Trabajo y otros aspectos relacionados con la construcción de software; la segunda sección se centra en el análisis del hospital, donde se mencionan la estructura organizacional del hospital, los parámetros legales que se debían contemplar en el desarrollo del software, así como los parámetros institucionales. La siguiente sección, contempla todo lo relacionado al diseño del software, la arquitectura de la solución y el algoritmo implementado para la generación de los turnos rotativos; la cuarta sección contiene todo lo relacionado al desarrollo del software, la selección del lenguaje y la base de datos, así como la descripción general de la interfaz gráfica del software; finalmente, en la última sección se menciona el módulo de exportación del software.



## **Planteamiento del problema**

El hospital San Carlos (Saldaña) requiere de personal que desempeñe las labores de trabajo continuo por medio de un equipo de trabajo contratado bajo el concepto de turnos rotativos que se encuentre en la capacidad de prestar el servicio de urgencias de manera permanente; de esta manera, el hospital San Carlos se encuentra sujeto a la tarea de generar y gestionar los horarios de todo el personal médico, el cuerpo de enfermeros y el personal auxiliar de servicios generales cumpliendo la reglamentación establecida por lineamientos del hospital y el Código Sustantivo del Trabajo.

Con el objetivo de cumplir la normativa del Código Sustantivo del Trabajo y respetar los derechos laborales del personal, el hospital establece turnos de trabajo de 8 horas diarias, con días específicos de descanso los fines de semana para cada trabajador, siendo esta una de las razones por las cuales es necesario el personal adicional que pueda cubrir los turnos de descanso de otros trabajadores del cuerpo operativo los fines de semana, que pueda cubrir los permisos laborales y demás eventualidades; por tal motivo la generación de los horarios de trabajo y la gestión del mismo por parte del personal a cargo de esta labor, se encuentra sujeta a cierto tipo de problemas tales como la violación directa en las horas de trabajo que estipula el Código Sustantivo del Trabajo, incumplimiento en los contratos laborales en función de las horas y los turnos trabajado que el personal desarrolle incurriendo en sobrecostos para el hospital, así como diferencias notables entre los turnos y horarios de trabajos establecidos para el personal. Por tal motivo es necesario implementar un sistema capaz de gestionar óptimamente los horarios de trabajo en el hospital San Carlos, que puedan ser llevados a cabo por el responsable de dicha labor, minimizando la carga laboral y mejorando la distribución de la carga laboral entre los trabajadores.

Partiendo de la problemática establecida anteriormente, se planteará la siguiente interrogante: ¿Qué tipo de tecnología y modelos de desarrollo pueden ser empleados para generar un sistema capaz de gestionar la asignación de turnos rotativos para el hospital San Carlos (Saldaña)?



## **Justificación**

Para el hospital San Carlos Saldaña es de vital importancia el ajuste de horarios y la carga laboral que se demanda al personal operativo, donde se pretende evitar la asignación desbalanceada en el tiempo de trabajo de los empleados, con el fin de evitar incumplimientos en lo establecido por artículos del Código Sustantivo del Trabajo, los cuales hacen referencia a las horas de trabajo diario y semanal para la modalidad de contratos donde se establece que la figura de empleado ejecuta turnos de trabajo rotativos. Adicionalmente el hospital busca desarrollar cambios en las operaciones que permitan generar un ambiente laboral justo y equitativo entre el personal operativo. Por tal razón, el hospital precisa del desarrollo e implementación de un software que permita la generación de horarios de trabajo rotativo para satisfacer las necesidades anteriormente planteadas.

Otra de las razones más importantes de la implementación de software es el aumento de las empresas manufactureras que gestionan horarios de trabajo rotativos, puesto que muchas de las actividades que desarrollan se categorizan como actividades de alta prioridad y no deben ser interrumpidas, por lo cual se genera la necesidad de controlar dichas actividades, bien sea por razones relacionadas a la productividad de la empresa o del servicio ofrecido por la misma. Algunas de las empresas u organizaciones donde se evidencia este tipo de modelos de trabajo son aeropuertos, hospitales y hoteles.

De esta manera, se busca el aporte que genera el desarrollo de esta investigación siendo conveniente para diferentes tipos de empresas. Las tendencias del mercado actual y futuro obligan a las empresas a ser cada vez más competitivas. Toda empresa que desee competir en el mercado actual debe considerar la información como uno de los activos más importantes. Por esta razón, es necesario que la empresa disponga de los sistemas de horarios de Información adecuados para suministrar rápida y eficientemente la información. En la mayoría de casos, las empresas distribuyen todos sus sistemas de horarios rotativos de información en varias aplicaciones, con la ineficiencia y repetición de datos que esto supone. Por ello, una opción adecuada puede ser el diseño de un software para la generación y gestión de horarios de personal operativo donde se precisan turnos de trabajo rotatorios.



## Objetivos

Este proyecto tiene como **Objetivo General** desarrollar un software capaz de generar, asignar y programar el servicio de turnos rotativos del personal operativo del hospital San Carlos (Saldaña). Para lograr el objetivo anteriormente mencionado se requiere de los siguientes **Objetivos Específicos**:

- Identificar los parámetros legales e institucionales que se emplean para el desarrollo de la asignación laboral del personal operativo que cumple con horarios de trabajo de turnos rotativos.
- Diseñar el modelo UML que permite construir y documentar el sistema, definiendo los aspectos funcionales de manera gráfica y estructurada en función de los parámetros establecidos.
- Desarrollar el software por medio del lenguaje de programación adecuado, que permita alcanzar de manera efectiva y eficiente los requerimientos funcionales definidos.
- Desarrollar un módulo que permita exportar la información generada para ser gestionada por medio de herramientas Office de Microsoft.



## **Alcances y limitaciones**

El proyecto basado en el diseño y desarrollo de un Software de gestión de turnos rotativos, tiene como objetivo la fabricación de una herramienta computacional capaz de generar y asignar los horarios del personal operativo de instituciones y empresas, donde el concepto de contratación establece los intervalos de tiempo específicos que el personal debe llevar a cabo. Es claro que la asignación de horarios y los estándares varían según sea el tipo de contratación, la naturaleza del trabajo, la normativa legal y otros factores relacionados con las políticas institucionales de las empresas y el estado donde se encuentren operando; por tal motivo, el desarrollo de dicho sistema contemplará los estamentos de ley de Código Sustantivo del Trabajo que rigen en territorio colombiano, así como algunos patrones para la gestión de horarios para personal operativo en empresas de distintos sectores, hoteles, hospitales y demás.

El diseño de la herramienta computacional quedará habilitada para su único uso e implementación en ordenadores de forma directa, es decir que funcionará de manera local donde la gestión de horarios se desarrollaría de manera autónoma para cada organización, sin embargo, queda claro que su diseño y estructuración podría ser empleada a futuro como base para el desarrollo de un sistema web de asignación y gestión de horarios.



## Marco de Referencia

### ○ Antecedentes

En la actualidad existen numerosos sistemas diseñados para la gestión de recursos del sector empresarial e institucional, los cuales centran su diseño en el manejo de inventarios, administración de recursos, ventas, proyecciones y gestión de actividades, adicionalmente las empresas gestionan y desarrollan proyectos con la ayuda de sistemas capaces de controlar las variables más influyentes durante la ejecución. Estos sistemas son desarrollados con el objetivo de mejorar la productividad de las empresas, siendo este un aspecto fundamental para lograr los parámetros de competitividad de las empresas.

Para el año 2015 fue desarrollado el trabajo de fin de grado presentado por el estudiante de ingeniería en informática Mario Marugán Cancio de la Universidad de Castilla, el cual establece los parámetros de funcionalidad de un sistema empleado para gestionar todos los aspectos relacionados con la organización de turnos y horarios de trabajo del cuerpo de enfermeros de clínicas y hospitales donde se tiene la posibilidad de gestionar preferencias de turnos, con la finalidad de obtener un buen desempeño laboral y que las condiciones de salud física y mental se mantengan en buenos términos (Marugán Cancio, 2015). Adicionalmente el documento presenta un modelo constructivo de las tecnologías empleadas para el desarrollo del sistema de gestión, estableciendo un modelo significativo para el manejo de personal.

El documento titulado “Un Algoritmo Evolutivo Para la Planificación de Sistemas de Turnos Rotativos en Empresas de Servicios”, se basa en el desarrollo de un sistema de gestión de turnos validado en una empresa del sector de Telecomunicaciones, el cual tiene como finalidad ajustar los horarios de trabajo de personal exento de descansos dominicales obligatorios para generar un ambiente de trabajo adecuado e impactar positivamente la productividad de los trabajadores.(Ortiz & Berg, s. f.)

La facultad de ingeniería de la Universidad Peruana de Ciencias Aplicadas (UPC) ha publicado en el año 2010 un documento basado en el desarrollo de un sistema para la gestión de recursos en instituciones educativas donde se puede gestionar horarios de clases, aulas y asignación académica para el grupo de docentes de las instituciones; en este trabajo se documenta el diseño y el análisis informático de la administración llevada a cabo en la UPC, el cual finalizó con la construcción del software y la implementación de bases de datos que permiten la gestión de horarios y carga académica. (VÁSQUEZ, GONZÁLEZ, & AHUMADA, 2010)

Uno de los sistemas más complejos y robustos que se encuentran en el mercado, es el sistema de gestión de información empresarial SAP, el cual cuenta con una estructura digital para llevar a cabo la gestión de numerosas áreas de una compañía, siendo principal para determinar aspectos fundamentales en confiabilidad y mantenimiento de equipos, gestión de inventarios y de transporte, logística empresarial y numerosas actividades que



desarrolla una organización, incorporando una herramienta apropiada para generar y administrar los horarios de trabajo de personal en el módulo diseñado para gestionar el área de recursos humanos.(Software de gestión RR.HH, s. f.)

#### ○ **Turnos de trabajo rotativo**

Los turnos de trabajo rotativo suponen de disponibilidad de tiempo parcial y cambiante para el personal operativo de las empresas donde se contratan bajo esta denominación; es claro que en este tipo de trabajos y su modalidad deben respetar las normativas en el marco legal de las actividades laborales, por tal motivo el artículo “La regla del trabajo por turnos y la necesaria contextualización para su entendimiento” plantea los principales conceptos legales soportados mediante los artículos establecidos en el Código Sustantivo del Trabajo para personal contratado en territorio colombiano. (Valdivieso Rueda, 2012)

El documento NPT 455 denominado “Trabajo a turnos y nocturno: aspectos organizativos”, hace mención a algunos modelos de seguimiento de turnos rotativos, definiendo los principales modelos y características, donde se clasifican según la continuidad con la que una persona o grupo de personas desarrollan el cumplimiento de su puesto de trabajo de manera sucesiva (Nogareda & Nogareda, 1995). Las formas de organización de trabajos por turnos mencionadas son:

- Sistema discontinuo: Usado para puestos de trabajo que permiten discontinuidad los fines de semana y en las noches, requiriendo de dos turnos para cubrir mañana y tarde.
- Sistema Semi-Continuo: Es empleado para puestos donde se interrumpe el puesto de trabajo una vez por semana, requiriendo de tres turnos, para operar durante mañana, tarde y noche.
- Sistema Continuo: Es un sistema empleado donde no se permite la discontinuidad y donde los puestos de trabajo requieren de más personal que en el sistema Semi-Continuo, con el fin de cubrir los descansos.



### ○ **Código Sustantivo del Trabajo**

El Código Sustantivo del Trabajo establece los parámetros fundamentales para desarrollar buenas condiciones laborales para la ciudadanía, respetando los derechos de las dos figuras, empleado y empleador, que conforman y llevan a cabo un contrato laboral.

El Artículo 161 del Código Sustantivo de Trabajo establece que la jornada ordinaria de trabajo es la que convengan las partes, o a falta de convenio, la máxima legal de ocho (8) horas diarias al día y cuarenta y ocho (48) a la semana; constituyendo todo trabajo que supere la jornada máxima legal, trabajo suplementario o de horas extras, las cuales según el Artículo 22 de la Ley 50 de 1990, no podrán exceder de 2 horas diarias y 12 horas a la semana.(Ministerio de Protección Social, 2011a)

El Artículo N° 165 del Código Sustantivo del Trabajo establece que: Cuando la naturaleza de la labor no exija actividad continuada y se lleve a cabo por turnos de trabajadores, la duración de la jornada puede ampliarse en más de ocho (8) horas, o en más de cuarenta y ocho (48) semanales, siempre que el promedio de las horas de trabajo calculado para un periodo que no exceda de tres (3) semanas, no pase de ocho (8) horas diarias ni de cuarenta y ocho (48) a la semana. Esta ampliación no constituye trabajo suplementario o de horas extras.(Ministerio de Protección Social, 2011b)

Artículo modificado por el artículo 3o. del Decreto 13 de 1967. También puede elevarse el límite máximo de horas de trabajo establecido en el artículo 161, en aquellas labores que por razón de su misma naturaleza necesiten ser atendidas sin solución de continuidad, por turnos sucesivos de trabajadores, pero en tales casos las horas de trabajo no pueden exceder de cincuenta y seis (56) por semana.(Ministerio de Protección Social, 2011c)



- **Modelado de Software**

El modelado corresponde a la técnica empleada para representar y determinar los elementos necesarios que conforman un sistema o software, estableciendo los aspectos más importantes, las funcionalidades y la estructuración que corresponde al esquema programación empleado. El modelado para la generación de un sistema capaz de controlar y gestionar los horarios de los empleados en una empresa requiere del conocimiento transversal que se ha venido desarrollando a lo largo del tiempo en numerosas empresas; para ello el autor del libro “Gestión de talento humano” en el capítulo 16 segmenta y estratifica de manera clara los principales aspectos para el desarrollo y manejo de bases de datos y sistemas de información para el área de Recursos Humanos (RR.HH).(Chiavenato, 2009)

- **Diagramas UML**

El modelamiento UML (Lenguaje Unificado de Modelado), se refiere a un lenguaje de propósito general que permite estructurar, segmentar y estratificar las unidades lógicas y visuales de un software por medio de un sistema gráfico de referencias, de esta forma UML permite desarrollar proyectos centrados en el desarrollo de software o generación de sistemas de cualquier índole donde los componentes más importantes se diseñan de manera gráfica estableciendo relación con los demás componentes necesarios, de esta forma, se modela cada uno de los parámetros funcionales que permiten dar solución a las problemáticas por medio de la generación de software. (Rumbaugh, Jacobson, & Booch, 2000)

- **MVC**

MVC corresponde al acrónimo Modelo-Vista-Controlador, este estilo o patrón de arquitectura resulta ser uno de los esquemas de diseño de mayor aceptación y aplicación durante el desarrollo de sistemas (Freeman, 2013). La idea detrás de este patrón de arquitectura es segmentar el desarrollo del software en tres secciones o piezas, las cuales son:

- **Modelo**

Es aquel que contienen o representan los datos con los que trabajan los usuarios. Estos pueden ser modelos de vista simples, que solo representan datos que se transfieren entre vistas y controladores; o pueden ser modelos de dominio, que contienen los datos en un dominio empresarial, así como las operaciones, transformaciones y reglas para manipular esos datos.



- **Vista**

La vista se encarga de enderezar parte de los datos contenidos en el modelo, de tal forma que se constituye como la interfaz gráfica de usuario. De esta forma es como el usuario final del sistema es capaz de acceder y manipular los datos.

- **Controlador**

Esta sección se encarga de gestionar las peticiones, las operaciones lógicas sobre el modelo y seleccionar los aspectos importantes para renderizar y permitirle al usuario hacer uso del sistema a través de la “vista”. Esta sección contiene en su mayoría las instrucciones lógicas más importantes, los accesos a las bases de datos y la comunicación con otros procesos.

- **Programación de Turnos Rotativos**

Los modelos de programación y generación de turnos rotativos se han elaborado con el objetivo de establecer un patrón de diseño para el periodo de planeación de los turnos y horarios del cuerpo de trabajo de las organizaciones e instituciones, para ello algunas soluciones que involucran la sistematización, que permita minimizar conflictos en los horarios y turnos, así como violaciones en las reglamentaciones y normas relacionadas. En el documento “Efficient Generation of Rotating Workforce Schedules” se menciona dos tipos de modelos para la programación de los horarios de trabajo por turnos rotativos; el primero de los modelos es el **cíclico (Cycle Workforce Schedule)** y el segundo es el modelo **no-cíclico (noncycle Workforce Schedule)** (Nysret, Johannes, & Wolfgang, s. f.).

- **Cíclico**

Implementado para optimizar el horario de trabajo estableciendo una carga operativa equivalente entre todos los empleados, donde los turnos de trabajo sucesivos pueden ser visualizados en una matriz entre los días de trabajo de todo el periodo de planeación y los empleados, como el modelo mostrado en la Figura 1, («Rotating 8 Hour Shift Schedule | 24/7 Shift Coverage | Learn Employee Scheduling», s. f.).

Empl	Día 1-7							Día 8-14							Día 15-21							Día 21-28																		
1	D	D	D	D	D	D	D			T	T	T	T	T	T	T			N	N	N	N	N	N	N	N														
2			T	T	T	T	T	T			N	N	N	N	N	N	N			D	D	D	D	D	D	D	D	D	D	D										
3	T	T			N	N	N	N	N	N	N						D	D	D	D	D	D	D	D			T	T	T	T	T	T								
4	N	N	N	N													D	D	D	D	D	D	D	D			T	T	T	T	T	T			N	N	N			

Figura 1. Modelo de programación cíclico para 4 empleados en turnos rotativos de 8 horas



Como se puede observar en la Figura 1, una matriz que ilustra el modelo cíclico, la rotación de turnos (D,T,N) se ve reflejada en el siguiente bloque de horario semanal (para este caso), es decir, que para un empleado el siguiente bloque de trabajo de siete días corresponde al horario asignado en la semana anterior del siguiente empleado, de esta forma, se cumple la rotabilidad y se garantiza los días de descanso para el cuerpo de trabajo. Los turnos de trabajo se definen como:

D: Turno Diurno, entre 7 am y 3 pm

T: Turno de Tarde, entre 3 pm a 11 pm

N: Turno Nocturno, entre 11pm a 7 am

Vacío: Días de descanso

En este modelo existen algunas variables a considerar como lo menciona (Nysret et al., s. f.); donde los bloques de trabajo ( $m$ ), son definidos como los días sucesivos de trabajo de un empleado y separados por días de descanso o por la terminación de un nuevo horario; el número de empleados ( $n$ ), la longitud en días del horario ( $w$ ), el cual convencionalmente corresponde a 7 (una semana) para ajustar la rotación de forma semanal, finalmente otra de las variables a considerar es el número de días de un periodo de planeación total, el cual, para el modelo cíclico corresponde al producto entre el número de empleados y la longitud del horario, como se puede observar en la Figura 1, donde el número de empleados es cuatro y la longitud en días de cada horario es de siete, dando como resultado una longitud de todo el periodo de planeación de 28 días.

En (Nysret et al., s. f.) Para hacer uso del modelo, uno de los problemas considerados en el documento muestra la variedad de posibles soluciones para un horario semanal de 9 empleados bajo el modelo Cíclico como el mostrado en la Figura 2.

Employee/day	Mon	Tue	Wen	Thu	Fri	Sat	Sun
1	D	D	A	A			
2	A	A	A	N	N	N	
3		D	D	N	N	N	
4		A	A	A	A		
5			D	D	D	D	D
6	D			D	D	D	N
7	N				A	A	N
8	N	N				A	A
9	A	N	N				

Figura 2. Posible horario de un modelo cíclico con bloques de trabajo 46546555

Donde los turnos se definen como:



D: Day (Día)

A: Afternoon (Tarde)

N: Night (Noche)

Por tal motivo existen algunas restricciones que se denominan cambios prohibidos como (N-D), (N-A), (A-D), por motivos de designación de carga laboral y horarios adecuados de trabajo. Para ello el objetivo fundamental es la creación de los posibles bloques semanales que cumplan con las restricciones laborales mencionadas; (Nysret et al., s. f.) Implementaron un pseudocódigo que permite determinar las posibles soluciones que puedan satisfacer las restricciones de manera eficiente.

### ▪ No-Cíclico

En este modelo de planeación la organización del horario no trabaja directamente bajo un patrón de diseño establecido, por el contrario, intenta satisfacer la mayoría de las preferencias individuales de los empleados que deben cumplir turnos de trabajo rotativo (Nysret et al., s. f.).

Algunos de los métodos para la generación de horarios de trabajo rotativo bajo un modelo no cíclico es la implementación del algoritmo Greedy (voraz) y Greedy Aleatorio, el cual busca de manera iterativa encontrar la mejor solución local, es decir tras cada iteración busca asignar el horario que permita satisfacer las reglas establecidas. Dichas reglas vienen dadas por las necesidades de los empleados y las restricciones generadas por el Código Sustantivo del Trabajo. Este método puede ser modificado parcialmente bajo la introducción de parámetro de aleatoriedad, donde se obtiene un valor bajo una distribución uniforme y se establece un umbral mínimo que indique la selección de solución óptima, en caso contrario, donde el umbral no logre ser sobrepasado se selecciona de manera aleatoria la asignación de los horarios en dicha iteración, con el fin de buscar una solución óptima general y diferente a la local.

Estos métodos son computacionalmente costosos, puesto que el número de combinaciones por empleado y asignación representan un espacio de probabilidades considerables, donde muchas de las iteraciones ejecutadas pueden no generar una solución general adecuada.

Día	L	M	X	J	V	S	D
Empleado 1	N	N					
Empleado 2		D	D	D			
Empleado 3			T	T	T		
Empleado 4							

Figura 3. Ejemplo de esquema de un horario demandado por el personal operativo, donde se espera el cumplimiento mínimo de las asignaciones que se presentan



Como se puede ver en la Figura 3, en un modelo no-cíclico, donde se pretende satisfacer las necesidades de los empleados existen asignaciones de horarios pre-existentes, es decir, dichas asignaciones deberán respetarse al igual que las reglas generales establecidas por el empleador (empresa) y las reglas definidas por el Código Sustantivo del Trabajo, de esta forma se debe lograr completar la asignación de horario semanal con las restricciones mencionadas.

Una solución al problema puede ser la implementación del algoritmo Greedy, de esta forma se construye un grafo jerárquico, donde se puede solucionar a partir del primer día y seleccionando la solución de óptimo local, o implementando el parámetro de aleatoriedad para verificar soluciones adicionales. Este algoritmo se puede ejecutar múltiples veces con el objetivo de descartar soluciones inadecuadas y almacenar dichas combinaciones con el objetivo de no desarrollar los mismos recorridos sobre el grafo.

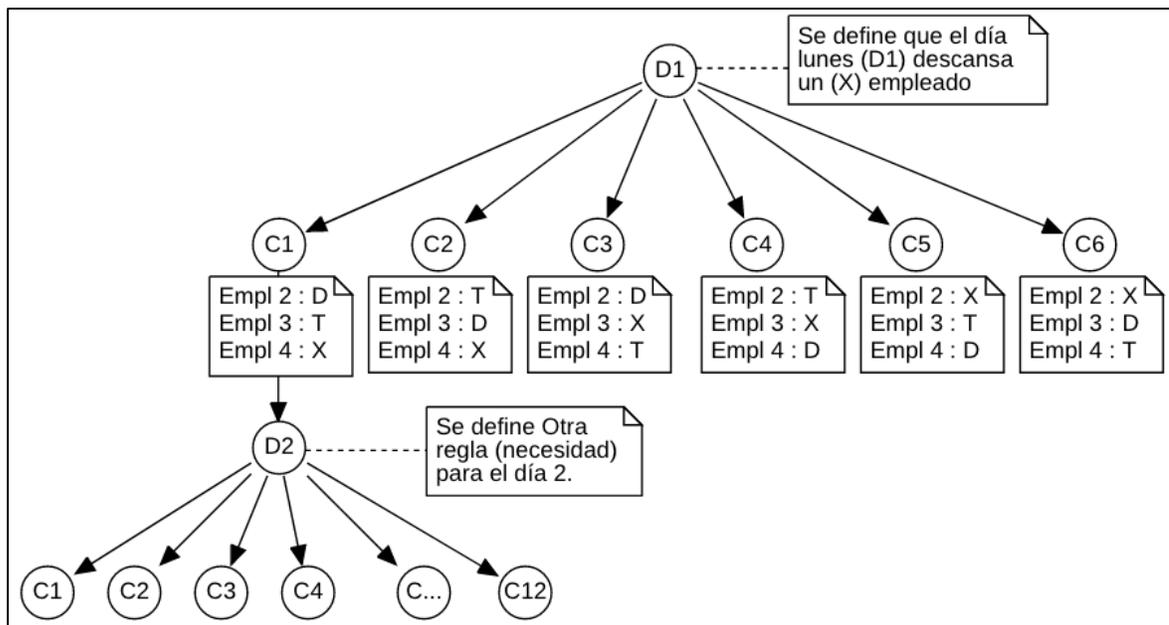


Figura 4. Esquema de implementación del algoritmo Greedy.

En la Figura 4 se puede observar las posibles combinaciones para la iteración que corresponde al día uno (lunes) del problema planteado en la Figura 3. De esta forma es posible evidenciar que el número de combinaciones posibles para lograr la implementación semanal del problema planteado crece significativamente a medida que se aumentan los días de asignación. Por esta razón el algoritmo Greedy resulta costoso en términos computacionales y el modelo no-cíclico se prefiere para casos donde las necesidades del personal son de muy alta prioridad.

Otro de los modelos empleados para la generación de horarios es el modelo de programación multi-objetivos, el cual intenta reducir el número de conflictos y violaciones, maximizando las preferencias del personal (Labidi, Mrad, Gharbi, & Louly, 2014).



- **Lenguajes de Programación**

Los lenguajes de programación presentan una gran variedad de posibilidades frente a los desafíos del mundo actual, es claro que los lenguajes de programación no están diseñados para solucionar problemas específicos de la industria y demás, por el contrario, cada uno de los lenguajes centra su efectividad en la manera cómo es posible solucionar los problemas, por medio de la naturaleza y virtudes que entrega cada lenguaje.

Algunos de los lenguajes más populares es el lenguaje Java, el cual es orientado a objetos, empleado en múltiples aplicaciones a nivel industrial y local; según el autor Pablo Augusto Sznajdleder “Java es un lenguaje de programación de propósitos generales. Podemos usar Java para desarrollar el mismo tipo de aplicaciones que programamos con otros lenguajes como C o Pascal”. (sznajdleder, 2016)

Existen numerosos lenguajes destinados a la industria y su utilización depende de muchos factores, sin embargo, la facilidad de entendimiento de algunos lenguajes como Python permiten obtener gran aceptación frente a los nuevos programadores técnicos, por tal motivo se ha venido implementando como lenguaje de iniciación a la programación para la generación de aplicaciones, tal como lo ha establecido el departamento de tecnologías y computación del instituto tecnológico de Monterrey.(Ramírez, 2010)

- **Bases de Datos**

Las bases de datos son la representación integrada de conjuntos de entidades y sus relaciones, en esencia almacenan las diferentes entidades de los sistemas que hacen uso de ellas con el objetivo de desarrollar persistencia de la información, lo que permite que dicha información, sea utilizable por múltiples sistemas procesos o programas (Camps Paré & Universitat Oberta de Catalunya, 2005). Actualmente las bases de datos pueden ser clasificadas en dos categorías donde se distinguen inicialmente por el modelo de almacenamiento y los datos que contienen, las cuales se presentan a continuación.

- **Estructuradas**

La primera categoría corresponde a las bases de datos **estructuradas**, las cuales almacenan los datos en formatos de tipo tabla y desarrollan relaciones entre estas por medio de las variables contenidas en ellas, para ello se basan en el modelo relacional de Peter Chen (Camps Paré & Universitat Oberta de Catalunya, 2005). Es importante mencionar que en este tipo de bases de datos las tablas generadas son inmutables, es decir que los registros siempre serán descritos por las mismas variables una vez se ha diseñado y desplegado cada tabla.

Algunos de las bases de datos estructuradas más comunes son:



- MySQL
- Oracle
- SQLserver
- IBM DB2

- **No Estructuradas**

La segunda categoría se trata de las bases de datos **no estructuradas**, las cuales almacenan los datos, como su nombre lo indica, de manera *no estructurada* o *semiestructurada*. En este tipo de bases de datos la información no se almacena en formatos tipo tabla y no se agrupan a partir de relaciones como las mencionadas anteriormente; estas bases de datos suelen emplear formatos diferentes y dichos formatos establecen la subcategoría en la que se encuentren. Las cuatro subcategorías en las que se pueden enmarcar dichas categorías son: Key-Value, Column, Document y Graph (Madison, 2015).

A continuación, se presenta el esquema principal de funcionamiento de las cuatro categorías con algunas de las bases de datos más comunes que entran en dicha categoría.

- **Key-Value**

Este tipo de bases de datos almacenan la información en formato de pares clave-valor; donde la clave representa la posición o índice y el valor representa la información a almacenar, de esta forma se puede acceder a la información de manera rápida. Algunos de las bases de datos de tipo clave-valor más destacables son:

- MemcacheDB
- DynamoDB.

- **Column**

Las bases de datos orientadas a columnas permiten el almacenamiento de información en formatos de tipo tabla, similar a las bases de datos estructuradas, sin embargo, estas tienen la capacidad de cambiar el número de variables que describen cada registro, es decir, cada fila dentro de la tabla. Algunos ejemplos de bases de datos orientadas a columnas son:

- Apache Cassandra
- HBase

- **Document**



Este tipo de bases de datos implementan pares clave-valor al igual que las bases de datos de tipo Key-Value. La diferencia fundamental de este tipo de bases de datos es que cada valor es un documento que contiene información semiestructurada o no estructurada. Es importante mencionar que los principales formatos de almacenamiento de dicha información son XML (eXtensible Markup Language) o JSON (JavaScript Object Notation). Las bases de datos más populares del mercado orientadas a documentos son:

- MongoDB
- CouchDB

- **Graph**

Las bases de datos orientadas a grafos almacenan la información sobre entidades, la cual se representa como los nodos dentro de un conjunto (grafo), adicionalmente, una de las características más notorias de estas bases de datos es que la relación entre pares de entidades se almacena y es representada como la unión entre nodos; de esta forma resultan útiles para analizar relaciones entre entidades. Una de las bases de datos orientadas a grafos y más comunes del mercado es Neo4J.



## Metodología

El desarrollo de un sistema de información implica una serie de etapas, las cuales van desde la determinación de los aspectos funcionales del sistema en función de la problemática establecida, hasta el desarrollo e implementación del mismo como modelo de solución viable. Partiendo de esto, el desarrollo del software para la generación y gestión de personal del hospital San Carlos Saldaña se lleva a cabo en múltiples etapas.

En la primera etapa se desarrolla el **Análisis del Hospital**, donde se identifican los lineamientos y necesidades claras, con el fin de comprender la operación real, los parámetros de funcionamiento, las reglas que rigen la generación de los turnos en el hospital San Carlos y así mismo establecer los lineamientos del desarrollo del sistema de información. Adicionalmente en dicha etapa, se estudian los parámetros legales determinados por el Código Sustantivo del Trabajo (CST).

En la segunda etapa, **Diseño del Software**, se realiza el diseño general del modelo, donde se contemplan las entidades más importantes en el hospital, con el objetivo de considerar cada una de ellas en la definición de la arquitectura de solución; adicionalmente, en esta etapa, se realiza la selección y el ajuste del algoritmo a implementar teniendo en cuenta los lineamientos establecidos por el CST y el propio hospital. Es importante mencionar que el modelado general se desarrolla a través de esquemas UML, los cuales resultan ser la solución práctica para definir y diseñar a nivel de arquitectura de software la problemática sin la necesidad de incurrir en el desarrollo del sistema y su programación

La tercera etapa se centra en el **Desarrollo del Software**, en la cual se realiza la selección del lenguaje de programación, contemplando principalmente dos de los lenguajes de programación orientada a objetos más populares, Java y Python, sin embargo, se expresan las razones por las cuales Java es seleccionado para la construcción del software. Otro de los aspectos más importantes que se comprenden en esta etapa es la selección de la base de datos empleada, donde se especifica las razones por las cuales se implementa MongoDB como solución al problema de almacenamiento asociado a este tipo de sistemas. Finalmente se presentan el diseño final de la interfaz gráfica del software, la cual es el único medio de interacción con el usuario final.

Finalmente, la última etapa hace referencia al módulo de exportación, donde se presenta el funcionamiento y detalles generales de esta unidad, la cual soluciona el problema de compatibilidad con herramientas como Excel de Microsoft y Numbers de OSX.



## **Análisis del Hospital**

El hospital San Carlos se encuentra ubicado en el kilómetro 1 vía Purificación, en el municipio de Saldaña, Tolima. La principal actividad y razón de la empresa hospital San Carlos Saldaña es la atención de la salud Humana del 100% de la población del municipio, mejorando continuamente los servicios de prevención de enfermedades y promoción de la salud.

Actualmente el hospital cuenta con un número de 13 empleados directos, de los cuales 9 pertenecen a la unidad de urgencias que corresponde al cuerpo de empleados que opera bajo la modalidad de turnos de trabajo rotativo, por tal motivo la generación del software como producto del presente documento se basa en la modalidad de trabajo para dicho grupo de empleados.

- **Estructura Organizacional**

El hospital San Carlos cuenta con una estructura organizacional lineo-funcional, donde las diferentes dependencias reciben el direccionamiento de todas las actividades desde la junta directiva y la gerencia del hospital, donde adicionalmente cada una de las dependencias se especializa en su única actividad (Martínez, 2011).

Como se observa en la Figura 5, el organigrama del hospital actualmente cuenta con cerca de 16 dependencias que realizan tareas específicas para el funcionamiento de toda la operación y la prestación del servicio de salud para la población de Saldaña, dirigidos por la coordinación médica o la administrativa («Estructura Organizacional - Acerca de la entidad - hospital San Carlos, Saldaña Tolima», s. f.). En la Figura 5, las dependencias marcadas (Color Azul) son aquellas dependencias donde los empleados pueden llevar a cabo el desarrollo de turnos de trabajo rotativos, por tal motivo se observa que no solo la unidad de urgencias puede estar beneficiada a partir de la generación de un sistema de gestión de horarios, por tal motivo en el desarrollo del presente documento no limitará el funcionamiento del sistema a la unidad especificada anteriormente, lo que se verá en la siguiente sección durante el modelamiento y creación del sistema.

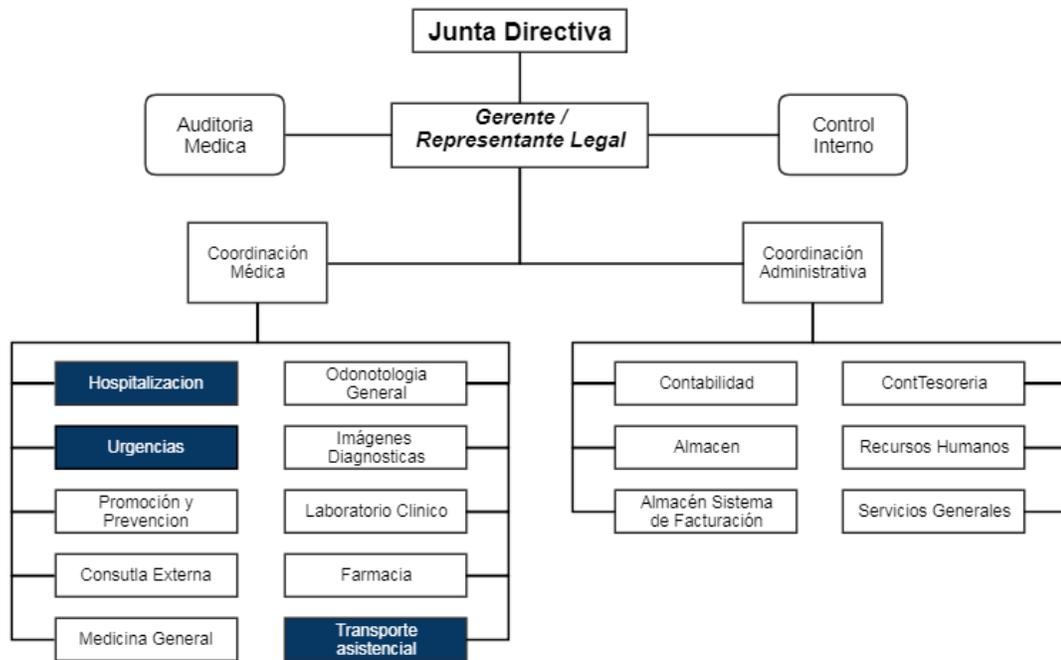


Figura 5. Organigrama del hospital San Carlos Saldaña

Uno de los aspectos que se deben aclarar del porque el software queda centrado en la manipulación, generación y control de turnos del personal operativo de las áreas marcadas es debido a que las demás áreas no requieren de actividad continua y de labores de trabajo interrumpidas, por ende, para este tipo de trabajo algunos de los artículos como lo son el 165 y 166 no tienen aplicabilidad, debido a que su horario de trabajo cumple una jornada típica de 8 horas diarias de lunes a sábado, o con posibilidad de modificación entre los días lunes y viernes para que el día sábado sea día de descanso; esta modificación se rige bajo el artículo 164 del Código Sustantivo del Trabajo.

#### ○ **Parámetros legales (Lo Referente al CST)**

El Código Sustantivo del Trabajo establece cerca de 10 artículos que representan las reglas, procedimientos y excepciones para la correcta disposición de la carga laboral, esta se encuentra medida en horas de trabajo por parte del empleado como servicio prestado hacia el empleador, es decir, que establece los lineamientos legales para la distribución de la jornada de trabajo de los empleados. Es importante destacar que la jornada de trabajo se define como el tiempo invertido en horas de trabajo en función de un periodo de tiempo establecido como unidad de medida, por tal motivo la jornada de trabajo se define como las de trabajo por semanas, por meses o por otra cantidad de días fijos, p.ej. 21 días.



Uno de los artículos de mayor referente para la constitución de la carga laboral medida en horas es el artículo 161 del Código Sustantivo del Trabajo, donde se establece y define la jornada ordinaria de trabajo como una carga de 8 horas por día o 48 horas de trabajo por semana, esta situación establece uno de los principales parámetros necesarios dentro de la definición de carga laboral del software desarrollado, sin embargo es de vital importancia comprender que los artículos posteriores permiten establecer distintas configuraciones laborales en función de las necesidades demandadas por el empleador y aprobadas bajo consentimiento expreso en la contratación por parte del empleado.

Otro de los artículos a tener en cuenta es el artículo 165 del Código Sustantivo del Trabajo, donde se establece que en ciertas situaciones en las cuales no se demanda actividad continua en las labores desarrolladas por el empleado en trabajo por turnos, las horas de trabajo diarias pueden extenderse superando las 8 horas diarias o las 48 horas semanales, sin embargo se deben compensar satisfactoriamente para que en un periodo de tres (3) semanas, el promedio de las horas trabajadas no supere 8 horas por día o supere las 48 horas por semana.

Para establecer los parámetros legales que el software debió tener presente durante la creación y el análisis de los horarios de trabajo de turno rotativos del hospital San Carlos, fue necesario almacenar los límites de horas de trabajo en función de lo pautado en los artículos referentes a las horas de trabajo en el Código Sustantivo de Trabajo, con el objetivo de determinar en qué circunstancias no se respeta el número de horas establecidas para la jornada ordinaria y la jornada máxima de trabajo, así mismo dotar al software de la capacidad de realizar el cálculo de las horas extras durante los ciclos de trabajo definidos. Es importante destacar que el ciclo de trabajo es la longitud en días donde los trabajadores cumplen, en conjunto, un ciclo regular de trabajo antes de realizar cambios para garantizar equidad en la carga laboral de todos los trabajadores.

A partir de lo expuesto por (Valdivieso Rueda, 2012) en el documento basado en el entendimiento los artículos del Código Sustantivo del Trabajo para regular de manera adecuada la carga laboral de trabajos por turnos mencionados en este documento como turnos de trabajo rotativo, se especifican algunas contextualizaciones importantes a tener cuenta en función de los artículos 162 y 166 para la parametrización legal del software generado, las cuales son:

El artículo 162 (CST) regula las condiciones bajo las cuales se puede autorizar laborar horas extras.

El artículo 166 (CST) autoriza el trabajo sin solución de continuidad, estableciendo como día laborable de la semana el día domingo. Lo anterior, en lo que respecta a la jornada de 56 horas, tendría el siguiente efecto:

- Se elimina el día de descanso obligatorio.



- La eliminación del día de descanso obligatorio implica que el día domingo es un día laboral y por lo tanto entraría dentro de la jornada ordinaria de la empresa, a razón de 8 horas diarias por 7 días (es decir, que las 56 horas de la semana son ordinarias).
- Se mantiene el límite máximo de trabajo diario de 8 horas, de tal forma que la hora 9 sería hora extra.
- En la medida que el servicio se requiere sin solución de continuidad, este deberá ser prestado a razón de 3 turnos de 8 horas día. Si el turno por día es superior se causan horas extras.

#### ○ Ciclos de Trabajo y el CST

A continuación, se observa un diagrama donde se ilustra el posible rango del número horas de trabajo de un empleado para un periodo de una semana en función de los artículos que aplican en cada uno de los escenarios, donde se observa lo estipulado por el artículo 161, 165 y 166, así como lo regulado frente a las horas extras por el artículo 22 de la ley 50 de 1990.

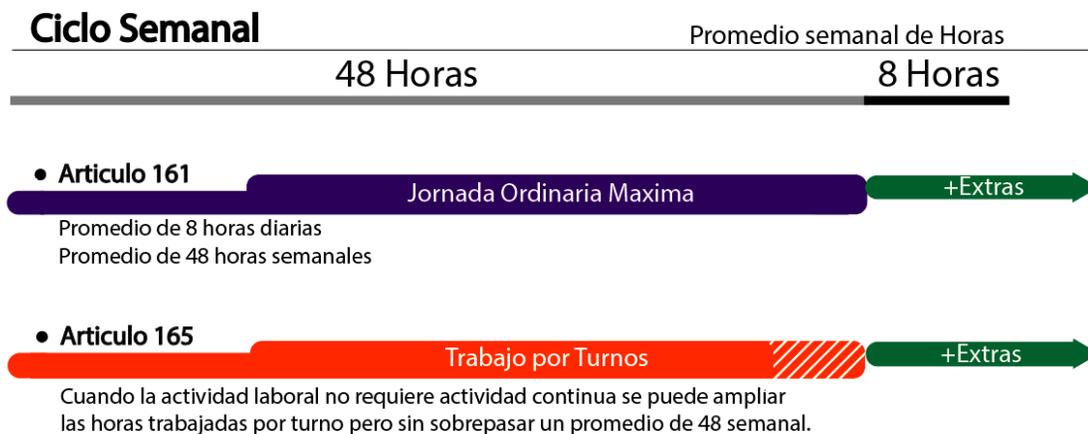


Figura 6. Horas de trabajo por semana en función de los artículos del CST

En la siguiente figura se observan las horas por turnos y los casos donde se aplican las horas extras en función de los artículos, de esta forma es posible observar bajo que artículo se regula el trabajo de los empleados según las horas trabajadas por turnos y el límite de trabajo suplementario.

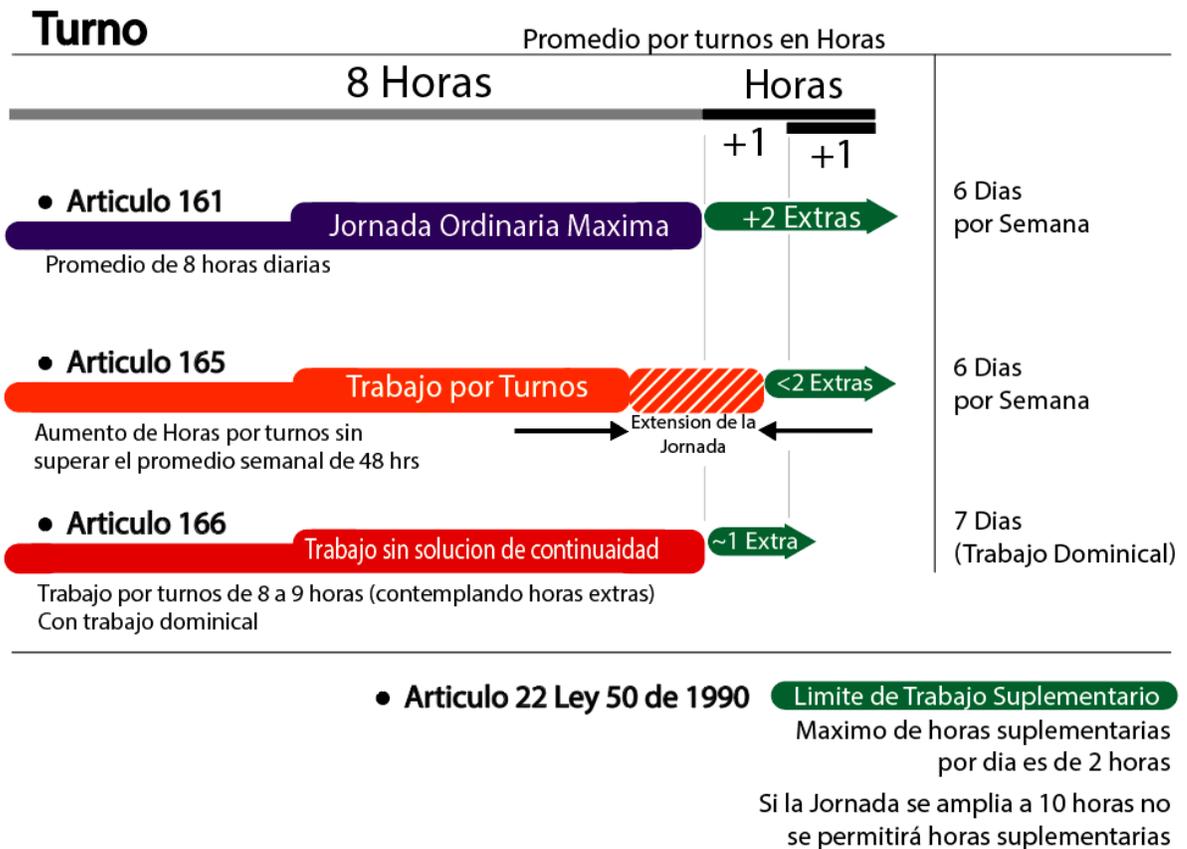


Figura 7. Horas de trabajo por turno en función de los artículos del CST

A partir de las Figuras 6 y 7, queda claro que la jornada ordinaria establece un parámetro inicial de trabajo máximo de 48 horas a la semana y un promedio de 8 horas diarias, sin embargo, se observa cómo es posible bajo el artículo 165 del Código Sustantivo del Trabajo, las horas de trabajo por día puedan superarse, con la salvedad de que el promedio de horas trabajadas a la semana no exceda a 48; por otro lado, se observa el posible número de horas trabajadas bajo el artículo 166, en el cual el número de horas trabajadas por semana puede ascender hasta un máximo de 56, donde el día domingo se contempla como un día laboral de ocho horas, con la novena hora como trabajo suplementario. Adicionalmente, en las Figuras 6 y 7 es importante apreciar cómo se distribuye las horas extras o trabajo suplementario, para con un máximo de 2 horas por día o 12 semanales, con la condición de que si la jornada se amplía a 10 horas por algunos de los artículos mencionados (165, 166) no se permitirá trabajar horas extras en el mismo día. Es importante aclarar que algunos de los artículos especificados en las Figuras 6 y 7 no contemplan horas extras, sin embargo, en la práctica es posible que se requiera de horas de trabajo suplementario por motivos de fuerza mayor, lo que significa que para efectos prácticos en el desarrollo del software, este fue diseñado para registrar y reconocer las horas suplementarias extraoficiales, que incluso pueden sustentarse por el artículo 163 del Código Sustantivo del Trabajo (Ministerio de Protección Social, 2011d).



### ○ **Parámetros institucionales**

EL modelo de generación de horario actual del hospital San Carlos presenta irregularidades, puesto que, tras analizar su funcionamiento se determinó que se basa en un modelo de trabajo por turnos de tipo no cíclico, donde se busca garantizar las preferencias de los trabajadores con periodos de trabajo rotativos sin ciclos definidos, sin una regla fuerte que permita disminuir las inconsistencias y colisiones de turnos para distintos empleados (generalmente del cuerpo de enfermeros).

Por la situación documentada en el párrafo anterior, la gerencia del hospital se encontraba en la necesidad de establecer un sistema para la manipulación adecuada de los horarios de turno de trabajo y el registro adecuado de las horas totales, esto, con el objetivo de realizar un grupo específico de actividades adicionales que facilitarán y agilizarán el proceso de gestión del hospital con lo que respecta a sus empleados y la carga horaria de los mismos.

Para determinar con exactitud las necesidades del hospital se desarrollaron un grupo de socializaciones entre los integrantes del proyecto y la gerencia del hospital durante un periodo de 3 días del mes de julio del 2018. Según las socializaciones, así como el entendimiento del funcionamiento del hospital y gestión del mismo, se encontraron 2 aspectos fundamentales que debían ser tratados y gestionados a partir del desarrollo del software y el cumplimiento de los objetivos del presente proyecto. El primero de los aspectos es el planteamiento de un modelo cíclico adecuado para el número de empleados de las unidades que trabajan bajo el concepto de trabajo por turnos que permita establecer carga de trabajo igualitaria entre los empleados, aspecto que el software deberá contemplar para asignar de manera adecuada los horarios del personal analizado; el siguiente aspecto es el cálculo de las horas trabajadas de todos los empleados, con el objetivo de evidenciar incumplimientos en los artículos de Código Sustantivo del Trabajo y facilitar el cálculo de las horas totales trabajadas mes a mes, lo que le permita al administrador encargado desarrollar fácilmente el cálculo de los montos a liquidar por empleado (No se incluye un módulo para el cálculo de dicho monto por efectos de la variedad en la contratación y aspectos de parafiscales no contemplados en este proyecto).

A pesar de que el sistema deberá cumplir con los dos requerimientos mencionados anteriormente y demandados por la gerencia del hospital, se analizaron algunos aspectos adicionales a tener en cuenta en cada uno de requerimientos, los cuales son:

- Posibilitar cambios en el horario de personal por parte del administrador del sistema, con el objetivo de registrar las horas de trabajo reales desarrolladas por el personal.



- Posibilitar el cálculo del total de horas desarrolladas por cada uno de los empleados por mes y por semana, contemplando los cambios repentinos en el horario a causa de permisos e incapacidades.
- Prevención de incumplimientos de lógica secuencial del horario y alertas en la violación de los artículos del Código Sustantivo del Trabajo cuando se realicen cambios.

#### ○ **Modelo operativo antiguo del Hospital**

Previo al desarrollo del presente trabajo, el hospital San Carlos desempeñaba sus labores de generación de turnos de trabajo a través de un modelo no cíclico, sin el cumplimiento de una regla que permitiese equilibrar la carga laboral y evitar incurrir con incumplimientos del Código Sustantivo del Trabajo. Este modelo de generación de turnos rotativo se desarrollaba mediante el balanceo de las horas semanales trabajadas entre los empleados, donde se diseñaban múltiples periodos de trabajo con duración exacta de una semana. Uno de los mayores inconvenientes encontrados bajo esta metodología, es que en ocasiones generaba sobrecarga laboral en algunos de los empleados frente a otros durante el transcurso de las semanas, debido a que el intento por equilibrar las horas semanales y los días de descanso de los trabajadores se dificulta a medida que se avanza y se deben asignar compensaciones de descanso cuando uno o más trabajadores trabaja de manera desequilibrada en días anteriores. Otro de los inconvenientes relacionados con la ausencia de horarios establecidos y equilibrados a largo plazo, es la modificación del mismo por parte del coordinador de los empleados, donde los días de permiso o incapacidades afectan fuertemente el balanceo de las horas semanales entre los diferentes empleados.

El siguiente diagrama enseña un periodo comprendió de cuatro semanas para cuatro empleados anonimizados del hospital San Carlos entre el 5 de noviembre de 2018 y el 30 del mismo mes, donde se puede observar los turnos de trabajo de los empleados a lo largo de todo el periodo de trabajo.



Semana 1							
	L	M	Mi	J	V	S	D
Empleado 1	D	D	D	D	T	X	T
Empleado 2	X	T	T	T	Perm	D	D
Empleado 3	T	A	A	A	A	A	X
Empleado 4	A	X	S	S	T	S	A

Semana 2							
	L	M	Mi	J	V	S	D
Empleado 1	T	T	T	T	T	D	D
Empleado 2	D	D	D	D	A	A	X
Empleado 3	A	A	A	Inc	Inc	T	T
Empleado 4	S	S	S	T	D	X	A

Semana 3							
	L	M	Mi	J	V	S	D
Empleado 1	X	T	A	A	A	A	X
Empleado 2	D	D	T	T	X	D	D
Empleado 3	T	X	S	S	T	T	T
Empleado 4	A	A	D	D	D	X	A

Semana 4							
	L	M	Mi	J	V	S	D
Empleado 1	T	S	S	S	X T	T	T
Empleado 2	A	A	A	A	A	X A	A
Empleado 3	X	T	T	T	V	V	V
Empleado 4	D	D	D	D	D	D	X D

Figura 8. Horario de trabajo desarrollado por el hospital San Carlos en el periodo comprendido entre 05/11/18 y 02/12/18

En el siguiente diagrama se observa un recuento de horas y tipos de turnos desarrollados a partir de las asignaciones trabajo que se enseñan en la Figura 8.

Recuento	X	V	Perm/inc	D	T	A	Total Horas
Empleado 1	4	0	0	5	11	4	160
Empleado 2	4	0	1	10	5	8	184
Empleado 3	3	3	2	2	10	8	160
Empleado 4	4	0	0	16	2	6	192

Figura 9. Recuento de Horas trabajadas y tipos de turnos desempeñados por los trabajadores en el periodo comprendido entre 05/11/18 y 02/12/18.

Como se pudo observar en las figuras anteriores, el modelo empleado por el hospital San Carlos no estandariza el proceso de generación de horarios y genera un desbalance claro en un periodo de trabajo cercano a un mes. Donde el equipo de coordinación del hospital busca balancear el número de horas trabajadas durante periodos de una semana y que los días de descanso para los trabajadores queden asignados los fines de semana o días cercanos a estos. Como se evidencia en la Figura 9 el número de horas trabajadas por los empleados presenta un desbalance evidente.

Claramente el modelo empleado por el hospital San Carlos para la generación de los turnos rotativos de los empleados, corresponde a una implementación del algoritmo greedy para un modelo no cíclico, donde se busca satisfacer de manera semanal el problema de equilibrio horario, sin contemplar la distribución del tipo de turnos desarrollados por cada uno de los empleados. Es decir, que la premisa que plantea el algoritmo greedy (algoritmo voraz) de solucionar de manera local el problema, se aplica durante la selección de turnos por día y el número de horas totales por cada semana, donde no se contemplan ciclos más extensos, lo que propicia el desarrollo de fallos en el equilibrio del número de horas del



trabajo cuando se avanza semana a semana y se debe organizar y compensar nuevamente los días de descanso, lo que resume el problema en una falta de organización con la necesidad de priorizar la compensación de días de descanso, cuando se tienen, sobre el equilibrio semanal de horas de trabajo.

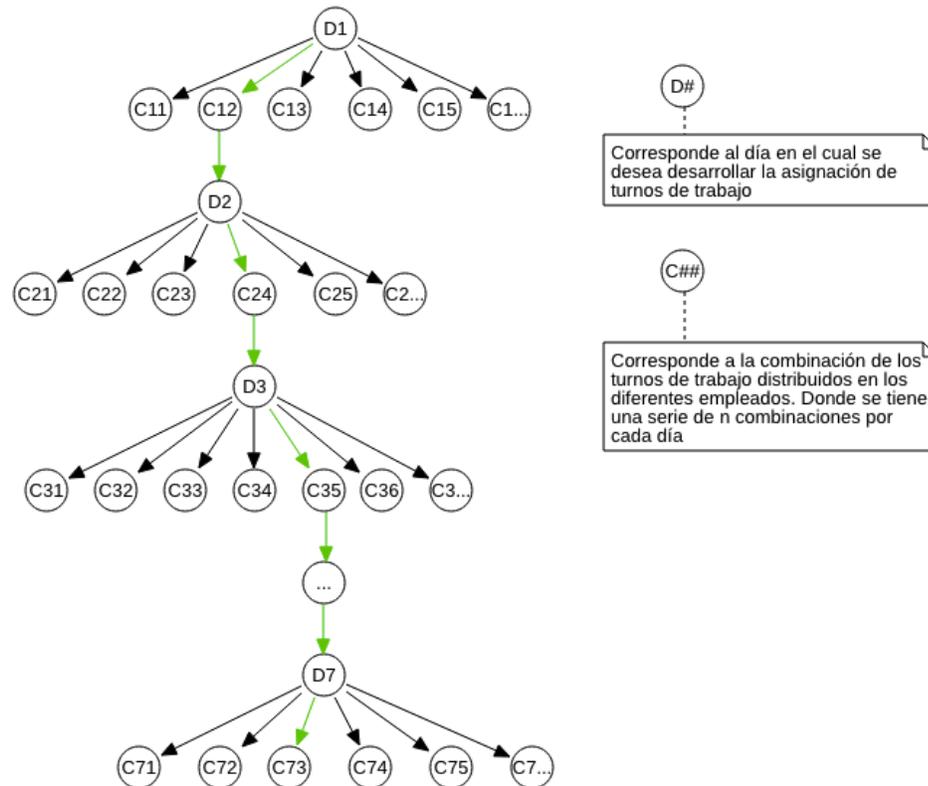


Figura 10. Esquema de selección de posibilidades bajo el modelo antiguo del hospital.

Del esquema planteado en la Figura 10 se ejemplifica un posible conjunto de combinaciones a través del camino señalado (en verde), donde se sabe que:

- $D\#$ : Corresponde al día en el cual se desea desarrollar la asignación de turnos de trabajo.
- $C\#\#$ : Corresponde a la combinación de los turnos de trabajo distribuidos en los diferentes empleados.

Este planteamiento permite definir un cierto conjunto de variables que pueden ser empleadas para dar una definición matemática del problema que antiguamente se debía resolver en el hospital San Carlos. La primera de ellas es  $i$ , la cual corresponde al día de la semana, definida en el esquema anterior como ( $D\#$ ); la siguiente variable es  $k$ , la cual corresponde al número del empleado del conjunto ( $K$ ) de empleados que se desea modelar; y finalmente  $T_k^i$  que corresponde al número de horas de trabajo desarrolladas por el empleado  $k$  durante el día  $i$ .



De esta forma es posible, en términos estadísticos y a través de métodos de optimización, minimizar la varianza entre el total de horas sumadas a la semana por cada empleado, contemplando el conjunto de restricciones que corresponden a los días de descansos y compensaciones. Lo anterior podría responder el problema que antiguamente se enfrentaba el hospital sin dejar la implementación de un modelo no cíclico, donde un planteamiento matemático del problema quedaría como:

$$(var(hs_k)) \quad (1)$$

Donde:

$$hs_k = \left( \sum_{i=0}^n T_k^i \right) \quad (2)$$

$$\underline{hs} = \frac{1}{K} \sum_{k=0}^K hs_k \quad (3)$$

$hs_k$ : Corresponde al total de horas trabajadas por semana para el empleado  $k$ .

$\underline{hs}$ : Corresponde al promedio de horas semanales trabajados por el grupo  $K$  de empleados.

Lo que finalmente la ecuación (1) queda:

$$\left( \sum_{k=0}^K (hs_k - \underline{hs})^2 \right) \quad (4)$$

Donde claramente el objetivo por parte del personal de coordinación durante la creación del horario era dar solución a la expresión anterior sin el uso de ninguna técnica definida y contemplando las restricciones mencionadas, donde adicionalmente no se había planteado una definición formal del problema. Por esta razón el modelo base del hospital corresponde a una implementación del algoritmo greedy mediante programación no cíclica, donde existe un fuerte énfasis en el equilibrio horario y descansos en los días cercanos a los sábados y domingos.



## Diseño del Software

Con el objetivo de generar un software capaz de suplir con los requerimientos funcionales demandados por la institución (Hospital San Carlos Saldaña), se debe modelar y diseñar una estructura a partir del organigrama empresarial de la institución y su modelo de funcionamiento; donde se pueda evidenciar las características mínimas necesarias para dar inicio a la programación orientada a objetos y permita generar un producto como medio de solución al problema planteado.

El diseño de un software se puede soportar en múltiples componentes y tecnologías, sin embargo, se hace uso únicamente de aquellos elementos necesarios para la generación de los servicios que cumplan con los requerimientos funcionales previamente establecidos. El software desarrollado como modelo de solución a la problemática planteada en el presente documento hace uso de tecnologías centradas en el manejo de información, entornos de desarrollos y componentes adicionales que facilitan el desarrollo e integración de los diferentes componentes que componen la solución.

### o Modelado

El hospital cuenta con una estructura operativa como la descrita en la Figura 11, donde existen 3 áreas en las cuales se desarrollan turnos de trabajo de tipo rotativo y por los cuales se requería del desarrollo de un sistema o software capaz de gestionar los horarios de los empleados pertenecientes a dichas áreas. A partir de las entrevistas desarrolladas al cuerpo administrativo y operativo del hospital, se desarrolló un diagrama que permite ejemplificar cómo se organizan y distribuyen los empleados en relación a las áreas. Cabe destacar que el modelo oficial de la empresa corresponde al organigrama definido en la Figura 5, y el diagrama de la Figura 13 es diseñado desarrollado como un esquema base para la generación de clases y objetos durante el desarrollo del software y los diagramas UML necesarios.



Figura 11. Modelo base para la creación de diagramas UML

Como se puede evidenciar en el diagrama anterior, cada empleado pertenece a un grupo y este a su vez a un área en particular, bien sea Hospitalización, Urgencias, Transporte Asistencial e incluso servicios generales. A partir del modelo base se estableció que el software debía segmentar al menos las categorías definidas como: **grupo** y **empleado**, ya que el **área** resulta indiferente para el software y se puede modelar a través de una característica única de la entidad grupo; otra de las razones por lo cual se decide modelar únicamente las dos entidades mencionadas es porque no existen grupos que pertenezcan



a múltiples áreas, lo que supone que un grupo será definido siempre por el nombre de la única área al que pertenece.

La entidad **empleado** se define claramente para hacer referencia a cada uno de los individuos que pertenecen al cuerpo de enfermeros, el equipo de hospitalización y demás áreas identificadas en la Figura 5; por otro lado la entidad **grupo** hace referencia al conjunto de empleados, modelados por la entidad **empleado**, que desarrollan las proyecciones de turnos de trabajo de tipo rotativo entre sí, lo que permite definir diversos conjuntos que cumplen su propio modelo y secuencia de rotación de turno sin intervenir en otros grupos, dando como resultado un aspecto esencial de funcionamiento para dar solución al problema planteado.

Tanto la entidad “**Grupo**” como la entidad “**Empleado**” presentan características únicas que las definen, lo que en términos de programación orientada a objetos (POO) hace referencia a que son definidas a través de atributos y funcionalidades, recibiendo el nombre de “Clase” en lugar de entidad; esta situación permite que una clase sea definible en programación a través de lenguajes específicos y dote a los sistemas o programas de la capacidad de crear instancias de dichas clases para su manipulación, dando como resultado programas que modelan internamente aspectos esenciales del mundo real, de tal manera que se procesan los datos que conforman las instancias de clase a través de una máquina (Ordenador, Servidor, etc) capaz de interpretar el lenguaje orientado a objetos.

En términos de diseño y bajo el apartado anterior, las entidades del hospital (**Grupo** y **Empleado**) son clases modeladas a través de la programación, donde deben ser definidas por un conjunto de atributos, que en esencia pueden ser nuevas instancias de otras clases, las cuales definen aspectos imprescindibles del mundo real a ser modelados a través de la programación; dando como resultado un modelo de 2 o más clases. Partiendo de lo anterior y bajo la problemática a resolver, existen otros aspectos necesarios que pueden ser modelados por otras clases, entre ellos los turnos que deberán desarrollar los empleados y las fechas en las que el algoritmo de cada grupo opera. Como resultado final se diseña una estructura que modela las características mínimas de los aspectos anteriormente mencionados, la cual se puede observar en la Figura 12.

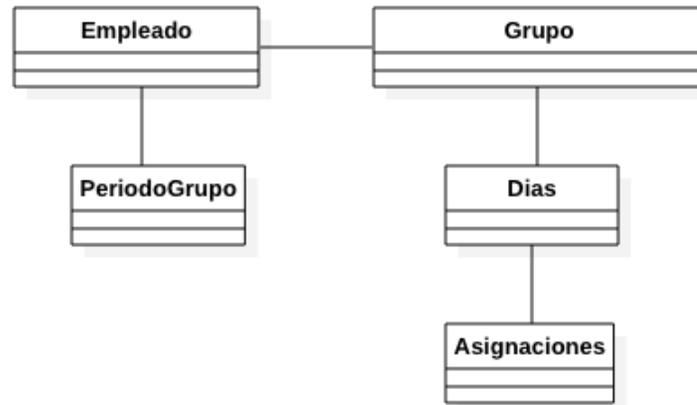


Figura 12. Modelo de las Clases identificadas para el funcionamiento del software

Como se puede observar en la Figura 12 existen 3 clases adicionales a las comentadas anteriormente; **PeriodoGrupo**, **Días** y **Asignaciones**. Estas clases o entidades se establecen como necesarias bajo la perspectiva de diseño establecida, donde la clase **Grupo** además de modelar aspectos esenciales como su nombre y el patrón de asignación de turnos, debe contener de manera imprescindible las fechas en las cuales se desarrolla la asignación de turnos para el conjunto de empleados que lo conforman; para el caso de los empleados (Clase **Empleado**) se debe modelar los periodos (fechas) a los cuales se encuentran asignados a uno o más grupos, es decir, los rangos de fechas en los cuales no se puede desarrollar asignaciones de turnos adicionales durante la creación de nuevos grupos puesto que generaría conflicto de asignación de turnos.

Como se mencionó en el párrafo anterior, un **Grupo** modelado por el software siempre tiene una fecha de inicio y una fecha de finalización (Por defecto se asumió tres meses de vigencia), y entre dichas fechas deberá almacenar los días en los cuales se desarrollan las asignaciones de turnos. Para modelar dichos turnos y los días se emplea las clases denominadas como **Días** y **Asignaciones**, donde la clase **Asignaciones** corresponde a aquella que modela el turno asignado a un empleado en particular, es decir, el tipo de turno (Nocturno, Diurno, Descanso, ect), la hora de inicio y hora finalización. Para el caso de la clase **Días**, se emplea como medida de solución para modelar al conjunto de asignaciones de un día en particular entre las fechas de vigencia del grupo al que pertenecen. En esencia y describiendo estructuralmente el diagrama de la Figura 12, un grupo (clase **Grupo**), se encuentra conformado por un conjunto de empleados (clase **Empleado**), y una lista de días (clase **Días**) que corresponde al periodo de tiempo donde será vigente; para cada uno de los días se crean asignaciones de turnos (clase **Asignaciones**), que modelan los horarios de entrada, salida, descanso y vacaciones de cada uno de los empleados registrado en el grupo.

Ya que cada grupo se encuentra conformado por un par de fechas que definen el periodo de vigencia y operación, es posible determinar los periodos de ocupación de cada empleado. Para ello se emplea la clase **PeriodoGrupo**, la cual recibe el par de fechas de



grupo al que cada empleado pertenece y lo asigna internamente a una lista de Periodos del empleado en cuestión, de esta forma se gestiona los periodos de ocupación para evitar inconsistencias por asignación de turnos múltiples en fechas iguales.

- **Arquitectura de Solución**

Para desarrollar software de manera efectiva y bajo un esquema formulado se suele implementar diferentes estilos de arquitectura que permiten dar solución a los problemas de manera segmentada, seccionando los aspectos principales del software a construir. Uno de estos estilos de arquitectura es el Modelo-Vista-Controlador o MVC; este estilo de arquitectura pretende separar y seccionar tres áreas importantes de un software.

- **Modelo**

El primero de ellos es el modelo, el cual corresponde a la abstracción del problema que se pretende trabajar con ayuda del software, para el caso del gestor de turnos desarrollado en el presente, equivale al mismo modelo que se muestra en la Figura 11 (empleado-grupo), donde las clases que allí se mencionan serán la abstracción del problema a modelar del mundo real.

- **Vista**

La segunda área que se desea segmentar bajo el esquema del MVC, es justamente la Vista, que equivale a la interfaz gráfica con la que el usuario del software pretende interactuar, claramente esta sección se centra en aspectos de diseño y pretende resolver problemas de entendimiento y facilidad de uso, siendo un aspecto que se documenta más adelante.

- **Controlador**

Finalmente, la última área que se pretende seccionar bajo el estilo de arquitectura MVC es el controlador, el cual se encarga parcial o totalmente de gestionar todos los aspectos lógicos del software, incluyendo la manipulación del modelo para generar instancias de grupos que le permitan operar y desarrollar las acciones que den solución a los requerimientos funcionales planteados para el software. En el caso del gestor de turnos esta sección se encarga de desarrollar múltiples acciones, entre ellas la aplicación de las restricciones lógicas de horarios, de generación, manipulación de empleados, grupos y los horarios asociados a los mismos; de igual manera esta sección del software se encarga de alertar la violación del Código Sustantivo del Trabajo y aplicar los modelos de generación de horarios que se comentarán en secciones posteriores. Otro de los aspectos más importantes de los que se encarga esta sección para el software de gestión de turnos en desarrollo de la conexión y ejecución de operaciones CRUD (Create, Read, Update, Delete) sobre la base de datos en la que se soporta el proyecto.



## ○ UML

Para definir de manera gráfica los componentes principales el software se desarrolló un diagrama UML que permite contemplar los tres componentes del patrón de diseño MVC (Modelo-Vista-Controlador). De esta manera cada uno de los componentes y clases diseñadas resuelven problemas a nivel técnico y de infraestructura del software, así como las necesidades demandadas por el hospital y los requerimientos legales.

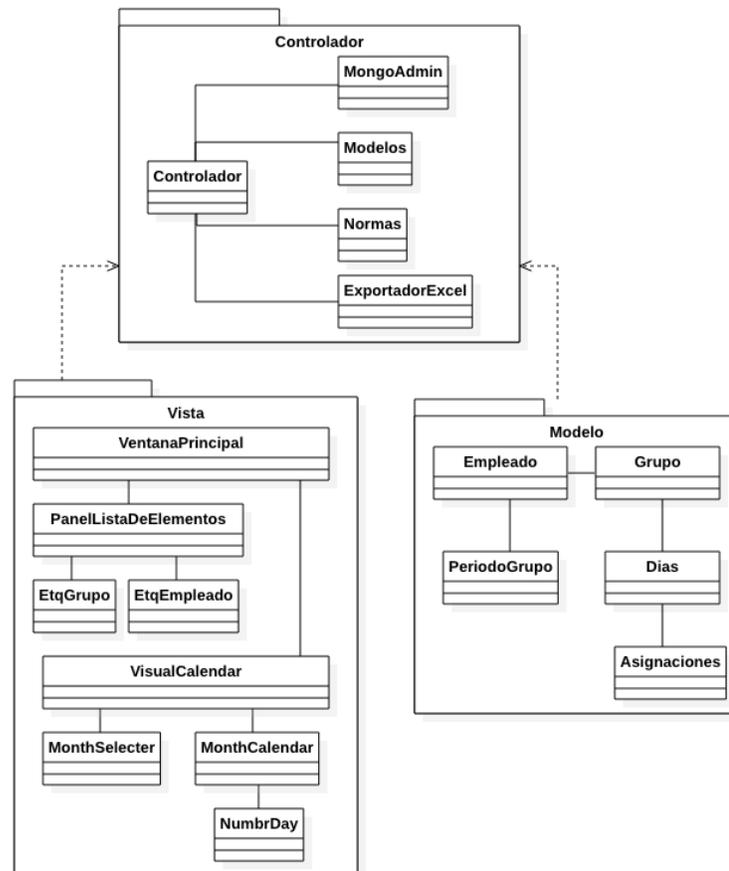


Figura 13. Diagrama UML del software de gestión de turnos bajo el patrón de arquitectura MVC

Como se puede observar en la Figura 13, el diagrama UML que representa el “Modelo” del patrón de diseño MVC cuenta con 5 clases, donde cada una tiene un conjunto de parámetros y métodos necesarios que le permitan al controlador interactuar con las instancias que reflejan el problema planteado.

Claramente la clase con el mayor número de métodos es la clase “Grupo”, esto se debe a que existen múltiples acciones a desarrollar sobre cada grupo, incluyendo validaciones, lectura y escritura de datos, así como modificaciones de fechas de operación. Esta situación se ve reflejada de igual manera en las demás clases, sin embargo, cabe destacar



que la Clase “PeriodoGrupo” es empleada para facilitar el control de los periodos de actividad de cada grupo, donde se pueden realizar consultas a partir de fechas.

Para dar claridad a los aspectos más importantes del software, se desarrolla una explicación de cada una de las clases contenidas en el componente llamado “Modelo”, donde es posible visualizar sus atributos y métodos. Sin embargo, para un mayor entendimiento del software y los diagramas UML desarrollados, se anexa al presente documento el código del programa con su respectiva documentación, mediante comentarios desarrollados en IDE (Entorno de desarrollo), así como los archivos referentes a los diagramas UML.

A continuación, se listan y mencionan los aspectos más importantes de cada clase del “Modelo” con su respectivo diagrama de clase.

- **Empleado:**

La clase “Empleado” se encarga de modelar y contener los aspectos más relevantes de la información asociada a un empleado, como lo es el nombre, sexo, cargo al interior del hospital, profesión, entre otros. Adicionalmente existe 2 atributos de alta importancia denominados “\_id” y “periodosGrupo”; el primero de ellos es empleado para almacenar de manera única un identificador con el cual será asociado cada empleado en la base de datos (MongoDB); el segundo atributo contiene una lista de los diferentes periodos de pertenencia de los distintos grupos generados para modelar la organización del personal operativo del hospital. El atributo “periodosGrupo” contiene instancias de la clase “PeriodoGrupo”, la cual será mencionada y explicada más adelante.

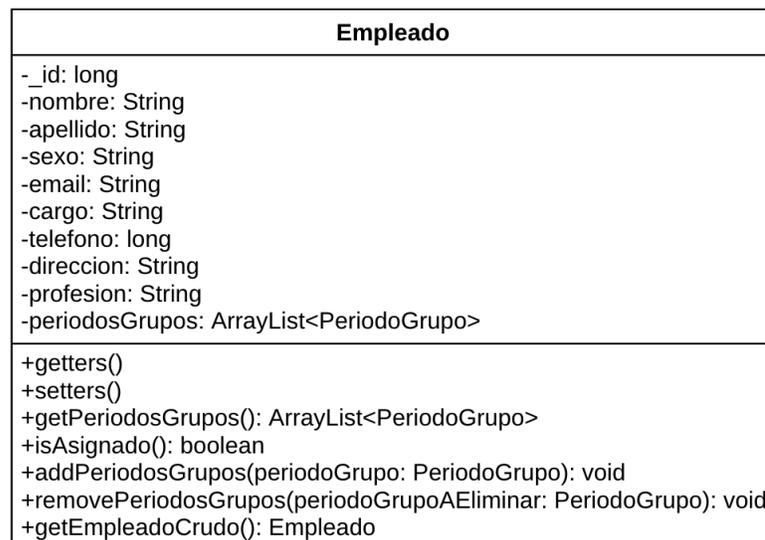


Figura 14. Diagrama de la clase Empleado



- **Grupo:**

Esta clase modela todos los aspectos asociados a la agrupación y organización de los empleados (ver Figura 15), donde se almacenan atributos como el nombre del grupo “nombre”, la fecha de inicio y fecha de finalización del grupo en “fechaInicio” y “fechaFinal” respectivamente, así mismo se almacena aspectos asociados al algoritmo y los días de inicio, sabiendo que es posible que la fecha de inicio asociada al día de la semana puede dar inicio a diferentes combinaciones según dadas por el algoritmo.

La clase “Grupo” Claramente contiene dos atributos adicionales los cuales hacen referencia a la lista de empleados que los conforman y una lista de las instancias de la clase “Dia”, las cuales a su vez almacenan las asignaciones de los turnos.

Grupo
-nombre: String -fechaInicio: String -fechaFinal: String -nombreModelo: String -diaInicioAlgoritmo: int -listaEmpleadosCrudos: ArrayList<Empleado> -diasGrupo: ArrayList<Dia>
+Grupo(nombre: String, listaEmpleadosCrudos: ArrayList<Empleado>, fechaInicio: Date, fechaFin: Date, nombreModelo: String, numDiaInicioAlgoritmo: int) +getNombre(): String +getListEmpleadosCrudos(): ArrayList<Empleado> +getFechaInicio(): Date +getFechaFinal(): Date +getNombreModelo(): String +getDiaInicioAlgoritmo(): int +getDiasGrupo(): ArrayList<Dia> +setDiasGrupo(diasGrupo: ArrayList<Dia>): void +getPosicionEmpleado(empleado: Empleado): int +isActivo(): boolean +getPeriodoGrupo(): PeriodoGrupo +getFechas(): String +getDiasMatrizCalendario(fechaActualDelInicio: Date): ArrayList<Dia> +getDiaByFecha(fecha: Date): Dia

Figura 15. Diagrama de la clase Grupo

- **PeriodoGrupo:**

Las instancias de clase son contenidas únicamente por instancias de la clase empleado, y hacen referencia a los periodos en los que un empleado (el que las contiene) pertenece a un grupo en el periodo de actividad de dicho grupo. Esta clase facilita el control de asignaciones de un empleado a múltiples grupos en periodos de tiempo donde puede existir colisiones, de esta forma se garantiza una pertenencia a un grupo a la vez para cada uno de los empleados. Esta clase solo requiere del nombre del grupo y de las fechas de inicio y fin que definen al grupo que pertenece el empleado que contiene la instancia de esta clase.

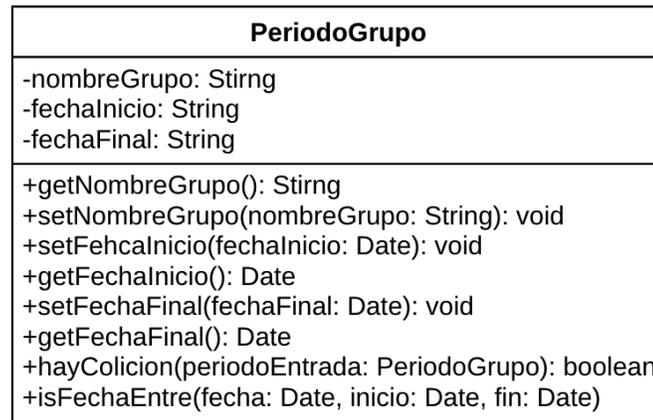


Figura 16. Diagrama de la clase PeriodoGrupo

- **Dia:**

La clase “Dia” es empleada para almacenar las asignaciones de los turnos de cada uno de los empleados, contenidas por el primer atributo de la clase. Es importante mencionar que las asignaciones contenidas en esta clase hacen referencia a los turnos de cada uno de los empleados contenidos por una instancia de la clase “Grupo” en el mismo orden que fueron asignados. Las instancias de esta clase son contenidas directamente por instancias de la clase “Grupo” y son generadas de tal forma que representan los días operativos de los grupos. El segundo atributo hace referencia al día del algoritmo, es decir, la posición en el conjunto de combinaciones de turnos formados tras la ejecución total del algoritmo sin repetir ciclos.

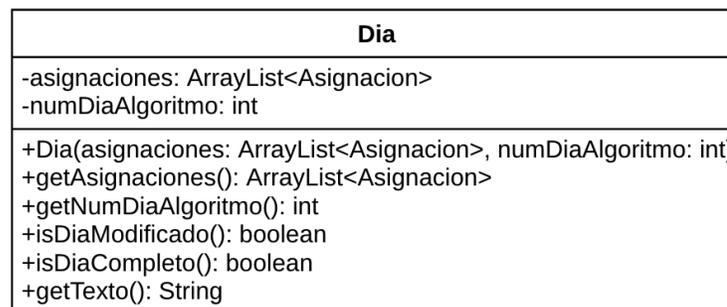


Figura 17. Diagrama de la clase Día

- **Asignación:**

La clase “Asignación” permite modelar y contener los aspectos que definen un turno de trabajo, las instancias de esta clase siempre son contenidas por los días que representan el periodo de trabajo de cada Grupo. Como se puede observar en el diagrama UML de la Figura 13. Es importante mencionar que esta clase contiene dos atributos que almacenan el tipo de turno a ejecutar (ver Figura 18), donde el primero, denominado “turnoOriginal” almacena el turno del empleado generado durante la aplicación del algoritmo y el segundo, denominado “turno”, almacena el tipo de turno que se ha asignado tras modificaciones o



no, de esta forma es posible regresar el tipo de turno original y recuperar asignaciones si tener que ejecutar el algoritmo nuevamente.

Esta clase permite almacenar la hora de inicio y el tiempo invertido durante el desarrollo del turno, en los atributos “horaln” y “tiempoAsignacion” respectivamente. Adicionalmente cuenta con una serie de métodos que permiten modificar cada uno de los atributos de manera directa o indirecta.

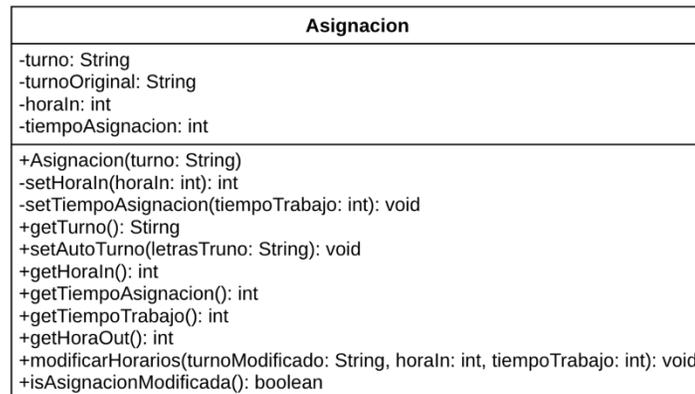


Figura 18. Diagrama de la clase Asignación

### ○ Alcances y limitaciones del software

- El inicio de las secuencias de rotación es parametrizado por la fecha en la que se define el inicio de un grupo.
- Cada empleado pertenece única y exclusivamente a un Grupo dentro del periodo de tiempo que define al grupo.

### ○ Algoritmo de Turnos Rotativos

El algoritmo de generación de horarios se desarrolla a partir de los días de descanso que son asignados a cada empleado. Por tal motivo se plantean los días viernes, sábado, domingo y lunes como hábiles para descanso, cumpliendo con la condición de asignación de descanso próximo a los fines de semana para el conjunto de empleados.

Durante la socialización y entendimiento del problema mediante las entrevistas desarrolladas en el hospital San Carlos al personal operativo y de coordinación se aclararon algunos puntos con respecto al número de empleados que se requería por grupo. Dando como sugerencia la implementación de un algoritmo, o algoritmos que pudiesen contemplar la generación de turnos para grupos de cuatro o cinco empleados.



A continuación, se enseña el diagrama de flujo del algoritmo necesario para la creación de horarios para un periodo de una semana, el cual fue implementado para la construcción del software que resuelve el problema planteado. Es importante tener en cuenta que el algoritmo implementado a continuación puede generar horarios de trabajo rotativo para grupos de la cantidad de personas recomendadas por el hospital san Carlos.

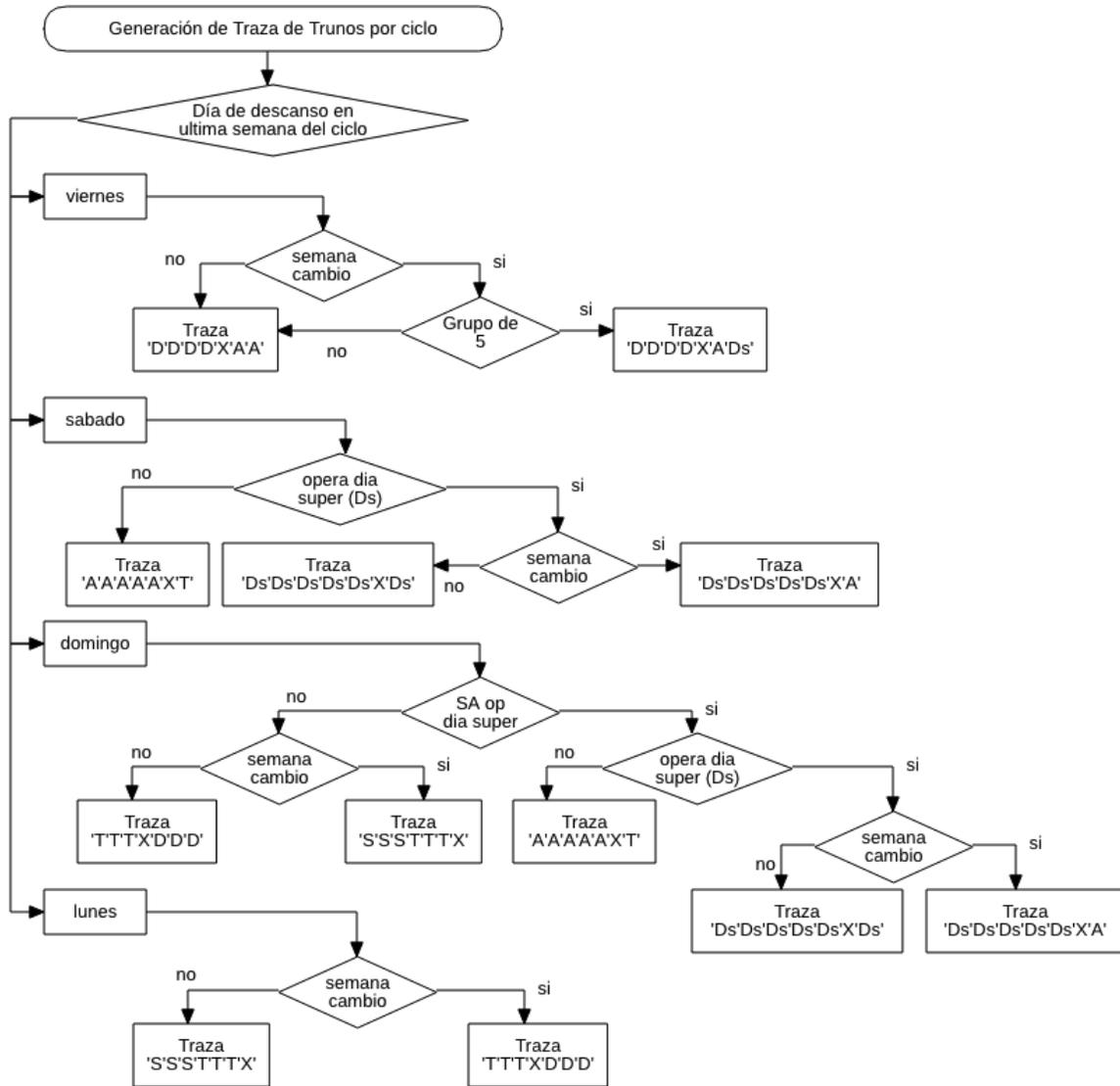


Figura 19. Algoritmo de turnos rotativos para el hospital

Las “Trazas” que se mencionan en la Figura 19 corresponden a la asignación del tipo de turno que se llevará a cabo durante una semana, comenzando por el día martes y finalizando el lunes de la siguiente semana. En este punto se debe hacer mención al concepto de ciclo, el cual hace referencia a una longitud de cuatro semanas y corresponde al periodo de tiempo en el cual un empleado tiene la asignación exclusiva de



supernumerario. Esto da a entender claramente que, al finalizar el ciclo, la asignación de supernumerario pasa a un empleado diferente tras finalizar la cuarta semana, esto, hasta que nuevamente se renueva por completo la asignación de turnos cuando todos los empleados hayan desempeñado el cargo de supernumerario. De esta forma, el proceso denominado “Generación de trazas de turno por ciclo” se puede ejecutar múltiples veces con el fin de lograr el rango de tiempo deseado, donde el algoritmo decidirá la traza a implementar semana a semana.

Las 12 posibles “Trazas” que se ejemplifican en la Figura 19 cumplen una función fundamental en la generación de horarios para cada empleado y se constituyen como un arreglo de 7 asignaciones de turnos que van desde el día lunes hasta el día domingo, por tal motivo en función de los días de descanso que se asignan a cada empleado las diferentes trazas serán seleccionadas.

Es importante mencionar que las trazas definidas en la Figura 19 se han implementado con el fin de equilibrar las horas de trabajo de manera prolongada para cada empleado,

Los turnos definidos en las trazas corresponden a:

- D : Turno de trabajo durante la jornada diurna.
- T : Turno de trabajo durante la jornada de Tarde.
- A : Turno de Trabajo durante la jornada de Amanecida.
- X : Día de descanso
- S : Supernumerario
- Ds : Supernumerario adicional (Suele asignarse a la jornada Diurna)

Es importante mencionar, que se ha seleccionado un periodo de un año como rango de tiempo máximo para la generación de horarios de un grupo, como consideración para renovar grupos o posibilitar cambios adicionales al menos una vez por año.



## Desarrollo del Software

### ○ Selección del Lenguaje

Para el desarrollo del software fue necesario seleccionar un lenguaje de programación en el cual fue posible el modelado de estructuras compuestas, es decir, variables que tuviesen múltiples atributos y permitiesen describir fácilmente aspectos del problema, para ello se establece como necesidad un lenguaje orientado a objetos que permitiera lograr dicha solución. Otra de las grandes ventajas de la programación orientada a objetos (POO) es la facilidad entregada al desarrollador a través de la reutilización de código, segmentación del mismo y disminución del tiempo de desarrollo frente a otros modelos de programación no orientados a objetos.

Otro de los aspectos a tener en cuenta durante la selección del lenguaje es la versatilidad y la facilidad para desarrollar las diferentes unidades del software, por esta razón se escogieron de manera preliminar los lenguajes Java y Python, los cuales son orientados a objetos y cumplen cada una de las condiciones y necesidades previamente planteadas. Sin embargo, con el objetivo de trabajar únicamente con un lenguaje, se selecciona el lenguaje Java debido a su gran soporte en la web, su compatibilidad con múltiples sistemas operativos, la facilidad de implementación y construcción de componentes visuales adecuados para el usuario final.

### ○ Base de Datos

Con el objetivo de desarrollar persistencia, se decide trabajar con un motor de base de datos que se ajuste al problema, de tal forma que el almacenamiento represente fácilmente la estructura que el modelo contempla. Para ello se contemplan tres etapas, el diseño de la estructura de almacenamiento, que naturalmente se encuentra relacionada con el modelo de la arquitectura MVC; la siguiente etapa es la selección de motor de base de datos idóneo y la última etapa es el análisis del ciclo de la información.

### ▪ Diseño

El objetivo en esta etapa es modelar las entidades que la base de datos manejará al interior, para ello se hace uso del concepto extraído de las mismas entidades del diseño del software y el modelo general de la arquitectura de solución, ver Figura 7 y 9. Partiendo de esto, la base de datos almacenará la información bajo el mismo esquema planteado, lo que permite la integración fácil y eficiente entre los datos que persisten en la base de datos y el modelo que manipula el software o el controlador desde la perspectiva de la arquitectura de la solución.



```
_id: 111012345
nombre: "Edwin Fabian"
apellido: "Mora"
sexo: "Masculino"
cargo: "Enferemer@"
telefono: 3112020456
direccion: "Alvania 2"
profesion: "Enferemer@"
periodosGrupos: Array
  0: Object
    nombreGrupo: "Grupo"
    fechaInicio: "2019-04-22 12:00:00"
    fechaFinal: "2019-07-22 12:00:00"

_id: 1232445
nombre: "Cristhian"
apellido: "Rojas"
sexo: "Masculino"
cargo: "Auxiliar"
telefono: 31145678
direccion: "Saldaña"
profesion: "Enferemer@"
periodosGrupos: Array
  0: Object
    nombreGrupo: "Grupo"
    fechaInicio: "2019-04-22 12:00:00"
    fechaFinal: "2019-07-22 12:00:00"
```

Figura 20. Esquema de almacenamiento de los empleados y los PeriodosGrupos

Por tal motivo, el modelo de las bases de datos debe tener la capacidad de almacenar la entidad de los empleados y los grupos, así como los demás elementos que le ayudan a controlar las fechas de operación y evitar asignaciones dobles, por esta razón la base de datos deberá tener la capacidad de extender su formato original y almacenar más elementos a medida que los empleados y grupos tienen más asignaciones de turnos y combinaciones entre dichas entidades, uno de los ejemplos más importantes es la asignación de **PeriodosGrupos** a los empleados, puesto que un empleado puede tener desde cero asignaciones hasta un número indefinido de estas.

```
_id: "Grupo"
fechaInicio: "2019-04-22 12:00:00"
fechaFinal: "2019-07-22 12:00:00"
nombreModelo: "Modelo de 4 empleados a 8 Horas"
diaInicioAlgoritmo: 1
listaEmpleados: Array
  0: Object
    _id: 111012345
    nombre: "Edwin Fabian"
    apellido: "Mora"
    telefono: 0
    periodosGrupos: Array
  1: Object
    _id: 1232445
    nombre: "Cristhian"
    apellido: "Rojas"
    telefono: 0
    periodosGrupos: Array
  2: Object
  3: Object
diasGrupo: Array
  0: Object
    asignaciones: Array
    numDiaAlgoritmo: 1
  1: Object
    asignaciones: Array
    numDiaAlgoritmo: 2
  2: Object
  3: Object
  4: Object
```

Figura 21. Esquema de Almacenamiento de los grupos.

## ▪ Selección

Debido a la necesidad de expandir los elementos almacenados, como lo es para el caso de los empleados, se selecciona una base de datos orientada a documentos, las cuales pueden almacenar múltiples instancias representadas en formato JSON y estas pueden mutar a lo largo del tiempo, es decir, que pueden almacenar atributos adicionales o pueden dejar de contener atributos que previamente tenían almacenados. Por esta razón se



implementa MongoDB como una solución para el almacenamiento y persistencia de datos, donde se almacenan las entidades en formato JSON (Documentos).

- **Ciclo de la Información**

Si bien se ha planteado el diseño de las entidades que se alojarán en la base de datos, así como la selección comercial adecuada, se precisa de especificar el ciclo de la información y las etapas en la que esta se encuentra contenida por el software en términos de memoria (RAM) y en disco.

El ciclo de la información desarrollado por el software comienza durante la inicialización del mismo, donde cada uno de los controladores conecta con la base de datos y se desarrollan las peticiones para visualizar el conjunto de colecciones, de esta forma se lleva toda la información a memoria del computador y se visualiza para que el usuario pueda realizar los cambios pertinentes. Tras cada acción desarrollada por el usuario que contemple la modificación de algún dato se desarrolla una escritura o modificación sobre las instancias de la base de datos, lo que permite almacenar de manera instantánea los cambios.

- **Interfaz Gráfica**

A través de windowsBuilder (plug-in de Eclipse) se desarrollaron los componentes generales de visualización, los cuales permiten interactuar directamente con el controlador del software. Para ello fue necesario la implementación de múltiples componentes que ofrece WindowsBuilder, desde marcos de aplicación, paneles contenedores, etiquetas de información y elementos accionadores, como lo son los botones y etiquetas seleccionables que permiten a su vez ejecutar acciones sobre el controlador a través de eventos, los cuales son acciones específicas implementadas por Java para manipular diferentes tipos de acciones.

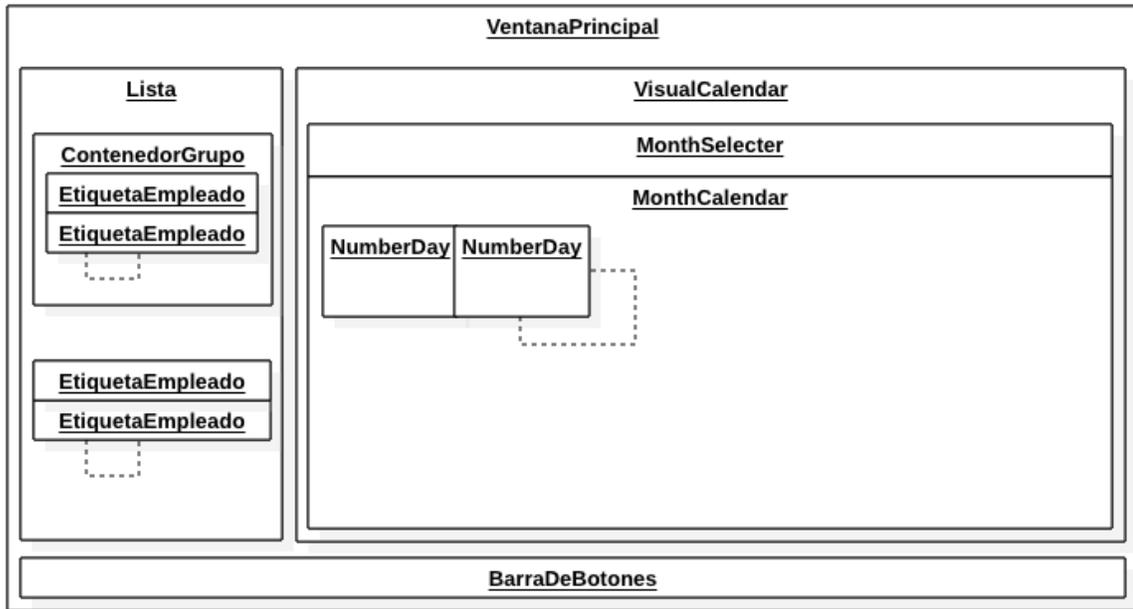


Figura 22. Componentes de la interfaz Gráfica

Es importante destacar que existe un único elemento diseñado a partir de una etiqueta que no contempla la estructura otorgada por WindowsBuilder, esta etiqueta es "NumberDay", la cual permite visualizar barras de colores asociadas a cada empleado, de tal forma que se puede visualizar los horarios como se observa en la siguiente figura.

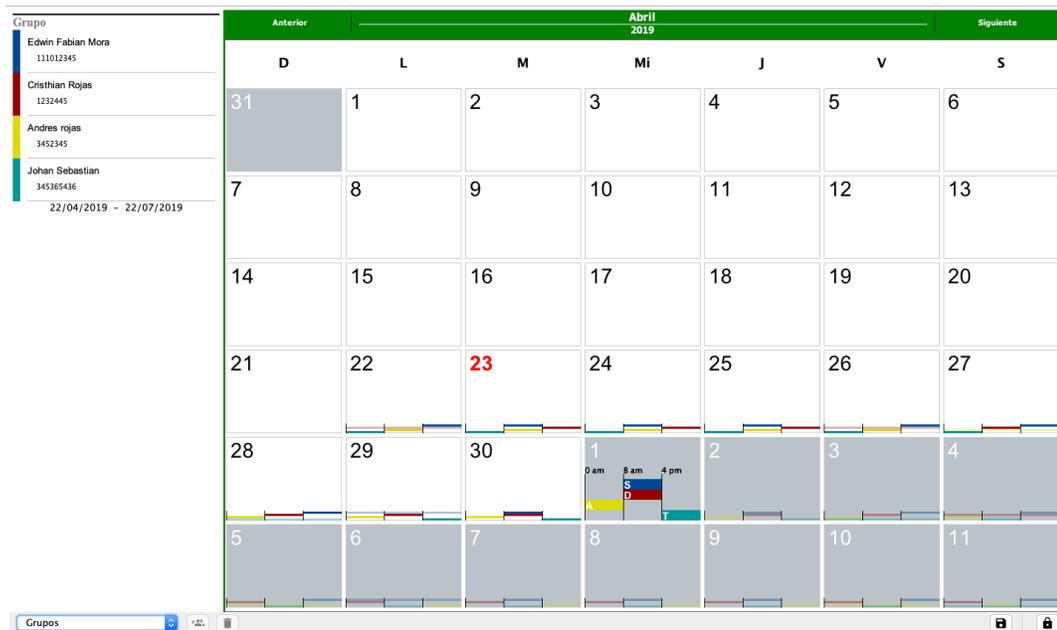


Figura 23. Interfaz gráfica principal del software (Visualización de Horarios)

A continuación, se puede visualizar una etapa importante durante la creación de los empleados y el módulo de visualización empleado.



**+ Creación de un Nuevo Grupo**

Nombre del Grupo:  
Grupo 2

Fecha de Inicio  
18/01/2020

Fecha de Final  
10/05/2020

Dia de inicio en la secuencia (Algoritmo):  
6: T D X A

Ver Esquema

Configuración de Trabajo (Modelo):  
Modelo de 4 empleados a 8 Horas

Info OK

Edwin Fabian Mora  
111012345

Cristhian Rojas  
1232445

Andres rojas  
3452345

Johan Sebastian  
345365436

OK Cancelar

Figura 24. Módulo de creación de grupos

Otro de los componentes más importantes del software es la visualización de horarios individuales, donde cada empleado es seleccionable y es posible visualizar los horarios que desarrolla, de esta forma se pueden observar las violaciones a los diferentes artículos del Código Sustantivo del trabajo.

Anterior		Abril 2019					Siguiente
D	L	M	Mi	J	V	S	
31	1	2	3	4	5	6	
7	8	9	10	11	12	13	
14	15	16	17	18	19	20	
21	22 Descanso Hora Entrada: 0 Hora Salida: 24 Total Dia: 24	23 Tarde Hora Entrada: 16 Hora Salida: 24 Total Dia: 8	24 Tarde Hora Entrada: 16 Hora Salida: 24 Total Dia: 8	25 Tarde Hora Entrada: 16 Hora Salida: 24 Total Dia: 8	26 Descanso Hora Entrada: 0 Hora Salida: 24 Total Dia: 24	27 Dia Hora Entrada: 8 Hora Salida: 16 Total Dia: 8	
28 Dia Hora Entrada: 8 Hora Salida: 16 Total Dia: 8	29 Dia Hora Entrada: 8 Hora Salida: 16 Total Dia: 8	30 Dia Hora Entrada: 8 Hora Salida: 16 Total Dia: 8	1 Dia Hora Entrada: 8 Hora Salida: 16 Total Dia: 8	2 Dia Hora Entrada: 8 Hora Salida: 16 Total Dia: 8	3 Dia Hora Entrada: 8 Hora Salida: 16 Total Dia: 8	4 Descanso Hora Entrada: 0 Hora Salida: 24 Total Dia: 24	
5 Amanecida Hora Entrada: 0 Hora Salida: 8 Total Dia: 8	6 Amanecida Hora Entrada: 0 Hora Salida: 8 Total Dia: 8	7 Amanecida Hora Entrada: 0 Hora Salida: 8 Total Dia: 8	8 Amanecida Hora Entrada: 0 Hora Salida: 8 Total Dia: 8	9 Amanecida Hora Entrada: 0 Hora Salida: 8 Total Dia: 8	10 Amanecida Hora Entrada: 0 Hora Salida: 8 Total Dia: 8	11 Amanecida Hora Entrada: 0 Hora Salida: 8 Total Dia: 8	

Figura 25. Horarios individuales



## Módulo de Exportación

Debido a que el módulo de exportación de los turnos de trabajo se estableció como una necesidad para entregar compatibilidad con herramientas ofimáticas de uso común, se desarrolló una unidad completa que permite transformar las entidades almacenadas en elementos entendibles desde una tabla de Excel, por esta razón en la sección inferior derecha el software cuenta con un botón de guardado, el cual permite exportar el grupo que se encuentre en visualización a un formato de horarios contenido en una tabla de Excel como se observa en la siguiente Figura.

Cédula	Nombre	Apellido	22-04-2019	23-04-2019	24-04-2019	25-04-2019	26-04-2019	27-04-2019	28-04-2019
111012345	Edwin Fabian	Mora	T - 16 : 24	S - 8 : 16	S - 8 : 16	S - 8 : 16	T - 16 : 24	T - 16 : 24	T - 16 : 24
1232445	Cristhian	Rojas	X - 0 : 24	T - 16 : 24	T - 16 : 24	T - 16 : 24	X - 0 : 24	D - 8 : 16	D - 8 : 16
3452345	Andres	rojas	D - 8 : 16	X - 0 : 24	A - 0 : 8				
345365436	Johan	Sebastian	A - 0 : 8	A - 0 : 8	A - 0 : 8	A - 0 : 8	A - 0 : 8	A - 0 : 8	X - 0 : 24

Figura 26. Esquema de horarios en Excel

Es importante destacar que por defecto el software desarrolla la escritura de dicho archivo en un directorio específico del ordenador en el que se encuentre instalado, por defecto es el escritorio, es decir que la ruta general para Windows es: "C:/usuario/escritorio/".



## Conclusiones

Los parámetros legales identificados e implementados en el desarrollo del software permiten evaluar el cumplimiento e incumplimiento de cada uno de los artículos contemplados en el código sustantivo del trabajo, resultando en una herramienta de validación de carga horaria de posible uso en todo territorio nacional, sin restricción alguna al hospital San Carlos o al sector salud. Claramente puede ser empleada como validación de cara horaria en las diferentes empresas con operaciones que demandan de turnos de trabajo rotativo, bajo el algoritmo implementado u otros mediante la actualización del mismo.

El cumplimiento de las necesidades planteadas por el hospital San Carlos, relacionadas con la cantidad de empleados necesarios por grupos y con las funcionalidades del software para el control y modificación de turnos de trabajo, permiten que la herramienta haya sido empleada para el seguimiento completo de carga laboral para cada empleado del hospital durante el periodo de prueba e implementación.

El modelo de turnos rotativos implementado en el hospital San Carlos Saldaña, bajo un esquema cíclico, y el patrón de diseño (Modelo UML) implementado cuenta con la capacidad para soportar múltiples combinaciones de personal y áreas de trabajo, así como la posibilidad de persistencia de la información, por esta razón se ha valorado como una herramienta útil en la gestión de las actividades y del hospital a lo largo del periodo de prueba e implementación, la cual permite evitar colapsos del ciclo de trabajo de los diferentes grupos y evita posibles conflictos que pueda acarrear demandas laborales relacionadas a la carga horaria por malas prácticas durante la gestión y asignación de turnos.

La selección e implementación del lenguaje JAVA para la construcción del software permitió el cumplimiento total de las necesidades planteadas por el hospital San Carlos y los requerimientos en términos legales; donde claramente la posibilidad de diseñar el software bajo programación orientada a objetos permitió seccionar, distribuir y generar múltiples módulos individuales que satisfacen cada una de las necesidades y requerimientos a nivel técnico. Adicionalmente el patrón de diseño seleccionado junto al lenguaje permitió integrar completamente el software con unidades adicionales como el motor de base de datos NoSQL y el módulo de exportación.

El módulo de integración del software representa un punto clave al momento de realizar acciones adicionales, como pueden ser la realización de pagos de nómina y publicación de horarios al cuerpo de empleados. Por tal motivo la integración con herramientas como Excel de Microsoft dan como resultado un software altamente adaptable, con la entera posibilidad de ser una solución para otras empresas que requieran de un modelo de control durante el proceso de asignación de turnos. Queda claro que en trabajos futuros el software puede contemplar reglas adicionales, según el territorio donde se implemente y



los modelos de contratación, sin embargo, la solución implementada en el hospital San Carlo Saldaña cumple los requerimientos funcionales establecidos y se desarrollan aportes adicionales que han facilitado el aprendizaje durante la puesta en marcha.



## Bibliografía

- Camps Paré, R., & Universitat Oberta de Catalunya. (2005). *Bases de datos*. Barcelona: UOC.
- Chiavenato, I. (2009). *Gestión del talento humano*. México: McGraw-Hill.
- Estructura Organizacional - Acerca de la entidad - Hospital San Carlos, Saldaña Tolima. Recuperado 15 de julio de 2018, de <http://esehsc-saldana-tolima.gov.co/es/acerca-de-la-entidad/estructura-organizacional>
- Freeman, A. (2013). *Pro ASP.NET MVC 5* (Fifth edition). New York, New York: Apress.
- Labidi, M., Mrad, M., Gharbi, A., & Louly, M. A. (2014). Scheduling IT Staff at a Bank: A Mathematical Programming Approach. *The Scientific World Journal*. <https://doi.org/10.1155/2014/768374>
- Madison, M. (2015). NoSQL Database Technologies, *24*(1), 15.
- Martinez, O. R. L. (2011, octubre 28). Tipos de estructura organizacional. Recuperado 15 de julio de 2018, de <https://www.gestiopolis.com/tipos-de-estructura-organizacional/>
- Marugán Cancio, M. (2015). PHELT: Planificador de Horarios en Entornos Laborales a Turnos. Recuperado de <https://ruidera.uclm.es/xmlui/handle/10578/6656>
- Ministerio de Protección Social. CÓDIGO SUSTANTIVO DEL TRABAJO, Artículo 161 (2011).
- Ministerio de Protección Social. CÓDIGO SUSTANTIVO DEL TRABAJO, Artículo 165 (2011).
- Ministerio de Protección Social. CÓDIGO SUSTANTIVO DEL TRABAJO, Artículo 166 (2011).
- Ministerio de Protección Social. CÓDIGO SUSTANTIVO DEL TRABAJO, Artículo 163 (2011).
- Nogareda, C., & Nogareda, S. (1995). NTP 455: Trabajo a turnos y nocturno: aspectos organizativos, 10.
- Nysret, M., Johannes, G., & Wolfgang, S. Efficient Generation of Rotating Workforce Schedules, 19.
- Ortiz, C., & Berg, E. Un Algoritmo Evolutivo Para la Planificación de Sistemas de Turnos Rotativos en Empresas de Servicios. Recuperado de <http://www.inf.utfsm.cl/~mcriff/OPTIMA-2003/ac81.doc>



Ramirez, A. O. (2010). Python como primer lenguaje de programación, 16.

Rotating 8 Hour Shift Schedule | 24/7 Shift Coverage | Learn Employee Scheduling. Recuperado 18 de junio de 2018, de <http://www.bmscentral.com/learn-employee-scheduling/rotating-8-shift-schedule/>

Rumbaugh, J., Jacobson, I., & Booch, G. (2000). *El Lenguaje Unificado de Modelado. Manual De Referencia*. Rational Software Corporation: Addison Wesley.

Software de gestión del capital humano y soluciones RR. HH. Recuperado 6 de abril de 2018, de <https://www.sap.com/latinamerica/products/human-resources-hcm.html>

sznajdleder, P. (2016). *Java a Fondo* (3.<sup>a</sup> ed.). alfaomega. Recuperado de <http://www.alfaomega.com.co>

Valdivieso Rueda, D. F. (2012). LA REGLA DEL TRABAJO POR TURNOS Y LA NECESARIA CONTEXTUALIZACIÓN PARA SU ENTENDIMIENTO -. Recuperado 9 de abril de 2018, de <http://legal.legis.com.co/>

VÁSQUEZ, D. C., GONZÁLEZ, N. V., & AHUMADA, A. M. (2010). PROYECTO PROFESIONAL Sistema de Horarios - SiHo, 75.