



UNIVERSIDAD  
TECNOLÓGICA  
METROPOLITANA  
*del Estado de Chile*

# Proyecto Gestión de Salas

## Ingeniería de Software

**Académico: Sebastian Salazar Molina**

**Ayudante: Patricio Pérez Valverde**

**Desarrolladores del proyecto: Gladys Cerro**

**Neftalí Madariaga**

**Juan Pablo Sepulveda**

## INTRODUCCION

Como desarrolladores a menudo estamos sistematizando procesos, a veces de forma tediosa programando de manera “pura” sin percatarnos de los problemas que esto conlleva, sin pensar en que alguno de nuestros errores en el programa no puede ser solucionado por cualquiera ya que no llevamos un patrón de diseño o un marco de trabajo.

Sabemos que estructurar y normalizar la información es clave para el desarrollo, es ahí cuando llega a nuestros oídos el termino *framework* siendo un “esquema para el desarrollo de una aplicación”. Este esquema pretende optimizar la forma en que trabajamos al momento de programar.

Utilizamos un framework para crear aplicaciones o proyectos de gran magnitud, con gran cantidad de datos, manejándolos de manera más sencilla y ordenada.

Actualmente encontramos entidades en todo ámbito que, con gran cantidad de usuarios de todo tipo manejan aplicaciones que facilitan las tareas que realiza cada uno, aplicaciones que están desarrolladas bajo un framework que permite llevar un orden al programar.

En este proyecto nos enfocamos en el ámbito de la educación, aplicando nuestros conocimientos para abordar un tema que aqueja a nuestra universidad y que no todos han dado cuenta de que es un problema importante que resolver.

Hemos visto en nuestro pasar universitario como se aglomeran los estudiantes en secretaria de estudio para ver en que sala tienen clases, en un papel que esta pegado en el panel. Tambien hemos sido testigos del “tramite” que significa pedir una sala para hacer ayudantía o alguna clase extra a las habituales, y además sabemos que al salir de una clase debemos perder 5 minutos de los 10 que tenemos de descanso para ir a revisar en que sala tenemos la siguiente clase.

Como vemos, esta situación, es una perdida de tiempo. Así es como nace la idea de “Gestión de salas” una aplicación web que manejara gran cantidad de datos y tareas a realizar. Para esto utilizaremos un framework para que su desarrollo sea mas fácil y rápido.

Pretendemos con este proyecto dar máxima ayuda y solución a nuestro problema, además de dar un toque de modernidad al ocupar la tecnología para obtener la información que necesitamos.

# **1. OBJETIVOS**

## **1.1 Objetivo general**

Desarrollar una aplicación web a la que podamos acceder de manera segura estudiantes y funcionarios para obtener y manejar información acerca de la asignación de salas de toda nuestra universidad.

## **1.2 Objetivos específicos**

- Utilizar un framework (Laravel) para el desarrollo de la aplicación.
- Aplicar una metodología de desarrollo.
- Aplicar el patrón Modelo Vista Controlador (MVC).
- Modelar el problema utilizando un paradigma orientado a objetos.

## **1.3 Ventajas**

- Los alumnos podrán revisar su horario de manera online sin necesidad de perder tiempo en llegar a la sala.
- Los profesores podrán ejecutar la misma consulta de la igual forma.
- Se convierte en un proceso fácil de ejecutar por las personas indicadas siendo una aplicación amigable al usuario.

## **1.4 Desventajas**

- Una desventaja que podría existir es que tal vez hayan momentos en los que los usuarios no tengan acceso a internet donde no podrán ver las publicaciones en la página en un momento determinado.
- En los momentos que se cae el sistema dirdoc, también dejara de funcionar nuestra aplicación por el hecho de funcionar con el mismo sistema de entrada de usuarios.

## 2. REQUERIMIENTOS

### 2.1 Requerimientos funcionales

La aplicación consta de 4 tipos de usuarios; administrador, encargado, docente y alumno. Donde cada uno de ellos tiene distintos accesos a la información, los que se detallan a continuación.

- **Administrador**: Es el encargado de manejar la totalidad del sistema, esta facultado para agregar, modificar y eliminar usuarios, así como también tendrá completo poder sobre la información de los campus, facultades y todo lo que comprende la universidad en términos de información manejada en secretaria de estudios.
- **Encargado**: Este usuario esta encargado directamente de la asignación de salas para su campus correspondiente, pudiendo modificar el nombre de las salas, su capacidad y asignar una sala a un determinado ramo según el calendario académico. También podrá agregar información académica como por ejemplo asignaturas, cursos, etc.
- **Estudiante**: Los estudiantes podrían revisar toda la información respecto al horario del día, pudiendo consultar también por alguna sala un día específico.
- **Docente**: Tendrá las mismas opciones de búsqueda que un estudiante además de hacer consultas sobre su propio horario para saber en que sala debe dar clases.

### 2.2 Requerimientos no funcionales

Estos requerimientos son los solicitados por el usuario (en este caso el profesor)

- Las operaciones realizadas por cada perfil descrito deben estar regidas por un control de acceso que evite acciones indeseadas para cada perfil (ej: Un estudiante no puede modificar la capacidad de una sala).
- La operación de carga de datos es en función al volumen de datos, por lo que el sistema es considerado inusable si es necesario ingresar 1000 asignaturas una a una.
- El uso de credenciales debe ser el mas sencillo para los usuarios finales, como todos los usuarios pertenecen de alguna forma a la universidad deberían usar las mismas credenciales que el sistema DIRDOC.
- El sistema puede ser usado por un largo periodo de tiempo, sin recibir atención del equipo de desarrollo a cargo, por lo que debe considerar los cambios básicos del calendario académico universitario.

## **2.3 Restricciones**

- Aplicación usada solo con conexión a internet.
- Sera una pagina web a la que tendrán acceso solo usuarios definidos, no hay invitados.
- Se usara lenguaje PHP, base de datos PostgreSQL.
- Equipos con acceso a internet para abrir la pagina en cualquier navegador.
- Sistema con un diseño sencillo que sea amigable y fácil de manejar por el usuario.

## **2.4 Interfaces externas**

Se requiere una interfaz de usuario simple, que pueda ser manejada desde una ventana principal con la información general de cada ítem, donde además habrá botones en la parte lateral y superior para acceder a diferentes secciones de la pagina.

Para el uso de la aplicación es necesario contar con conexión a internet, un explorador como Mozilla Firefox o Chrome, Sistema operativo OS Linux o Windows XP, vista, 7 o superior.

## **2.5 Requisitos de rendimiento**

Se espera que pueda haber una gran cantidad de usuarios simultáneamente conectados. También proporcionar la mayor cantidad de información a los usuarios en poco tiempo.

### **3. DESARROLLO DEL PROYECTO BASADO EN PMBOK**

Siendo PMBOK el estándar más reconocido para manejar y administrar proyectos nos basamos en él para desarrollar con éxito el nuestro, se describe a continuación como trabajamos cada una de las partes del ciclo de vida del proyecto.

#### **3.1 Iniciación**

Definimos la manera de trabajar y la metodología que usaremos para llevar a cabo el trabajo. Al ser un proyecto trabajado en un entorno complejo por el hecho de tener que aprender a trabajar con herramientas nuevas y donde necesitamos obtener resultados pronto, como grupo decidimos trabajar bajo la metodología ágil SCRUM, definiendo las fortalezas de cada uno de los integrantes del equipo, dejando claro que trabajábamos en equipo por un objetivo. Así entendimos que teníamos un fuerte en base de datos, otro fuerte en programación HTML y un integrante un poco más entendido en Laravel, por lo que el complemento fue fundamental. Por supuesto nunca individual izándonos, intentamos ocuparnos un poco de todo cada uno para conocer y entender lo que contempla la totalidad del proyecto.

#### **3.2 Planificación**

Nos preocupamos de detallar cada uno de los objetivos y requisitos que debía tener la aplicación para que de resultados óptimos (detallado en el punto 1 y 2 de este informe).

Seguimos una estructura de trabajo simple donde declaramos los tiempos que dedicaríamos a trabajar y las tareas que haría cada uno en un tiempo determinado para ir así revisando los avances y proponer soluciones en el caso de haber algún problema.

##### **3.2.1 El tiempo**

Intentando compatibilizar nuestra carga académica con el trabajo de cada uno y el proyecto, nos propusimos trabajar de lunes a viernes (3 horas diarias) tanto en la Universidad como en nuestros hogares reuniéndonos los lunes y miércoles para mostrar avances y solucionar problemas.

Cronológicamente comenzamos a trabajar el día lunes 4 de mayo reuniéndonos cada dos semanas pudiendo trabajar con tiempo en cada una de las tareas que nos asignábamos además de estudiar para aprender a utilizar el framework, ya más avanzado el proyecto las reuniones fueron más intensas y seguidas, con el fin de revisar detalles y problemas más complejos que necesitaban de mas concentración y trabajo en equipo.

### **3.2.1 Costos**

Tomando en cuenta que trabajamos 3 horas diarias de lunes a viernes, y somos 3 participantes, todos desarrolladores, hacen un total de 585 HH.

En detalle:

68 días en total, 3 horas diarias, 3 desarrolladores

Trabajamos 612 HH

valor UF \$25106,77

Da un total de \$15.365.343 por el proyecto.

### **3.3 Ejecución**

Lo primero que tuvimos que hacer antes de empezar a desarrollar, fue preparar nuestro entorno de trabajo.

Instalamos los paquetes necesarios de Laravel, php y del motor de base de datos, y para comenzar de lleno a desarrollar ocupamos un editor de texto intentando que para todos los integrantes del grupo fuera similar.

Desde un comienzo intentamos generar archivos ejecutables, para ir revisando los cambios que había que hacer en el proyecto a medida que íbamos avanzando en las tareas individuales.

Con cada ejecutable examinado y probado, estábamos en condiciones de seguir trabajando en el siguiente avance.

### **3.4 Monitoreo y control**

Para llevar un monitoreo de nuestro trabajo y pensando también que no siempre estábamos en el mismo lugar físico como grupo, utilizamos GitHub como plataforma de desarrollo colaborativo, lo que más de alguna vez nos dio dolores de cabeza al no saber ocuparlo, teniendo que borrar repositorios, subir otros y una serie de problemas que solucionamos con el tiempo. Pero una vez que entendimos el funcionamiento de Git se nos hizo todo mas fácil a la hora de trabajar.

### **3.5 Cierre**

Para dar por terminado este proceso revisamos cada uno de los aspectos requeridos por el Usuario.

Se revisó el cumplimiento de los objetivos y resultados esperados. Logrando una satisfactoria respuesta.

Dimos un vistazo hacia atrás para ver nuestro trabajo con detalle, nuestro avance como grupo y las lecciones aprendidas, analizando nuestro desempeño y extrayendo información valiosa para el grupo pudiendo aprender y mejorar nuestro desempeño en proyectos futuros.

Terminado el trabajo y luego de haberlo analizado tomamos acciones como:

- Incorporar a próximos proyectos actividades que no habíamos visualizado.
- Identificar riesgos previamente no considerados.
- Reconsiderar estimados en la duración o en el costo de una actividad.

### **4. SOBRE NUESTRO TRABAJO**

Nuestro grupo tuvo a lo largo de este semestre un trabajo constante y perseverante independiente del estado en que se encontró nuestra casa de estudios, siendo 100% responsables y comprometidos con el grupo.

Al ser personas con distintas fortalezas podíamos repartir el trabajo fácilmente, eso no quita mencionar que también tuvimos dificultades. Por ejemplo en el aprendizaje de Laravel, dedicamos bastante tiempo a estudiar el framework y fue un proceso complejo.

También en momentos tuvimos que dejar cosas pendientes porque el fuerte de alguno de nosotros no siempre pudo ser sustituido por el resto, en ese caso trabajábamos en otras tareas y dejábamos lo pendiente para el momento adecuado.

Nos complementamos muy bien como grupo, fuimos comprensivos y empáticos (algo fundamental en el trabajo en equipo) y eso fue positivo porque supimos entender la situación de cada uno fuera del entorno de trabajo, lo que sirvió en momentos donde alguno no podía trabajar por motivos externos, siendo los demás pro-activos y avanzando en otras cosas en ese instante.



## CONCLUSIONES

Durante el desarrollo de este trabajo logramos concluir una gran cantidad de experiencias y aprendizajes, las cuales detallamos a continuación.

Empezando por el trabajo en equipo donde utilizamos metodologías ágiles, era un concepto el cual se nos menciono muchas veces en las cátedras pero no habíamos tenido la posibilidad de implementarlo en un proyecto real sino que simplemente lo teníamos considerado en la teoría. Este proyecto nos sirvió para aplicar este concepto manejándolo de la mejor manera y favoreciéndonos el hecho de ser un grupo dedicado, responsable y pro activo.

Pensando en que como estudiantes de Ingeniería en Informática debemos aprender principalmente a trabajar en el desarrollo de proyectos web, nos enfocamos a tratar de darle la mejor coordinación entre el equipo y la mayor dedicación posible a este proyecto el cual nos ayudo a entender el funcionamiento de las herramientas y la manera de ver el desarrollo de un proyecto de software. Además aprendimos a administrar un proyecto mediante una guía de fundamentos como lo es PMBOK integrándola en todas sus etapas.

Sobre trabajar con herramientas para el desarrollo de software aprendimos que sí, se hace mas fácil trabajar con un control de versiones y con un framework (cuando aprendes a ocuparlo), pero tiene un costo inicial de aprendizaje que puede no siempre ser bajo, sino que podría resultar bastante difícil aprender, pero a largo plazo facilita el desarrollo y el mantenimiento de las aplicaciones.