



UNIVERSIDAD DE GRANADA

GRADO EN INGENIERÍA INFORMÁTICA

**Trabajo Fin de Grado — Aplicación de gestión de
rutinas semanales basada en tecnologías web**

Juan Ocaña Valenzuela

21 de junio de 2021

Índice general

1. Introducción	2
1.1. Resumen	2
1.2. <i>Abstract</i>	3
1.3. Integración de conocimientos previos	4
2. Estado del arte	5
2.1. Crítica al estado del arte	5
2.2. Propuesta	6
2.2.1. Objetivos	6
3. Metodología	7
3.1. Metodología de desarrollo	7
3.2. Análisis de requisitos	7
3.3. Planificación temporal	8
3.3.1. Planificación inicial	8
3.3.2. Planificación final	8
3.4. Presupuesto	8
4. Desarrollo del proyecto	9
4.1. Primera iteración	9
4.1.1. Análisis	9
4.1.2. Diseño	9
4.1.3. Implementación	9
4.1.4. Pruebas	9
5. Conclusiones y vías futuras	10
5.1. Conclusiones del proyecto	10
5.2. Posibles modificaciones futuras	10
Bibliografía	11

Capítulo 1:

Introducción

Palabras clave: palabras clave

§1.1: Resumen

NOTA: provisional

Existe una gran oferta en cuanto a herramientas de organización personal. Desde activas comunidades que apuestan por formatos tradicionales como agendas y calendarios personalizados hasta numerosos servicios preparados para ajustarse a cualquier tipo de persona, escoger un método de organización puede suponer una odisea para alguien interesado en la materia.

Muchas de las herramientas más populares en este campo optan por ofrecer un servicio de suscripción —a menudo con versión gratuita—, que aporta ventajas tales como la sincronización de los datos del usuario en la nube o integración con servicios adicionales. Sin embargo, a veces estas opciones son demasiado complejas para algunas personas, que simplemente buscan algo más sencillo o acorde a sus necesidades.

Se plantea una propuesta de aplicación con una funcionalidad simple, orientada a asistir en la creación de rutinas y hábitos, utilizando como soporte un sistema de objetivos y una estructura de horario semanal. La idea principal es brindar una funcionalidad simple a la vez que intuitiva, haciendo énfasis en la facilidad de uso, optando por no incluir herramientas complejas, tales como un sistema de calendario o un bloc de notas —opciones que muchos de los servicios mencionados anteriormente implementan—.

La aplicación propuesta se ha desarrollado utilizando tecnologías web extendidas y con gran soporte en la actualidad, y una arquitectura cliente-servidor basada en el patrón *REST* [4].

§1.2: *Abstract*

Una vez esté aprobado el resumen, traducir.

§1.3: Integración de conocimientos previos

Para la realización de este proyecto, han sido útiles los conocimientos adquiridos en las siguientes asignaturas del grado:

- Fundamentos de Bases de Datos
- Fundamentos de Ingeniería del Software
- Ingeniería de Servidores
- Sistemas de Información Basados en Web
- Dirección y Gestión de Proyectos
- Metodologías de Desarrollo Ágil

Capítulo 2:

Estado del arte

§2.1: Crítica al estado del arte

El mercado de aplicaciones de organización personal está repleto de opciones, con algunas de ellas muy populares. Las aplicaciones estudiadas pueden dividirse en las siguientes categorías, según su principal enfoque en cuanto a la organización personal

Aplicaciones de agenda y calendario

Las aplicaciones de agenda y calendario son las más populares, debido a su enfoque principalmente profesional. Se trata de herramientas completas y potentes, con capacidad para gestionar diferentes calendarios simultáneos, recordatorios, exportación a distintos formatos e integración con otros servicios —como por ejemplo es el caso de Google Calendar, en el que se pueden planificar citas a través de *Gmail*, o adjuntar salas de videoconferencia de *Google Meet*—.

Una de las principales ventajas de los principales servicios de este tipo es su gratuidad. Al estar mantenidos en su mayoría por grandes empresas tecnológicas —Google, Microsoft, Apple, etc.—, no requieren de un esfuerzo adicional para el usuario del resto de sus respectivas aplicaciones.

Estos servicios, pese a estar preparados para un uso intensivo y profesional, también se adaptan al usuario medio en mayor o menor medida. No obstante, su complejidad puede resultar abrumadora, y ante todo, suplen necesidades específicas: una agenda y un calendario tradicionales.

Algunas de las aplicaciones de calendario más populares o influyentes son las siguientes:

- Google Calendar
- Microsoft Outlook
- Apple Calendar (anteriormente llamado *iCal*)

Aplicaciones de tareas

Otro enfoque popular a la hora de ofrecer herramientas de organización personal es el de las aplicaciones de tareas. Estos servicios ofrecen la posibilidad de planificarse a través de listas de tareas, a menudo con añadidos como notificaciones y recordatorios, similares a los presentes en las aplicaciones más tradicionales anteriormente mencionadas.

Estas aplicaciones utilizan la **tarea** como elemento base, construyendo todo su funcionamiento en torno a ella, y poniendo el foco en facilitar la interacción del usuario con la misma. Pese a estar presentes en entornos profesionales, muchas de estas aplicaciones se orientan hacia el usuario medio en primer lugar, ofreciendo una experiencia más sencilla e intuitiva.

No obstante, muchos de estos servicios cuentan con una funcionalidad limitada, a menudo como expositor de una suscripción; y aquellos que no disponen de esta opción suelen requerir de un esfuerzo adicional por parte del usuario —por ejemplo, *Tasks.org* [tas]—.

Algunos ejemplos de aplicaciones de tareas son las siguientes:

- any.do
- Todoist

- [Tasks.org](#)

§2.2: Propuesta

2.2.1: Objetivos

Capítulo 3:

Metodología

§3.1: Metodología de desarrollo

Para la elaboración del proyecto se ha planteado utilizar aspectos sólidos de diversas metodologías ágiles, como SCRUM [5] o *Kanban* [kan]. Al estar planteadas originalmente para equipos —generalmente pequeños—, ha sido necesario adaptarlas a un desarrollo individual. Se han seguido los principios del Manifiesto Ágil [3].

Los elementos a destacar de la metodología desarrollada son los siguientes:

Tablero *Kanban*

Utilizado para representar los posibles estados de cada tarea de forma visual.

Al ser *Kanban* un marco flexible, se han definido estados o columnas acordes a las necesidades del proyecto. Dado que el desarrollo ha sido llevado a cabo por un individuo, se puede prescindir de estados intermedios de aprobación de código por el resto del equipo.

Los estados definidos son los siguientes:

- **TO DO**
- **MAKING TESTS**
- **IN PROGRESS**
- **DONE**

El estado **MAKING TESTS** determina que los tests unitarios de una tarea están en desarrollo.

Desarrollo iterativo e incremental

Presente en metodologías como *SCRUM* o *eXtreme Programming*, el desarrollo iterativo permite tener una visión temprana y real del proyecto, así como una mayor flexibilidad a la hora de detectar e implementar cambios y mejoras.

Al tratarse de un proyecto centrado en el aprendizaje de nuevas tecnologías, esta flexibilidad ha permitido una mayor creatividad y un enfoque centrado en la implementación de la aplicación, utilizando documentación auxiliar en los casos necesarios.

Historias de Usuario

Las historias de usuario

§3.2: Análisis de requisitos

Listado de las historias de usuario

§3.3: Planificación temporal

3.3.1: Planificación inicial

La planificación que se hubiese seguido en condiciones ideales

3.3.2: Planificación final

La planificación seguida finalmente dados los distintos factores externos

§3.4: Presupuesto

12 ECTS - 300 horas de trabajo - 37,5 días laborables a jornada completa Gastos de licencias: 0 Gastos de infraestructura de desarrollo: ??? Gastos de mantenimiento posterior: investigar cuotas Azure, Google Cloud y AWS para alojar la aplicación

Capítulo 4:

Desarrollo del proyecto

§4.1: Primera iteración

4.1.1: Análisis

4.1.2: Diseño

4.1.3: Implementación

4.1.4: Pruebas

Capítulo 5:

Conclusiones y vías futuras

§5.1: Conclusiones del proyecto

§5.2: Posibles modificaciones futuras

Bibliografía

[kan] Qué es kanban: definición, características y ventajas.

[tas] Tasks.org | Tasks.org.

[3] Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R. C., Mellor, S., Schwaber, K., Sutherland, J., and Thomas, D. (2001). Manifesto for agile software development.

[4] Fielding, R. T. (2000). *Architectural styles and the design of network-based software architectures*. University of California, Irvine.

[5] Schwaber, K. and Sutherland, J. (2017). The scrum guide. 2017. *No. November*, pages 6–17.