

## TPE 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.

### Desarrollo del Proyecto

El proyecto será evaluado teniendo en cuenta dos aspectos: El proceso de desarrollo y la documentación del dominio del sistema.

Grupo de desarrollo: cada grupo estará formado por entre 5 a 6 estudiantes. Cada grupo deberá inscribirse antes del 25 de abril a través del formulario: <https://forms.gle/T32hrxRRU9FdFa5K8>

Proceso de Desarrollo: El proceso de desarrollo del TP será organizado siguiendo Scrum (but...). Se realizarán 1 warm-up y 2 sprints.

Para realizar las etapas, los alumnos deberán realizar los siguientes eventos y artefactos de Scrum:

- Sprint Planning (junto al Scrum Master / Product Owner)
- Utilizar un Scrum Board para organizar las tareas del Sprint Backlog.
- Sprint Review (junto al Product Owner)
- Sprint Retrospective

Con este fin, el ayudante asignado oficiará de Scrum Master y Product Owner. Todo el proyecto será gestionado a través de un proyecto Jira.

| Entrega                    | Entregable  | Fecha de entrega |
|----------------------------|---|------------------|
| <b>1</b><br><b>Warm-up</b> | <ul style="list-style-type: none"><li>• Backlog (User stories) - Jira</li><li>• User story map</li><li>• Documento de visión</li><li>• Definición del criterio de DONE</li><li>• Especificación con casos de uso de una</li></ul> | 9/5              |

|                              |   |      |
|------------------------------|---|------|
|                              | funcionalidad (a definir con el ayudante)   |      |
| <b>2</b><br><b>Sprint #1</b> | <ul style="list-style-type: none"> <li>• Resumen Sprint retrospective (sprint anterior)</li> <li>• Cambios en entregables anteriores (si los hubiese)</li> <li>• Especificación “refinada” de user stories acordadas con el Product Owner - Jira</li> <li>• Sprint backlog - Jira</li> <li>• Scrum board - Jira</li> <li>• Incremento (Potentially Shippable Product)</li> <li>• Diagrama de clases de alto nivel</li> <li>• Diagrama de clases del estado actual del sistema</li> </ul>  | 13/6 |
| <b>3</b><br><b>Sprint #2</b> | <ul style="list-style-type: none"> <li>• Resumen Sprint retrospective (sprint anterior)</li> <li>• Cambios en entregables anteriores (si los hubiese)</li> <li>• Especificación “refinada” de user stories acordadas con el Product Owner - Jira</li> <li>• Sprint backlog - Jira</li> <li>• Scrum board - Jira</li> <li>• Incremento (Potentially Shippable Product)</li> <li>• Diagrama de clases del estado actual del sistema</li> <li>• Diagrama de secuencia (a definir con el ayudante)</li> <li>• Diagrama de deployment</li> </ul> | 4/7  |

Los templates de documentación deben ser descargados del Moodle de la materia. Para la realización de los diagramas de UML se sugiere el uso de la herramienta Google Draw.io.

Las diferentes entregas se realizarán **únicamente a través de un repositorio Git que cada grupo deberá crear para este fin (y al cual deberá darse permisos de lectura al ayudante asignado)**. Solo se considerarán válidas las entregas realizadas **hasta las 23:59hs de la fecha estipulada**.

Para la nota final del TPE se tendrá en cuenta cada una de las entregas, analizando no solo el desarrollo del problema sino cómo utilizaron Scrum para organizar el proceso de desarrollo. También se valorará positivamente el uso del repositorio Git para trabajar de forma colaborativa.

Todos los detalles faltantes deberán ser consensuados con el ayudante asignado.