

PAB - Roadmap

Context

Según la página oficial del ITBA hay 183 alumnos con diferentes porcentajes de beca, cursando carreras de grado. Según la reglamentación de la institución, cada uno de ellos debe estar dispuesto a colaborar o participar una cierta cantidad de horas en distintas tareas de apoyo dentro de la facultad, como pueden ser proyectos, ayudantías o trabajos administrativos. La cantidad de horas depende del tipo y porcentaje de beca recibida.

Actualmente, no existe un único sistema que englobe todas las actividades de los alumnos becados. La Secretaría Académica le entrega una lista de alumnos becados a cada uno de los Departamentos de cada carrera de la facultad, quienes luego manejan esa información de manera independiente.

Esto genera conflictos en la asignación de becarios para las distintas actividades disponibles. Por un lado, el alumno no tiene conocimiento sobre las posibles opciones vacantes, por lo que muchos no cubren su cantidad de horas o lo hacen sin motivación alguna. Por otro lado, el solicitante pierde tiempo buscando alumnos becados realmente disponibles e interesados en la actividad propuesta y sufre desmanejos con los becarios ya asignados a alguna de sus actividades.

Current system

En el caso del Departamento de Informática, existe un archivo Excel compartido a los profesores y otras personas interesadas, en el cual se listan los alumnos becados y el detalle de si están haciendo horas de becario o no. El interesado publica un anuncio en Campus para que los alumnos disponibles se postulen, o se comunica vía mail con un alumno becado en particular para consultarle acerca de su disponibilidad y su interés en la propuesta. El alumno puede aceptar o declinar la propuesta. Pero no todo resulta tan funcional como parece.

Los solicitantes

Las personas con becarios a cargo tienen dificultades debido a la descentralización de la información. Muchos becarios que ya tienen horas asignadas son contactados nuevamente con otras propuestas ya que el archivo Excel en cuestión, no se actualiza periódicamente, lo que genera confusión tanto en el alumno como en el solicitante.

Cada vez que un becario acepta una propuesta, se deben completar los siguientes formularios impresos: “Solicitud de tareas de apoyo no remuneradas de becarios”, “Compromiso de tareas de apoyo no remuneradas” y “Certificación de tareas de apoyo no remuneradas de becarios”. El primer y tercer formulario son completados por el solicitante del servicio, mientras que el segundo es completado por el estudiante.

Además, el solicitante debe entregar todos los meses un formulario por cada becario a su cargo, donde confirme que el alumno sigue realizando las tareas de apoyo para las cuales fue asignado. Esto se hace a modo de “renovación de contrato”, confirmando la continuidad del alumno en tal tarea, y es necesario para actualizar la lista de becarios libres que proporciona la Secretaría Académica. Este documento debe ser autorizado por las

autoridades correspondientes en cada entrega mensual, lo cual es caracterizado como repetitivo y tedioso por los mismos profesores protagonistas.

Los Becarios

Por último, dado que los becarios suelen tener conocimiento de unas pocas propuestas disponibles, muchas veces se ven obligados a aceptar alguna tarea que no les interesa mucho, sólo para cubrir las horas asignadas. Estas tareas podrían optimizarse si hablamos de actividades que generen motivación y despierten interés en los alumnos, dándoles a conocer oportunidades vacantes de las cuales no se han enterado por otros medios.

OKRs

Objective

Construir una plataforma para el control de las tareas no remuneradas de alumnos becados en el Departamento de Informática del ITBA, permitiendo conectar adecuadamente a alumnos becados con tareas de apoyo propuestas por profesores de este departamento.

Key Results

- Lanzar un MVP en Android en el que:
 - Un profesor pueda buscar un alumno libre adecuado a la tarea propuesta y solicitarle horas, pudiendo éste aceptar o rechazar la propuesta.
 - Un profesor pueda agregar nuevas actividades a un feed, donde un alumno pueda postularse a las tareas que le generen un mayor interés.
- Lograr que la búsqueda sea más rápida y eficiente comparado con el sistema actual

Team

Miembros del Equipo:

- | | |
|------------------------------|-----------------------------|
| → Brula, Matías Alejandro | Full Stack Developer |
| → Galderisi Isola, Martín | Product Researcher |
| → Oviedo, Estanislao Joaquín | Infrastructure |
| → Tallar, Julián | Team Lead (Product Manager) |
| → Vuoso, Julián Matías | Designer |

Strategy

La estrategia del cuatrimestre se dividirá en 2 pasos:

- 1) Programar el sistema PAB con el scope definido.
- 2) Experimentar con usuarios reales el sistema y mejorar su funcionamiento.

Nos llevará $\frac{3}{4}$ del cuatrimestre desarrollar el sistema con las features necesarias para el básico funcionamiento, entre las cuales están:

- Manejo de 3 tipos de usuarios (Alumno, Profesor y Administrador).
- Generar base de datos consistente con los datos de cada alumno becado.
- Manejo de eventos (generación, postulación, invitaciones, etc.).
- Sistema de notificaciones.

Luego de haber terminado dichas tareas, el cuarto restante del cuatrimestre se compondrá en testear la aplicación para que la misma tenga una usabilidad fácil junto a un funcionamiento sólido y libre de errores dentro del scope definido.

Habiendo dicho esto, el **objetivo fundamental** del cuatrimestre es lograr una plataforma funcional con las cuatro features funcionando correctamente.

SPRINTS

Sprint 1

- **Sprint Goals**
 - Conocer a fondo el problemático sistema de horas de becarios en uso
 - Estudiar las mayores problemáticas vigentes en la asignación de actividades para becarios, desde ambas perspectivas
 - Identificar las limitaciones legales e institucionales en el desarrollo de nuestro proyecto
 - Documentar el proyecto: Visión y Roadmap
 - Asignación de roles y tareas en general en nuestro equipo de trabajo
 - Decidir en base a las necesidades y recursos del equipo, la plataforma en la cual desarrollar el proyecto
 - Idear y desarrollar un nombre y un logotipo para la plataforma
- **Risks**
 - Encontrar poca colaboración de los integrantes del Departamento de Alumnos del ITBA, responsables del manejo actual de los becarios
 - Demorarse en la elaboración de la idea
- **Sprint Plan**
 - Realizar una investigación en el ITBA, reuniéndonos con alumnos becados y profesores con becarios participando en sus actividades.
 - Visitar el Departamento de Alumnos del ITBA para conocer cómo manejan ellos el sistema.
 - Identificar los objetivos del proyecto y plasmarlos en la documentación del mismo.
 - Idear y desarrollar un nombre y logotipo para la plataforma.
 - Decidir una plataforma en la cual desarrollar el proyecto.
- **Sprint Retrospective**
 - Tardamos mucho en definir la idea del proyecto principalmente debido a la búsqueda de soluciones en lugar de buscar problemas. Esto llevó a un retraso en todas las partes del proyecto, fundamentalmente en la investigación del mismo.
 - Tuvimos inconvenientes en la división de tareas debido a problemas de comunicación dentro del grupo.
 - Modificamos el scope del proyecto en la mitad del sprint como parte de la investigación realizada.
 - Tuvimos demoras en la decisión de la plataforma donde se desarrollará el proyecto. Decidimos realizarla en Android.

Sprint 2

- **Sprint Goals**

- Decidir la plataforma en la cual desarrollar el proyecto. Elegimos Android.
- Diseñar un prototipo completo de la aplicación de Android en Adobe XD, con todas las funcionalidades que debe ofrecer la aplicación.
- Investigar acerca del manejo de base de datos en Android utilizando Firebase e implementar una API de extracción e inserción de datos en la misma.
- Investigar acerca del proceso de autenticación de usuarios en Android a partir de una base de datos e implementar una API de autenticación para usuarios de Google.
- Investigar acerca del desarrollo de una aplicación de Android y la integración entre el front-end (XML) y el back-end (Java).
- Desarrollar una vista básica de la aplicación en Android.

- **Risks**

- Tener demoras en el diseño del prototipo que retrasen el desarrollo de la vista básica, o que modifiquen su diseño una vez arrancado.
- Encontrar dificultades en la comprensión del manejo de bases de datos y en el desarrollo de aplicaciones en Android.

- **Sprint Plan**

- Decidir en conjunto la plataforma en la cual desarrollar el proyecto. Elegimos Android.
- Realizar una división de las tareas a desarrollar.
- Martín Galderisi realizará el diseño de un prototipo de la aplicación de Android en Adobe XD.
- Julián Vuoso realizará la investigación e implementación acerca del manejo de la base de datos en Android utilizando Firebase.
- Estanislao Oviedo realizará la investigación e implementación acerca de la autenticación de usuarios en Android.
- Matías Brula y Julián Tallar realizarán la investigación y el desarrollo de una vista inicial de la aplicación en Android.

- **Sprint Retrospective**

- La presentación realizada en la demostración del sprint resultó ser muy densa por la longitud de los videos adjuntos, sumado al hecho de no incluir casi texto que acompañe a los mismos.
- Recibimos críticas justificadas acerca de la vista de alumnos becados en el prototipo. Luego de revisarlas, notamos que es muy poco clara, similar al Excel que justamente estamos intentando eliminar. Será revisada al momento de implementarla en Android.
- Tuvimos algunos inconvenientes de comunicación respecto a algunas funcionalidades implementadas. No estuvo claro si una de las tareas planificadas había sido terminada hasta el día de la presentación.

Sprint 3

- **Sprint Goals**

- Encontrar una manera de visibilizar el contraste entre el sistema actual y el propuesto para reforzar el impacto de nuestro proyecto.
- Realizar el layout general completo de la aplicación.
- Integrar el login realizado en el sprint anterior al resto de la aplicación.
- Realizar una AppBar definitiva para la aplicación
- Diagramar la base de datos a utilizar en Firebase

- **Risks**

- Encontrar poca colaboración por parte de los docentes para conseguir el archivo Excel con alumnos becados.

- **Sprint Plan**

- Conseguir el archivo Excel utilizado para el manejo de alumnos becados dentro del Departamento de Informática.
- Realizar el layout de Mis proyectos, Explorar, Becarios y Notificaciones con información hardcodeada, pero tomada dinámicamente de archivos Java.
- Realizar una AppBar útil para toda la aplicación.
- Tomar el login realizado y agregarlo como la primera pantalla de la aplicación.
- Diagramar la base de datos a utilizar en Firebase y completarla con datos para realizar las pruebas

- **Sprint Retrospective**

- Encontramos nuevas dificultades en el login de la aplicación. Una vez loggeado, no aparecía más la pantalla de login y no teníamos control sobre cuándo sí lo hacía. Por ello, no pudimos realizar la integración por completo.
- No llegamos a implementar el layout de alumnos becados debido a la falta de tiempo por superposición de fechas de entrega de las distintas materias.

Sprint 4

- **Sprint Goals**

- Integrar el manejo de la base de datos a la aplicación.
- Investigar la generación de notificaciones Push con Firebase y Android.
- Separación entre Layout de alumno y profesor según usuario loggeado.
- Creación de actividades desde la aplicación.

- **Risks**

- Gran parte de los objetivos del sprint están relacionados con el manejo de Firebase en Android, por lo que cualquier dificultad encontrada en él será una gran traba en el resto de las tareas.

- **Sprint Plan**

- Sincronizar la información de cada una de las vistas con Firebase. Las vistas de Explorar actividades y Explorar becarios deben visibilizar únicamente aquellos elementos que sean de interés a la persona loggeada. Por ejemplo, la vista de explorar actividades no debe mostrar aquellas en las que uno ya está inscripto.
- Fijar las opciones del menú de más opciones dentro de la toolbar.
- Realizar un layout funcional de creación de actividades. Esta opción solo debe estar disponible en caso de que la persona loggeada sea un profesor.
- Investigar acerca de la generación de notificaciones push en una aplicación Android de prueba.

- **Sprint Retrospective**

- Tuvimos conflictos en la separación de layouts de alumno y profesor. Lo resolvimos de manera provisoria duplicando la actividad principal y yendo a una u otra según el tipo de usuario loggeado, pero debemos investigar cómo hacerlo correctamente.
- Tuvimos conflictos con la recarga de información en la aplicación al modificarse la información. Para que aparezca el nuevo proyecto creado, debemos navegar nuevamente a esta vista. Debemos corregirlo en el siguiente sprint.

Sprint 5

- **Sprint Goals**
 - Realizar un refactor del proyecto para aplicar los conocimientos adquiridos en HCI (navegación, Jetpack, etc).
 - Implementar el manejo de notificaciones en la aplicación.
 - Realizar los layouts de las vistas faltantes en la aplicación.
 - Realizar un refactor de los layout realizados para que sean más agradables al usuario.
- **Risks**
 - Encontrar dificultades en la generación de notificaciones propias desde Android. Hasta ahora, solo ha sido testeado desde Firebase.
 - Al ser el sprint final, cualquier demora puede comprometer la entrega funcional de la aplicación.
- **Sprint Plan**
 - Convertir toda la navegación entre actividades a navegación entre fragmentos (a excepción del Login) para facilitar las transiciones y transferencias de argumentos.
 - Crear clases correspondientes para el manejo de la Api en una capa aparte para bajar el acoplamiento del proyecto. A partir de ella, crear clases ViewModel correspondientes para el manejo del repositorio en cada fragmento.
 - Implementar la generación y recepción de notificaciones Push para que alumnos y profesores puedan comunicarse adecuadamente.
 - Crear las vistas de proyecto y perfil de alumno.
 - Realizar tests en distintos dispositivos (emulados y físicos).
- **Sprint Retrospective**
 - La realización de los refactor llevó más tiempo del que pensábamos. Igualmente pudimos dividir las tareas para que esto no las retrase.
 - Tuvimos conflictos con la aplicación de observadores sobre las funciones de getteo de Firebase. Éstas agregan listeners al modificarse la Data que solamente se borran cuando se los quita explícitamente. Por ello, debimos realizar algunos listener simples.