

TPE Introducción a las Metodologías de Desarrollo de Software 2022

La startup TurnoFacil nos ha contactado para desarrollar un software que permita sacar turnos para médicos. En una reunión inicial que hemos tenido con TurnoFacil nos comentan que sus intenciones son contar con una aplicación web que ejecute en una cloud pública contratada por la empresa. Esta aplicación será comercializada por TurnoFacil a distintas instituciones médicas como clínicas y consultorios por lo que la misma deberá ser fácilmente instanciable para cada una de ellas.

Los dueños de TurnoFacil quieren que el sistema sea de fácil acceso a los pacientes por lo que la pantalla inicial de la app solo debe solicitar número de DNI. En caso de que sea la primera vez que el paciente ingresa a la app se le solicitará, antes de poder sacar un turno, nombre, apellido, dirección, teléfono, email, obra social y número de afiliado.

Una vez que el paciente ingresa al sistema tendrá dos opciones: sacar un nuevo turno y ver sus próximos turnos. Al sacar un nuevo turno el paciente deberá inicialmente seleccionar un médico filtrando, si así lo deseara, por especialidad, y obra social que atiende. Una vez seleccionado un médico, el sistema deberá mostrar los días y horarios en los que atiende el médico. Además, el sistema deberá solicitar un rango de fechas para mostrar los turnos disponibles y, si así lo quisiera, el paciente podrá indicar si busca un turno por la mañana o por la tarde. En caso de no existir turnos disponibles para el rango de fechas ingresado, el sistema consultará al paciente si quiere ver los turnos para la siguiente semana luego del rango de fechas ingresado. En caso de encontrar turnos disponibles, el paciente podrá seleccionar uno. Al seleccionar un turno, se le solicitará al paciente re-confirmar sus datos personales (nombre, apellido, dirección, teléfono, email, obra social y número de afiliado). Una vez hecho esto, se registrará el turno brindando un detalle del mismo (el mismo se enviará también por email) y se dará la opción de imprimirlo. En caso de haber sacado un turno con un médico que no trabaja con la obra social del paciente (o que cobra un diferencial para su obra social) deberá alertarse al paciente de esta situación, pero permitirá sacar el turno de todas formas.

Cuando el paciente quiere ver sus próximos turnos, el sistema deberá listarlos ordenados cronológicamente indicando por cada uno médico, especialidad, día y horario. Adicionalmente, se deberá dar la opción de cancelar un turno.

TurnoFacil también nos ha pedido que exista un portal dedicado para que secretarías y médicos puedan acceder a la información de los turnos. Para ingresar a la misma cada uno contará con un nombre de usuario y contraseña. En particular, por cada “instanciación” de la aplicación, se designará un responsable de la institución médica contratante que podrá crear usuarios para secretarías y médicos de esa institución. También será responsabilidad de esa persona asignar a cada secretaria el grupo de médicos con los que trabaja.

Cada secretaria podrá cargar días y horarios de un médico con el que trabaje para que sean informados a los pacientes al momento de buscar un turno. Además, tendrán la responsabilidad de cargar turnos para cada médico con el que trabaje. Para agilizar la carga de turnos, la secretaria podrá definir para una franja horaria de un día determinado la duración de cada turno. Dado que en ocasiones los médicos son llamados por urgencias,

las secretarias deberán poder cancelar o reagendar turnos. Además, para una mejor organización, la aplicación deberá proveer una forma de que una secretaria pueda listar todos los turnos venideros de los médicos con los que trabaja. Esta funcionalidad deberá permitir filtrar por médico, turno (mañana, tarde) y rango de días. Además, sería deseable que una de estas configuraciones pueda ser guardada para que se genere de forma periódica y se imprima automáticamente (por ejemplo, un listado de todos los turnos del día para cada uno de los médicos con los que trabaja).

Finalmente, cada médico deberá poder listar todos los turnos venideros que tiene. Esta funcionalidad deberá permitir filtrar por turno (mañana, tarde) y rango de días.

Objetivo del trabajo

Los objetivos del presente trabajo son que el alumno:

- pueda aplicar de forma práctica conceptos de Scrum en el desarrollo del TP.
- utilice UML para el modelado de sistemas.
- pueda coordinar su trabajo en un equipo reducido.

Desarrollo del Proyectos

Cada equipo de desarrollo estará conformado por entre 5 y 6 personas. **La fecha máxima de inscripción será el 13 de abril mediante el siguiente formulario:** <https://forms.gle/C1tmnKqhTatM7Wjz5>. El proceso de desarrollo del TP será organizado siguiendo Scrum (but...). Se realizarán 1 warm-up y 2 sprints siendo los sprint reviews:

- 11 de mayo: entrega warm-up con todas las user stories, user story map, definición del criterio de done y prototipo (sketch).
- 1 de junio: entrega Sprint #1
- 6 de julio: entrega Sprint #2 con diagrama de clases de lo implementado hasta el momento.

En cada sprint review deberá realizarse una presentación con slides mediante un video de **no mas de 15 minutos** (se recomienda enviarlo a través de un link de Youtube) que tenga la siguiente estructura:

1. Presentación del sprint goal
2. Informar que PBIs formaron parte del sprint backlog
 - a. Mostrar como las user stories se dividieron en tareas
 - b. Mostrar, apoyándose en Jira cuales de los PBIs del sprint backlog están en DONE y cuales no.
3. Explicar que funciono bien durante el sprint, que funciono mal y como se solucionaron los problemas que aparecieron.
4. Realizar una demo de los PBIs implementados durante el sprint en el incremento.

IMPORTANTE: se consideraran solamente aquellas entregas que se hayan enviado hasta las 23.59hs de cada uno de los días indicados.

IMPORTANTE: La participación en el video del Sprint review de todos los miembros del equipo será condición necesaria (pero no suficiente) para la aprobación del TP.

El docente asignado a cada grupo oficiará de Scrum Master y Product Owner. Todo el

proyecto será gestionado a través de un proyecto Jira y un repositorio Git asociado con dicho proyecto a los cuales deberán crear. En este sentido, los commits al repositorio deberán referenciar el ID de la tarea correspondiente de Jira. Además del código fuente y la información del Jira, los artefactos a entregar en cada sprint deberán incluir, cuando corresponda, el resumen de la sprint planning y la sprint retrospective. Para la realización de los diagramas de UML se sugiere el uso de la herramienta Google Draw.io.