

Proyecto Final

Tutor: Mariano Benitez

Integrantes:

- ❖ Agustin Pagnoni (52118)
- ❖ Gabriel Zanzotti (52067)

Índice

1. Abstract
 - 1.1. Enunciado oficial
 - 1.2. Qué es Bonita BPM y por qué fue elegido
 - 1.3. Arquitectura de Bonita BPM
2. Requerimientos del proyecto
 - 2.1 Requerimientos funcionales
 - 2.2 Requerimientos no funcionales
 - 2.3 Próximos pasos
3. Instalación
 - 3.1 Instalación modo desarrollador
 - 3.2 Puesta en producción
4. Implementación
 - 4.1 Flujo principal
 - 4.2 Agregar proyecto final
 - 4.3 Aceptación de proyecto de alumno autor
 - 4.4 Listado de proyectos disponibles
 - 4.5 Confirmación o cancelación de proyectos
 - 4.6 Notificar confirmación o cancelación de proyecto
 - 4.7 Confirmación o cancelación de proyecto de un alumno
 - 4.8 Confirmación y cancelación de proyectos
5. Casos de uso
 - 5.1 Alumno creador de proyectos
 - 5.1 Alumno autor de un proyecto
 - 5.3 Alumno que solo participa en un grupo
 - 5.4 Grupo cancelado y armado de un nuevo grupo
6. Conclusiones

1. Abstract

El proyecto consiste en desarrollar un sistema que le permita a la persona encargada de la materia “Proyecto Final” automatizar la mayor cantidad de tareas posibles y de esta manera simplificar las acciones iniciales del proceso.

Para lograr este objetivo se desarrolló una plataforma utilizando Bonita BPM (7.4.5), programa usado en la materia “ 72.71 - Integración de Sistemas de Información”. Con dicho programa se realizó todo el front-end y el back-end de la plataforma la cual permite a la persona encargada de la materia iniciar el proceso, manejar la presentación y selección de proyectos presentados tanto por alumnos como por profesores y el control de inscripción de los alumnos a cada proyecto.

Lo que se obtiene al finalizar un periodo de inscripción es un listado con los proyectos disponibles, los tutores a cargo de cada proyecto y los grupos de alumnos inscriptos en cada proyecto.

1.1 Enunciado oficial

Los procesos bien definidos son una pieza clave en toda gestión, y su automatización permite agilizar y optimizar las operaciones de dicha gestión.

El propósito de este proyecto es modelar los procesos “core” del Departamento de Ingeniería Informática y automatizarlos a través de una implementación con una herramienta BPM.

El relevamiento de procesos se realizará con la Dirección del departamento y la implementación deberá integrarse con los sistemas académicos ya existentes en la Universidad (SGA, Sakai, etc.).

1.2 Qué es Bonita BPM y por qué fue elegido

BPM (Business Process Management) es una disciplina en gestión de operaciones que usa diversas metodologías para lograr automatizar y optimizar procesos de negocio. Para representar los procesos se suele usar BPMN (Business Process Management Notation) que es un estándar para el modelado de procesos de negocio que provee una notación gráfica para especificar los procesos dentro de un diagrama de procesos de negocio. Esta notación gráfica tiene como principal objetivo la entendibilidad de los gráficos y su simpleza de lectura.

Bonita es una suite de software con versión open-source que sigue la disciplina BPM y está preparada para desarrollar el manejo de procesos de negocio y los flujos de trabajo asociados a un negocio. También utiliza la notación BPMN para el diagrama de procesos.

Desde el enunciado oficial se explicitaba que debía utilizarse una herramienta BPM. Dado que se había utilizado Bonita en una de las materias se decidió optar por esta herramienta de BPM. Dicha herramienta presenta grandes ventajas a la hora de hacer un desarrollo como el de este proyecto ya que tiene pre armadas varias funcionalidades que le permiten al desarrollador hacer foco en el flujo de la aplicación.

La más utilizada y destacable pasa por la simplicidad de envío de emails, por otra parte la abstracción de lo que es un sistema de usuarios también simplificó mucho el desarrollo ya que Bonita proporcionaba una aplicación funcional base adaptada a nuestro proyecto.

Otra apreciación respecto a Bonita tiene que ver con el orden que presenta el proyecto una vez finalizado y lo simple que es encontrar el punto exacto de falla gracias a esto. Una separación en pequeños

procesos y una visibilidad general de interacción entre los mismos permite entender de forma simple el proyecto.

Teniendo en cuenta que al proyecto se le pueden agregar múltiples funcionalidades extra en un futuro creemos que la elección de Bonita era necesaria ya que cualquier alumno que haya cursado la materia “Integración de Sistemas de Información” podría desarrollar dichas funcionalidades. Además la existencia de una versión gratuita hizo que podamos validar su potencial en este proyecto pudiendo optar por la versión paga para futuras implementaciones.

1.3 Arquitectura de Bonita BPM

Bonita BPM se divide en 3 partes:

1.3.1 - Bonita Studio

Bonita Studio es la pieza de software que ofrece para generar los diagramas de procesos, interconectar tareas y procesos mediante una herramienta visual que permite la edición de código y compilación. La interfaz es similar al entorno de desarrollo integrado “Eclipse”, ya que la implementación fue realizada modificando dicho programa.

En resumen es la herramienta encargada de crear, editar, compilar, encender y apagar el Bonita BPM Engine

1.3.2 - Bonita BPM Engine

El Bonita BPM Engine es una aplicación en Java que se ocupa de ejecutar los procesos definidos a través del Bonita Studio. El Engine se accede programáticamente vía un API, que ofrece una gama de servicios entre los cuales los que destacan son el acceso a la organización de grupos, usuarios y roles, como también el servicio que permite manejar los procesos que están en ejecución o para encender alguno. Es la pieza de

software que ofrece servicios para acceder al entorno donde corren los procesos como también a la base de datos. Su implementación es modular por lo que permite el uso de distintas bases de datos, que por default utiliza una H2. El Engine utiliza Hibernate como ORM para interactuar contra las posibles bases de datos que tiene en su lista de conectores.

1.3.3 - Bonita Portal

El Bonita Portal es la pieza de software que ofrece para visualizar e interactuar desde un navegador de internet hacia el Bonita Engine. Basado en Angular JS, ofrece una forma amigable de crear y modificar vistas y formularios. Ofrece también una forma visual en modo administrador, para gestionar los procesos del Engine.

2. Propuesta funcional

2.1 Requerimientos Funcionales

Al comenzar el proyecto se plantearon los siguientes puntos a realizar:

1. Configuración de fechas

Se podrán configurar las fechas de inicio y fin para publicación de proyectos finales de profesores y fechas de inicio y fin para publicación de proyectos por parte de los alumnos.

2. Presentación de proyectos

Se le enviará un mail a los profesores para que puedan presentar un proyecto final, dichos proyectos tienen un tutor (el profesor que los presenta), un cupo, un título y una descripción.

También se le enviará a los alumnos un mail para que puedan proponer un proyecto, eligiendo un título y una descripción del mismo. Una vez finalizado el periodo de presentación de proyectos el responsable podrá seleccionar los proyectos que se realizarán este cuatrimestre y podrá también editarlos para corregir detalles de los mismos. Una vez finalizada la selección el responsable podrá enviar los proyectos a los alumnos para que estos se inscriban a los mismos.

3. Selección de proyectos

Se le enviará un email a los alumnos que presentaron un proyecto que fué seleccionado, notificándoles su aprobación y dejándoles elegir los integrantes de su equipo. Una vez elegidos los mismos, se les enviará un mail a ellos notificándoles del grupo al que fueron elegidos y pidiéndoles confirmación de esta oferta. Mientras dicha oferta no sea aceptada o rechazada, no podrán ser elegidos para otro proyecto.

También se les enviará un mail al resto de los alumnos notificándoles los proyectos seleccionados para que ellos puedan anotarse. Cuando un alumno decide anotarse primero debe elegir el proyecto. Una vez elegido, éste no podrá ser elegido por otro alumno durante los próximos 15 minutos. Durante este tiempo el alumno que elige el proyecto deberá armar su grupo y al enviar la conformación del mismo todos los integrantes recibirán un email el cual deben confirmar dentro del plazo especificado anteriormente para consolidar el grupo (durante este tiempo no podrán ser elegidos para otro proyecto). Caso contrario el proyecto y los alumnos quedan libres para poder ser elegidos nuevamente.

Una vez que se arman todos los grupos estos le llegan al responsable quien podrá hacer una última edición de los mismos y enviarlos a los alumnos.

2.2 Requerimientos No Funcionales

Usabilidad

- ❖ El sistema debe proporcionar mensajes de error que sean informativos y orientados a usuario final.
- ❖ El sistema debe poseer interfaces gráficas bien formadas.

Extensibilidad

- ❖ Nuevas funcionalidades del sistema, tal como agregar nuevos flujos deberán poder ser realizadas sin mayores complicaciones. Personas con la lógica de negocio tienen que poder mirar el sistema y poder decidir dónde agregar la funcionalidad.

Flexibilidad

- ❖ Tanto los procesos, como los formularios o sus validaciones deberán poder ser fácil y rápidamente accesibles al cambio.

Compatibilidad

- ❖ El sistema tiene que ser compatible con cualquier implementación de BPMN

2.3 Próximos pasos

Además de definir las tareas a realizar en este proyecto también se plantearon posibles tareas para futuras implementaciones, las mismas tienen que ver con el seguimiento de los proyectos y la

conclusión de los mismos. Las funcionalidades que se plantearon son:

Seguimiento de proyectos

Una vez definidos los grupos es deseable mantener un estado del proyecto para ver la completitud del mismo. Para esto se pensó que se podrían agregar recordatorios por email para que los tutores reporten el estado actual del proyecto, incluyendo una serie de puntos los cuales se traducen a un porcentaje de completitud.

Conclusión de proyectos

Una vez que el porcentaje de completitud llegue al 100% el tutor pediría que los alumnos presenten el proyecto. Esto enviaría emails a los posibles jurados para que confirmen sus fechas de disponibilidad, una vez que los posibles jurados completen esta instancia se plantearían fechas disponibles para que los alumnos elijan. Una vez definido esto se le enviaría a los miembros del jurado un email para notificarles la confirmación de la presentación con la fecha y horario.

Fin de la materia

Una vez realizada la exposición el jurado recibirá una tarea para cargar el resultado final del proyecto y la persona encargada de los proyectos finales recibirá un email con el resultado final de la materia para que pueda cargar la nota.

3. Instalación

3.1 Instalación modo desarrollador

El repositorio del proyecto se encuentra en la organización ITBA de Bitbucket bajo el nombre de “bpm-departamental”. Dicho repositorio posee 2 archivos principales, el primero es “ITBA IT.xml” en el cual se puede ver una conformación de la organización que utiliza el sistema en formato xml, aquí se deberá agregar a los alumnos, profesores, tutores y al responsable de la materia Proyecto final para que dichas personas tengan acceso a la plataforma.

El segundo archivo es “ProyectosFinalesITBA-1.0.bos” este posee toda la información y es el archivo que se importa en Bonita para que el proyecto funcione. Tiene en su interior guardado todo el código funcional de la aplicación desarrollado por nosotros para ser ejecutado en cada etapa del proceso así como también el diagrama con todos los subprocessos que presenta.

Para probar el funcionamiento del proyecto deberá instalar la versión 7.5.4 de Bonita BPM desde su pagina oficial (<https://www.bonitasoft.com/downloads?version=7.5.4>).

Una vez instalado, abrir el programa ir al menú “diagrama” y en las opciones de “importar” elegir la opción de archivo BOS. Al elegir el archivo “ProyectosFinalesITBA-1.0.bos” del repositorio debería verse todo el flujo del proyecto.

Para finalizar la instalación se debe incluir el archivo “ITBA IT.xml” para que se inicialicen los usuarios de la aplicación y para asegurarse de que cada usuario tiene el rol correspondiente en la organización. Para lograr esto se debe ir al menú “organización” y hacer click en “importar”. Allí elegimos el archivo mencionado anteriormente y procedemos a publicar la organización. Para realizar la publicación se debe ir al menu “organización” y hacer click en “publicar”, esto

abrirá una vista en donde se deberá seleccionar la opción “ITBA IT” y hacer click en publicar.

En caso de querer hacer modificaciones en la organización también se puede ir al menú “organización” y seleccionar la opción “gestionar”. Esto abrirá una ventana que permite hacer cambios en la organización para luego publicarla nuevamente.

Una vez realizada esta configuración previa se deberá hacer click en el botón “run” en la parte superior del programa para iniciar la ejecución del mismo y así probar su funcionamiento.

3.2 Puesta en Producción

Para tener un servidor corriendo los servicios de Bonita BPM es necesario contar con el instalable de Bonita (en Windows o Linux) y ejecutarlo. Mantener el proceso de Bonita corriendo.

Para la configuración de los procesos de este trabajo, acceder al Portal de Bonita:

ip_del_servidor:8080/bonita

1. Import Business Data (bdm.zip)
 - a. Acceder al Portal de Bonita con el usuario *install* y password *install*
 - b. Ir a la solapa de BPM Services, apretar en el botón de “Pausa”, esto permite que se hagan cambios al entorno de Bonita.
 - c. Ir a la solapa Business Data Model, seleccionar el archivo “bdm.zip” y apretar en el botón “Activate”
2. Ir a la solapa Organization > Import/Export > seleccionar el archivo “ITBA IT.xml” del proyecto > Apretar en el botón de “Import” (notar que el XML es de muestra, el verdadero

tiene que ser actualizado haciendo un export con la información del SGA (el XML actualizado lo tiene y mantiene el tutor de este proyecto)

3. Hacer logout del usuario "install"
4. Hacer login con el usuario "admin", password default "bpm" (o ver dentro del archivo "ITBA IT.xml" su password). Activar el modo administrador de la siguiente forma:
 - a. En la esquina superior derecha activar el modo "Administrador" haciendo click al lado del Nombre del Usuario.
5. Dentro del Portal, instalar los ".bar" dentro de la carpeta del proyecto de este trabajo (8 archivos en total) y hacer lo siguiente para cada uno de los procesos:
 - a. Ir a la solapa BPM > Processes
 - b. Hacer click en "+Install" y seleccionar un proceso (".bar")
 - c. Buscar el proceso recién instalado en los procesos "Disabled"
 - d. Hacer click sobre el proceso, buscar y apretar el botón de "MORE ..."
 - e. Habilitar el proceso haciendo click en el botón estilo interruptor.
6. Activar el modo usuario repitiendo el paso 4.a
7. Ir a la solapa de "Processes", hacer click en el proceso "PresentacionProyecto" y apretar el botón de "Start"

4. Implementación

Al comienzo del proyecto se comenzó utilizando Bonita BPM 7.1, dicha versión era la última disponible y conforme el paso del tiempo las versiones fueron corrigiendo problemas que dificultaban gran parte del desarrollo, un ejemplo concreto de esto tiene que ver con la visibilidad directa de la base de datos. En un principio no se podían

ver las tuplas guardadas en la base y esto generaba varios problemas ya que nos costaba corroborar la persistencia de los datos y que se estuviera guardando lo que nosotros necesitábamos.

El diagrama comienza con el flujo de la *figura 1* y está compuesto de la siguiente manera:

4.1 Flujo principal

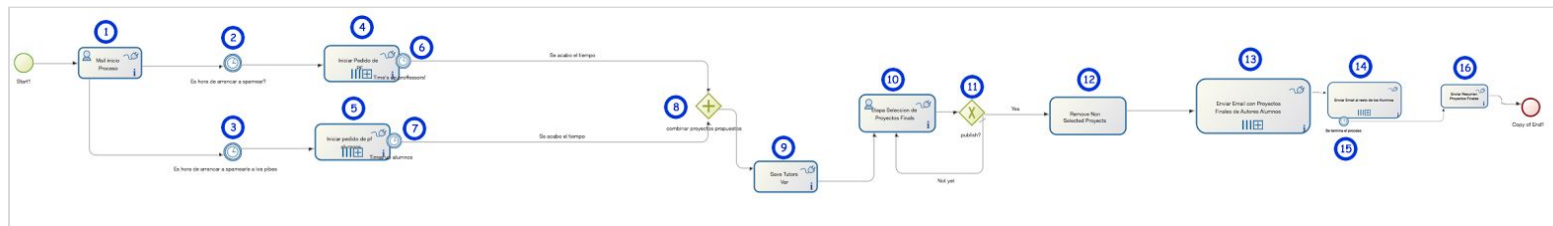


Figura 1: Diagrama del flujo principal.

1. Inicio de proceso

1.1 - email de inicio

Se envía un email al usuario con rol de BPM Admin notificándole que se aproxima un nuevo ciclo lectivo y se le proporciona un link a la plataforma para que complete el formulario de inicio del proceso. Un ejemplo de dicho email puede verse en la *figura 2*.

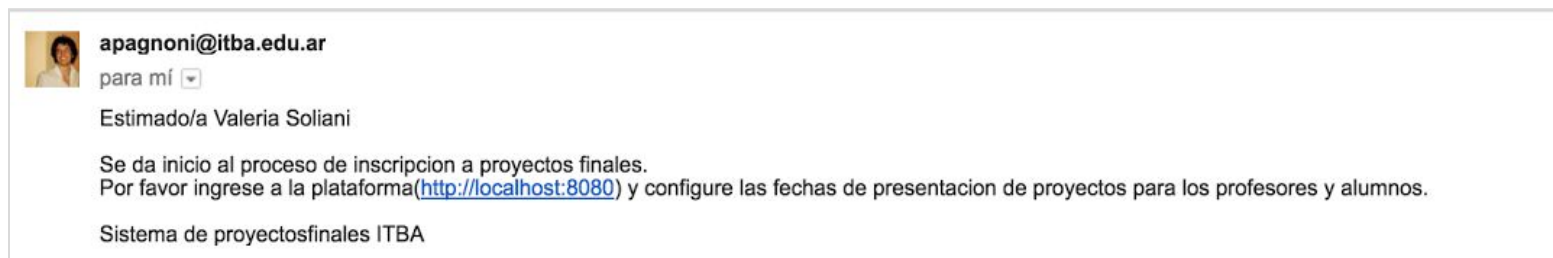
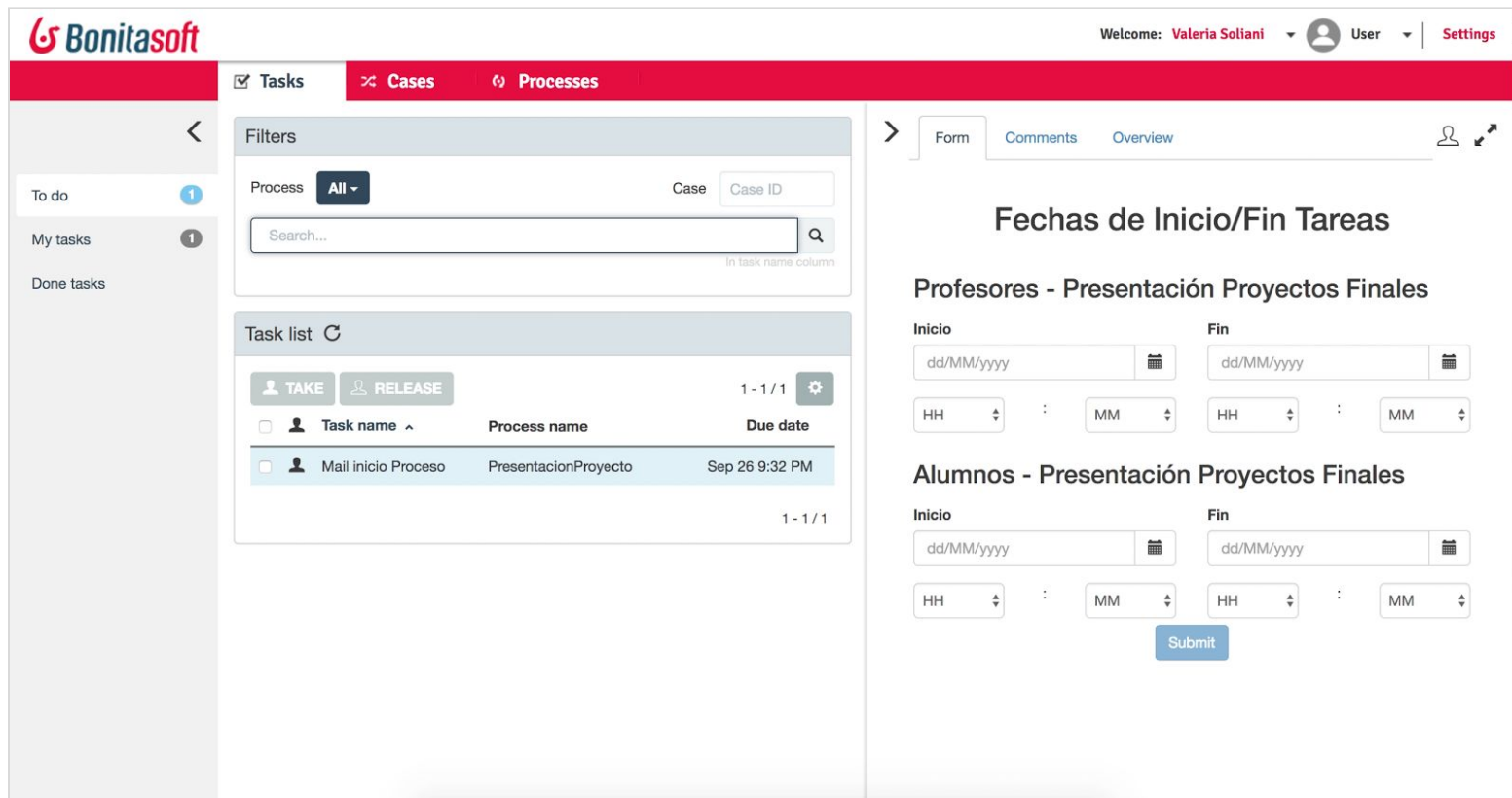


Figura 2: Email de Inicio.

1.2 - formulario de inicio de proceso

Se proporciona un form (como el que se puede ver en la *figura 3*) con 2 pares de campos de fechas y horarios de inicio y fin. Las fechas y horarios de inicio se utilizarán para comenzar el

envío de mails a profesores y alumnos informándoles que pueden ingresar a la plataforma y presentar proyectos finales mientras que las fechas y horarios de fin darán un cierre a la presentación de proyectos de profesores y alumnos.



The screenshot shows the Bonitasoft web application interface. At the top, there's a header with the Bonitasoft logo, a welcome message for 'Valeria Soliani', and user settings. Below the header, there's a navigation bar with 'Tasks', 'Cases', and 'Processes' tabs. The left sidebar contains a task list with 'To do', 'My tasks', and 'Done tasks' sections. The main content area is divided into two panels. The left panel shows a 'Task list' with a table containing one task: 'Mail inicio Proceso' under the process 'PresentacionProyecto', with a due date of 'Sep 26 9:32 PM'. The right panel is titled 'Fechas de Inicio/Fin Tareas' and contains two sections: 'Profesores - Presentación Proyectos Finales' and 'Alumnos - Presentación Proyectos Finales'. Each section has date and time pickers for 'Inicio' and 'Fin', and a 'Submit' button at the bottom.

Figura 3: formulario de inicio de proceso.

2. Condición de Inicio - emails inicio - profesores

Se utiliza un timer que se activa cuando se cumple la fecha y hora de inicio de profesores especificada en el paso anterior dando inicio al envío de emails.

3. Condición de Inicio - emails inicio - alumnos

Se utiliza un timer que se activa cuando se cumple la fecha y hora de inicio de alumnos especificada en el *paso 1* dando inicio al envío de emails.

4. Iniciador de subproceso - emails propuesta de proyecto - profesores

En esta etapa se inicia un subproceso para cada profesor (ver diagrama del subproceso en la *figura 7*). En dicho subproceso cada profesor podrá proponer proyectos finales. Además se envía un email como el que se puede ver en la *figura 4* para ofrecerle a los profesores que propongan proyectos finales.

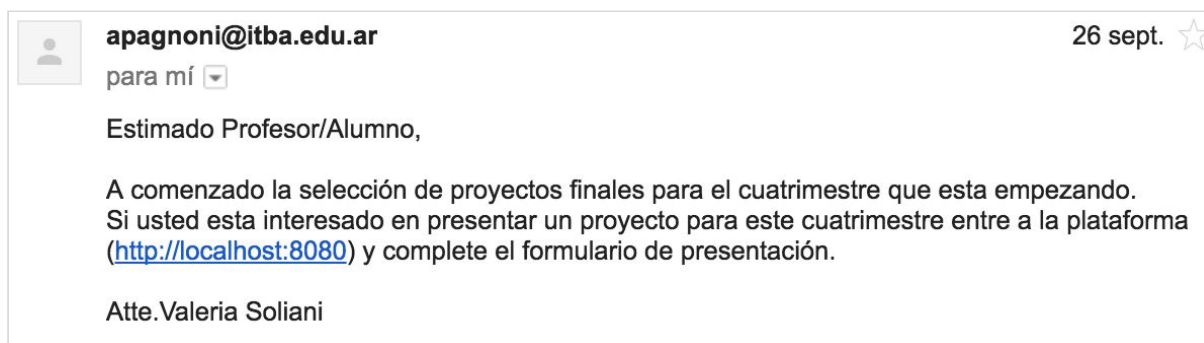


Figura 4: email para que alumnos y profesores presenten proyectos.

5. Iniciador de subproceso - emails propuesta de proyecto - alumnos

En esta etapa se inicia un subproceso para cada alumno (ver diagrama del subproceso en la *figura 7*). En dicho subproceso cada alumno podrá proponer proyectos finales. Además se envía un email como el que se puede ver en la *figura 4* para ofrecerle a los alumnos que propongan proyectos finales.

6. Condición de finalización por tiempo - profesores

Aquí se controla, utilizando la fecha de fin de profesores, cuando finaliza el tiempo de presentación de ellos para poder continuar con el proceso.

7. Condición de finalización por tiempo - alumnos

Aquí se controla, utilizando la fecha de fin de alumnos, cuando finaliza el tiempo de presentación de ellos para poder continuar con el proceso.

8. Condición de finalización de los 2 flujos

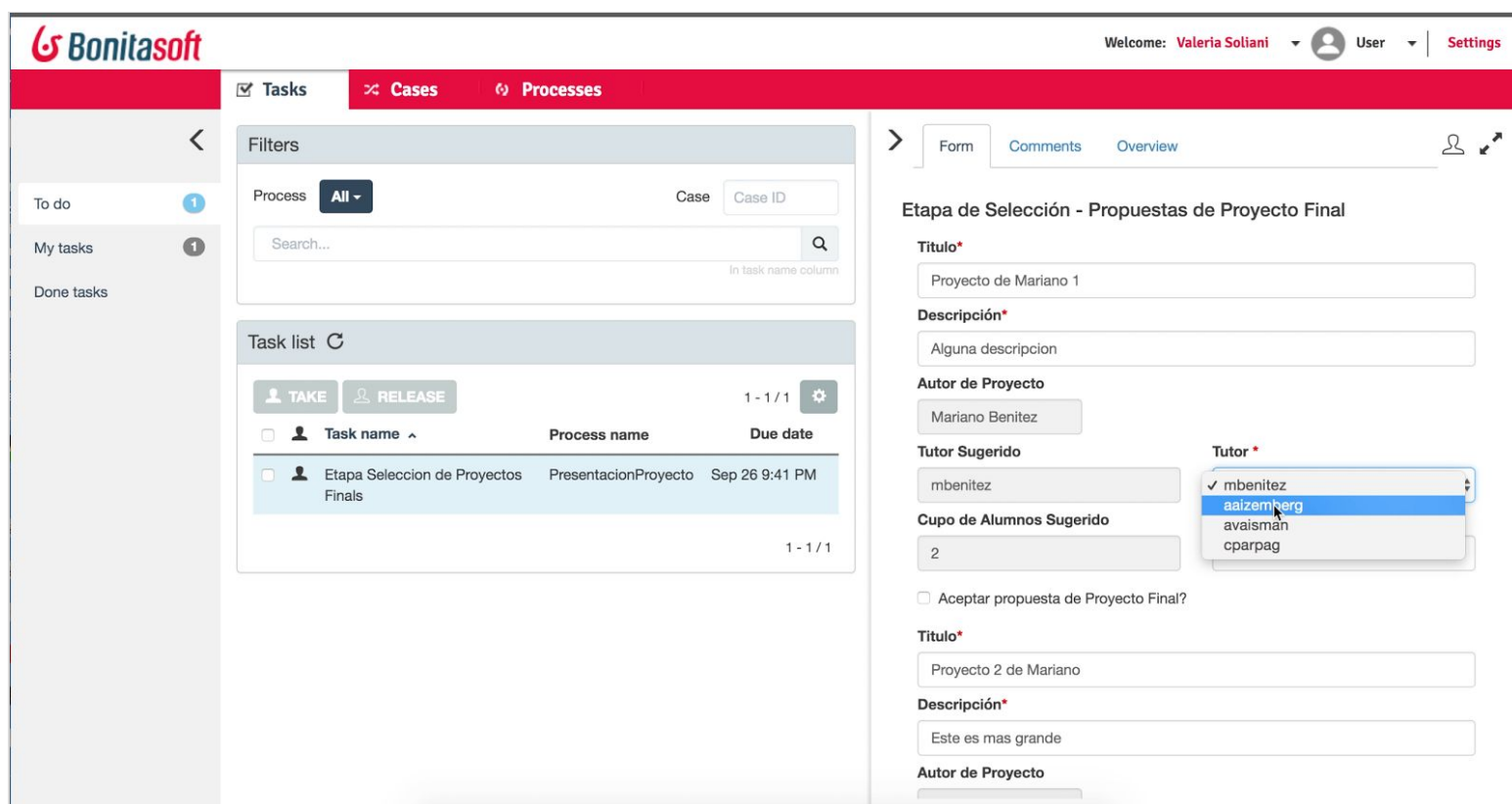
Se contempla que ambos flujos hayan terminado para continuar con el proceso.

9. Obtener tutores

Se obtienen los usuarios que poseen el rol de tutor para que luego sean asignados a los proyectos.

10. Selección de proyectos presentados

Se crea una tarea que será completada por el usuario con rol de BPM Admin. Cuando este entre a la plataforma verá un formulario con todos los proyectos presentados y podrá editarlos y elegir cuales podrán ser seleccionados durante el próximo ciclo lectivo. Dicho formulario le permite al usuario con rol de BPM Admin guardar sus cambios y enviar los proyectos seleccionados como puede verse en la *figura 5*.



The screenshot displays the Bonitasoft BPM Admin interface. The top navigation bar includes the Bonitasoft logo, a welcome message for Valeria Soliani, a user profile icon, and a settings link. The main interface is divided into three sections: a left sidebar with navigation links (To do, My tasks, Done tasks), a central task list, and a right-hand form area.

The central task list shows a table with the following data:

Task name	Process name	Due date
Etapa Seleccion de Proyectos Finals	PresentacionProyecto	Sep 26 9:41 PM

The right-hand form area is titled 'Etapa de Selección - Propuestas de Proyecto Final' and contains two forms for project selection. The first form has the following fields:

- Título***: Proyecto de Mariano 1
- Descripción***: Alguna descripcion
- Autor de Proyecto**: Mariano Benitez
- Tutor Sugerido**: mbenitez
- Cupo de Alumnos Sugerido**: 2
- Tutor ***: A dropdown menu with options: mbenitez (selected), aaizem, aavismán, cparpag.
- ☐ Aceptar propuesta de Proyecto Final?

The second form has the following fields:

- Título***: Proyecto 2 de Mariano
- Descripción***: Este es mas grande
- Autor de Proyecto**: (empty field)

Figura 5: Selección de proyectos presentados.

11. Condición de proyectos listos para publicar

En caso de que el usuario con rol de BPM Admin seleccione la opción de guardar en lugar de la de publicar esta condición dará falso y volverá al *paso 10* para seguir editando el listado de proyectos. En caso de publicarlos esta condición se cumple y se continúa con el flujo.

12. Remover proyectos no seleccionados

Cuando se decide que los proyectos van a ser publicados en esta tarea se remueven de la lista aquellos que no hayan sido seleccionados.

13. Iniciador de subproceso - emails proyectos seleccionados - alumnos autores de proyectos

Se inicia el subproceso de envío de mail (ver diagrama del subproceso en la *figura 9*) a los alumnos que hayan presentado alguno de los proyectos que fueron seleccionados. En este email (ver *figura 10*) se les notifica la aceptación de su proyecto e invita al alumno a elegir su equipo en la plataforma.

14. Iniciador de subproceso - emails proyectos seleccionados - alumnos

Se inicia el subproceso de envío de mail (ver diagrama del subproceso en la *figura 11*) a los alumnos con los proyectos a elegir. En este email (ver *figura 12*) se detallan los proyectos disponibles y se invita a los alumnos a ingresar a la plataforma para elegir el proyecto a realizar.

15. Condición de finalización de armado de grupos

Para que se dé por finalizado el proceso de selección de grupos se estipula que el mismo tomará una cierta cantidad de días definida por el usuario con rol de BPM Admin. Una vez cumplida esta cantidad de días la condición se cumple y el proceso sigue su flujo normal.

16. Email resultado de grupos

Esta es la etapa final del proceso donde se envía un email (Ver *figura 6*) al usuario con rol de BPM Admin con todos los grupos que se formaron y con los alumnos que quedaron sin elegir grupo (idealmente no debería haber alumnos que se encuentren en esta condición y todos deberían estar asignados a un grupo).

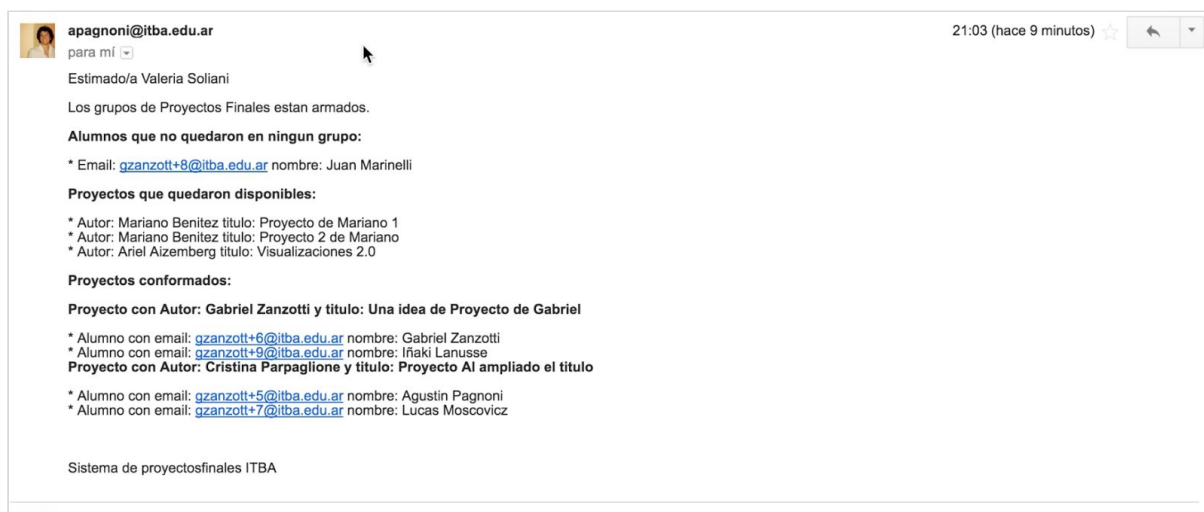


Figura 6: Email de resultado de grupos al finalizar el proceso.

4.2 Agregar proyecto final

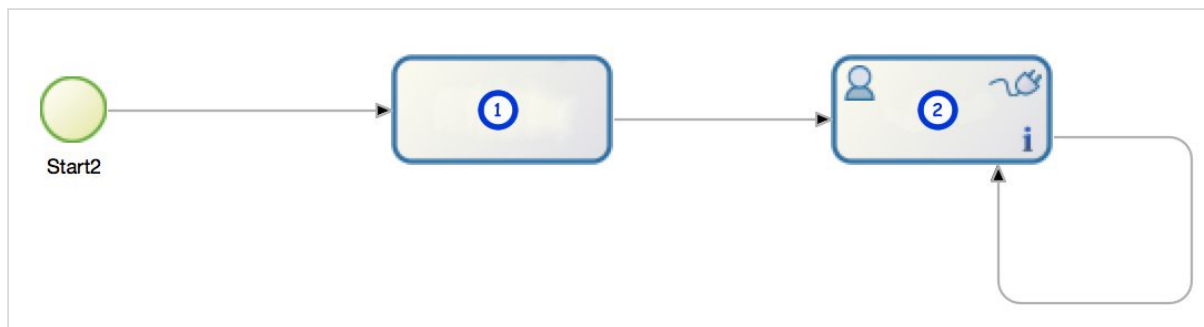


Figura 7: Diagrama del flujo de agregar proyectos.

1. guardar tutores en variable de proceso

Aquí se guardan los tutores en una variable de proceso para que puedan ser accedidos dentro del formulario del paso siguiente.

2. formulario de presentación de proyecto

En esta etapa tanto los alumnos como los profesores presentan sus propuestas de proyectos completando un formulario con los campos de título, descripción, tutor y cantidad de alumnos, dicho formulario puede verse en la *figura 8*.

Figura 8: formulario de presentación de proyecto.

4.3 Aceptación de proyecto de alumno autor

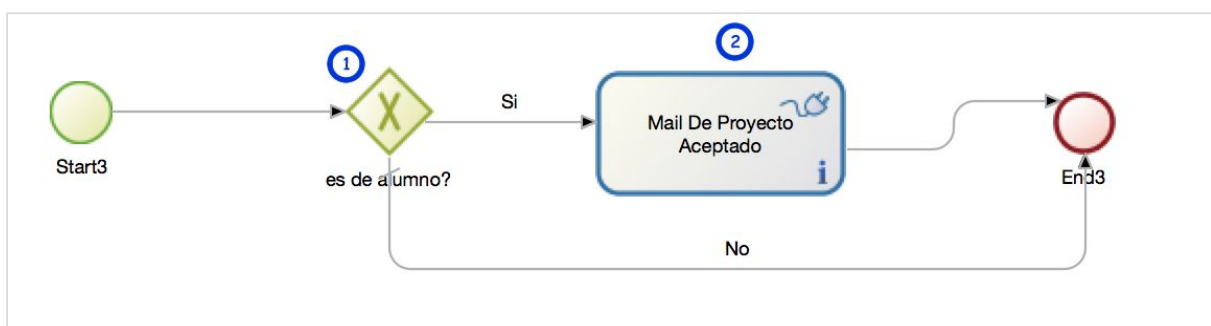


Figura 9: Diagrama del flujo de aceptación de proyectos de alumnos autores.

1. Condición de verificación de alumno autor

Se utiliza esta condición para verificar si el proyecto que ingresa a este proceso es de un alumno autor. Caso contrario este proceso termina.

2. mail de proyecto aceptado a alumno autor

En esta etapa se le envía un email al alumno autor del proyecto elegido invitándolo a ingresar a la plataforma para que envíe su propuesta a los compañeros con los que quiere realizar su proyecto. Dicho email puede verse a continuación en la *figura 10*.



Figura 10: email que le confirma a un alumno que su propuesta de proyecto fue aceptada para ser realizada este cuatrimestre.

4.4 Listado de proyectos disponibles

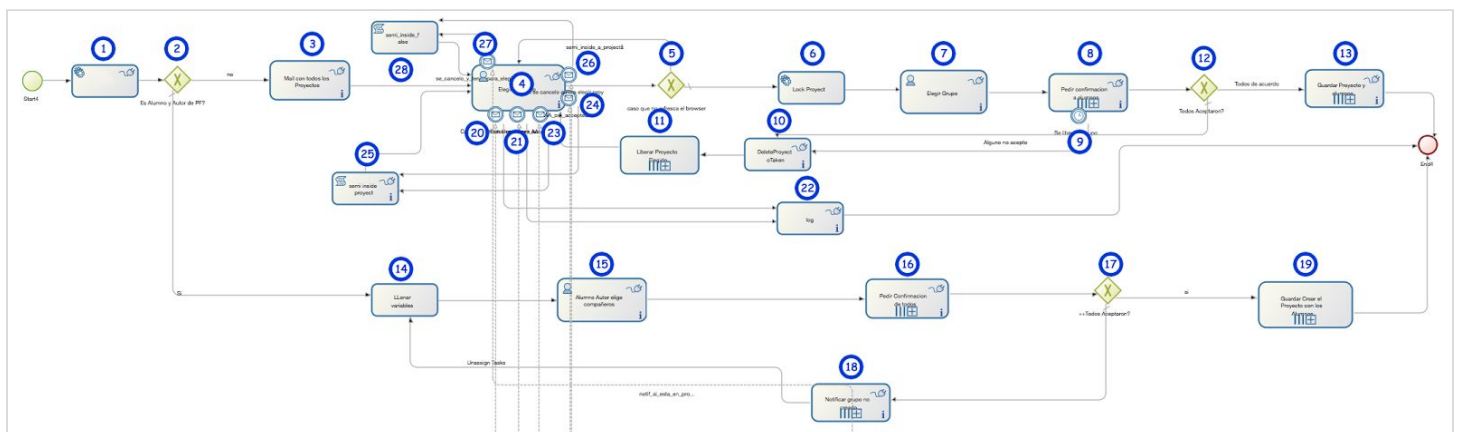


Figura 11: Diagrama del flujo de selección de grupo y proyecto.

1. Log de proyectos finales seleccionados

Aquí se crea un log que muestra los proyectos que fueron seleccionados por el usuario con rol de BPM Admin para ser elegidos por los alumnos.

2. Condición de verificación de alumno autor

En esta etapa se diferencia el proceso de selección de proyectos de un alumno que presentó un proyecto y este fue elegido del resto de los alumnos ya que el primero no tendrá que elegir proyecto mientras que el resto sí deberá comenzar por realizar la selección del proyecto a desarrollar.

3. Email con el listado de proyectos a alumnos no autores

En esta etapa se envía por email el listado de proyectos seleccionados a todos aquellos alumnos que no sean autores de proyectos seleccionados, dicho email puede verse a continuación en la *figura 12*.

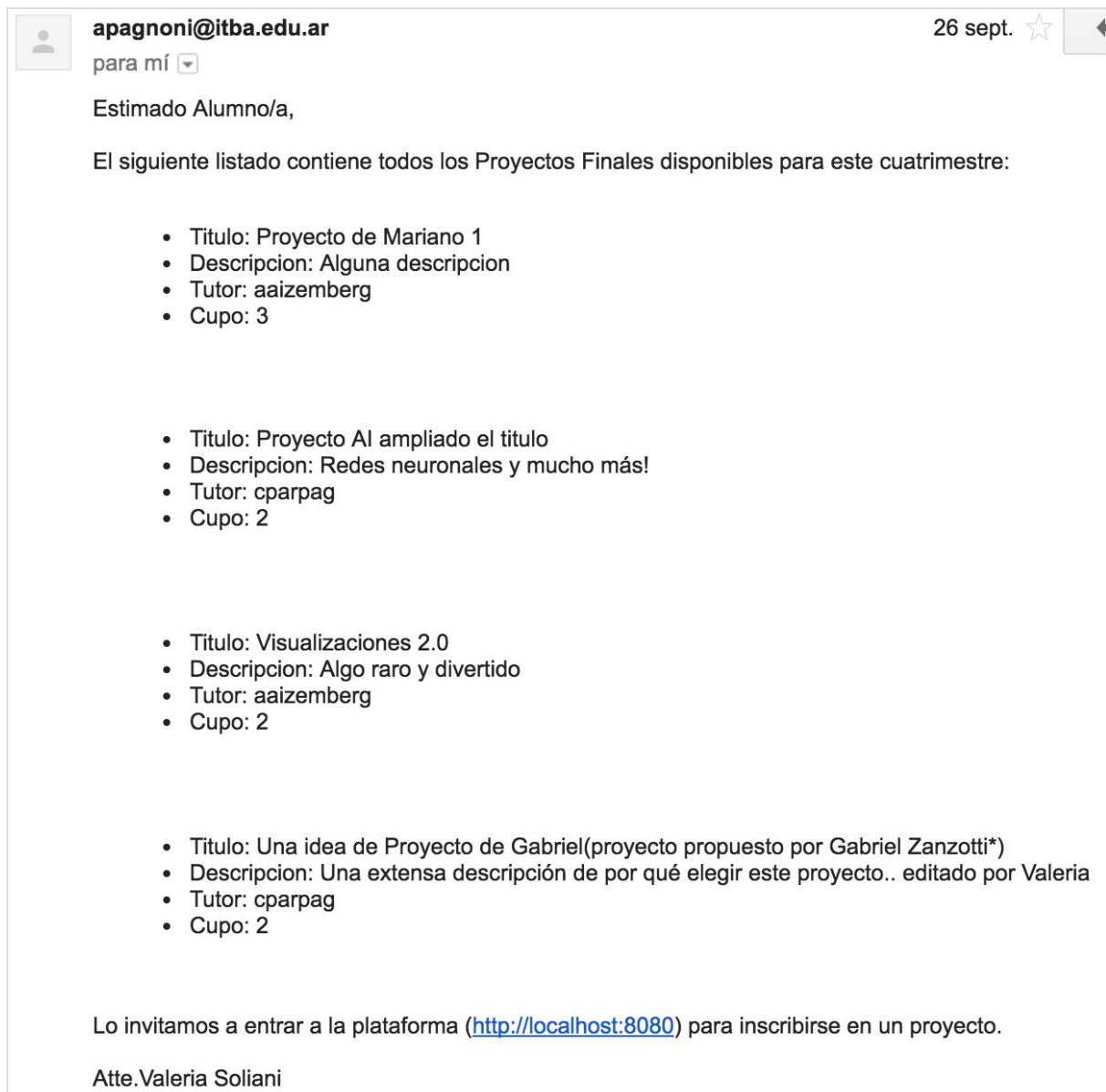


Figura 12: email con el listado de proyectos disponibles este cuatrimestre.

4. Formulario de elegir proyecto

Esta es la tarea inicial de este subproceso, aquí los alumnos que no son autores de un proyecto final pueden elegir el proyecto que desean realizar mediante un formulario en la plataforma (dicho formulario puede verse en la *figura 13*).

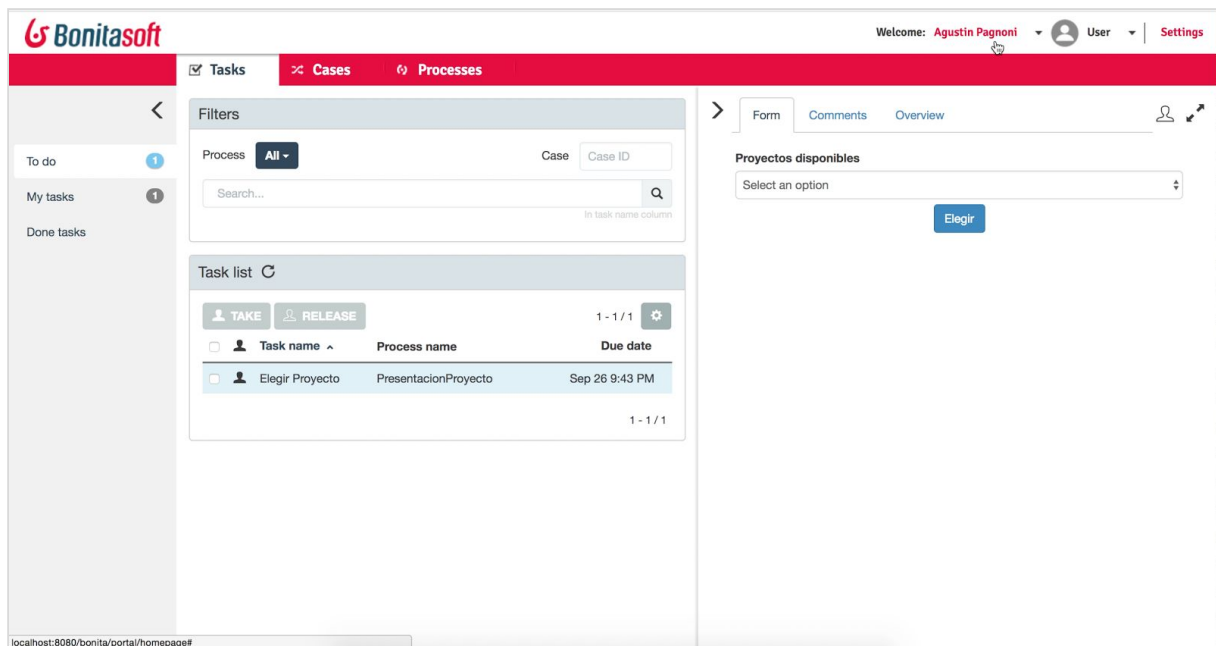


Figura 13: Formulario para elegir proyecto.

5. Condicional - ya aceptó un proyecto

Aquí se busca contemplar un caso borde en el cual una persona ya aceptó un proyecto final que le fue propuesto, pero el mismo todavía no fue confirmado por todos los miembros del equipo. Este condicional se encarga de verificar que el usuario no haya aceptado otro proyecto cuando intenta crear este nuevo. En caso de haber aceptado otro proyecto vuelve al paso anterior a la espera de confirmación o cancelación del grupo (en la *figura 19* se muestra el mensaje frente a esta situación) mientras que el caso comun seria que no haya aceptado otra propuesta y se prosigue con la creación del grupo.

6. Bloqueo del proyecto

Con el objetivo de prevenir que un alumno elija un proyecto y no un grupo para asegurarse de tener ese proyecto para el implementamos un bloqueo de proyectos temporales. Aquí se asocia el proyecto al alumno que lo eligió por un periodo de 15 minutos. De esta forma el equipo tiene tiempo suficiente para confirmar el grupo y este proyecto no podrá ser elegido por otro equipo durante dicho lapso de tiempo.

7. Formulario de elegir el grupo

Una vez definido el proyecto el alumno deberá completar un segundo formulario con los miembros de su equipo. Dicho formulario puede verse en la *figura 14*.

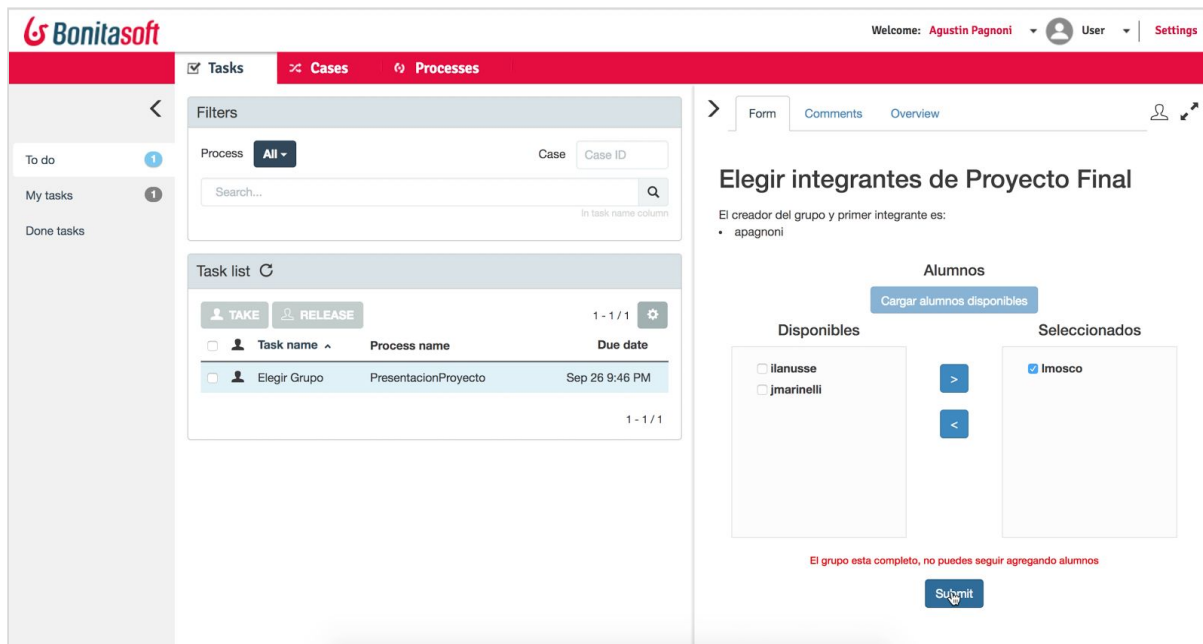


Figura 14: formulario para elegir integrantes del grupo.

8. Subproceso para pedir confirmación del grupo

Se crea un subproceso para que cada uno de los miembros del grupo pueda confirmar su participación en el mismo o pueda cancelarlo si no está de acuerdo con este (Ver *Figura 16*).

9. Condición de cancelación de proyecto

En caso de que el grupo no haya completado la confirmación a tiempo, cuando se cumplan los 15 minutos se ejecutará esta condición para volver al estado inicial de selección de proyecto.

10. Borrado de proyecto seleccionado

En esta etapa se desbloquea el proyecto elegido por el alumno ya que el grupo no completó a tiempo la confirmación.

11. Subproceso para notificar grupo cancelado

Aquí se ejecuta para cada uno de los integrantes del grupo el subproceso de la *figura 20*. Dicho subproceso será el encargado de notificarle a los alumnos involucrados en la propuesta de grupo que él mismo fue cancelado por falta de confirmación.

12. Condicional de aceptación del proyecto

Una vez concluido el *paso 8* se verifica con este condicional si los todos los alumnos aceptaron el grupo o si alguno lo cancelo. En caso de una cancelación se ejecutará el *paso 10* mientras que el flujo normal llevara a la consolidación del grupo.

13. Consolidación del grupo

13.1 Persistencia de alumnos y proyecto

En este paso se guarda el grupo confirmado y tanto los alumnos como el proyecto no pueden ser elegidos para futuros proyectos.

13.2 Subproceso de confirmación de grupo

Además de realizar la persistencia del grupo se ejecuta el subproceso de la *figura 20* para notificar a los miembros del mismo la confirmación del grupo.

14. Inicialización del proyecto del alumno autor

En esta etapa se inicializa el proceso de los alumnos autores. Aquí se precarga el proyecto del alumno autor. Siendo este el proyecto a realizar por el grupo.

15. Alumno autor elige grupo

Esta es la primer etapa de los alumnos autores, etapa en la cual eligen el grupo con quien quieren desarrollar el proyecto propuesto. Para esto deberán completar un formulario en la plataforma (dicho formulario puede verse en la *figura 15*).

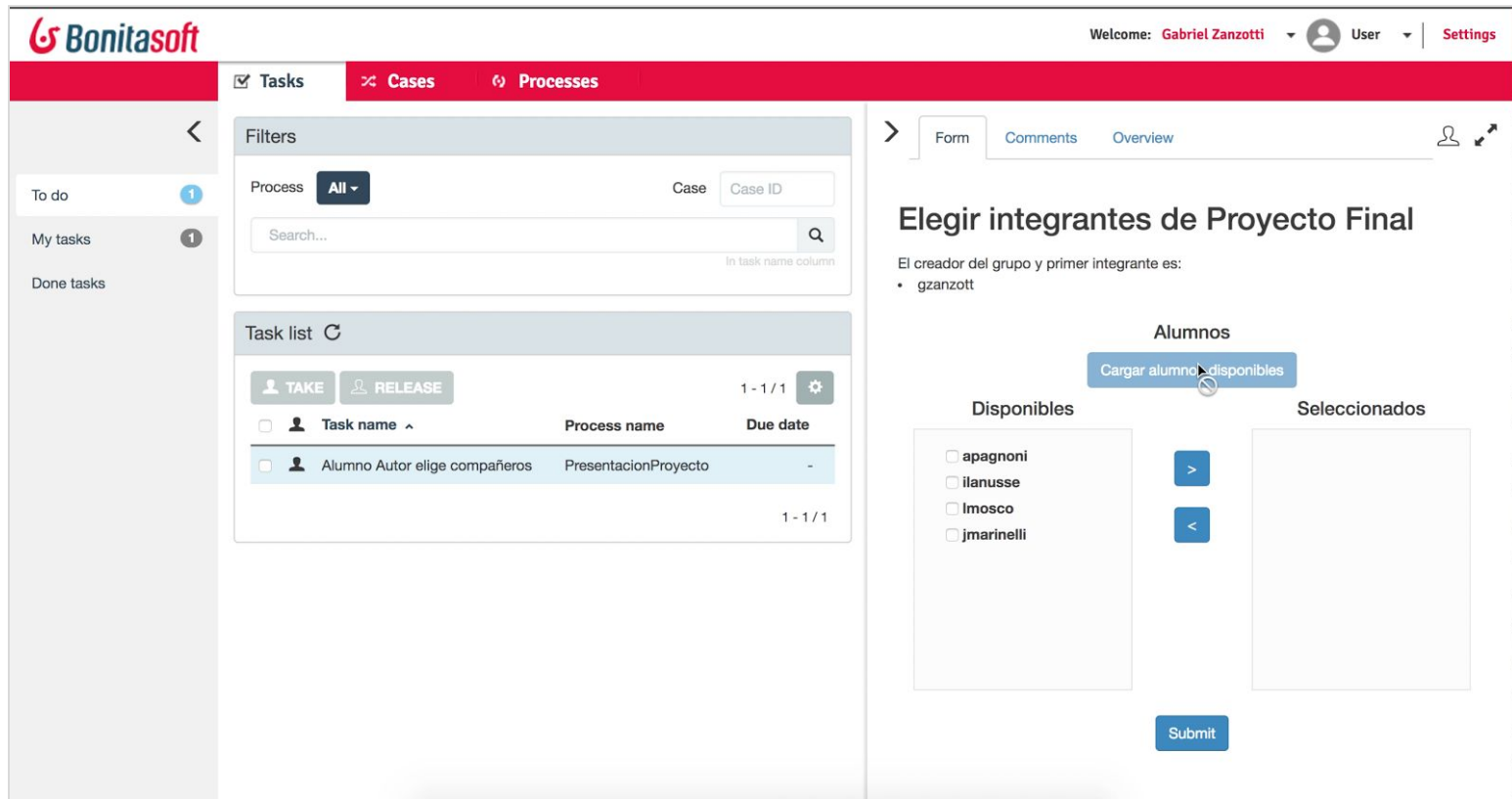


Figura 15: Alumno autor elige grupo.

16. Subproceso de confirmación de grupo

Una vez elegido el grupo, se corre el subproceso de la *figura 23* para que los alumnos confirmen o cancelen la propuesta del alumno autor.

17. Condicional de aceptación del proyecto

Al igual que en el *paso 12*, aquí se verifica que todos los alumnos que fueron invitados a participar del proyecto del alumno autor hayan confirmado su participación para confirmar el grupo. En caso de una cancelación, el alumno autor tendrá que realizar el proceso nuevamente.

18. Subproceso para notificar grupo cancelado

Si algún alumno no acepta la propuesta del alumno autor se ejecuta el subproceso de la *figura 24* para notificarle a todos los alumnos involucrados la cancelación del grupo.

19. Consolidación del grupo

19.1 Persistencia de alumnos y proyecto

En este paso se guarda el grupo confirmado y tanto los alumnos como el proyecto no pueden ser elegidos para futuros proyectos.

19.2 Subproceso de confirmación de grupo

Además de realizar la persistencia del grupo se ejecuta el subproceso de la *figura 24* para notificar a los miembros del mismo la confirmación del grupo.

20. Mensaje recibido - desasignar tarea por confirmación de grupo

Este evento de mensaje recibido es activado cuando se termina de confirmar un grupo en el *paso 4* del proceso de la *figura 20*. Su objetivo es desasignar la tarea de creación de grupo dado que esta ya no es necesaria.

21. Mensaje recibido - desasignar tarea por confirmación de grupo de alumno autor

De la misma manera que el mensaje anterior, este es recibido cuando se confirma el grupo, solo que en este caso es para proyectos de alumnos autores. El flujo siguiente es el mismo que en el *paso 20* y es llamado desde el *paso 8* del proceso de la *figura 24*.

22. Log de desasignación de tarea por completitud de proyecto

Esta tarea es llamada luego de la recepción de un mensaje en el *paso 20* o el *paso 21*, se realiza un log de lo ocurrido y se finaliza el proceso.

23. Mensaje recibido con la confirmación de un alumno.

Cuando un alumno confirma un grupo se envía este mensaje el cual se encarga de que dicho alumno no pueda inscribirse a otro proyecto hasta que el grupo que acaba de confirmar se confirme o se cancele.

24. Mensaje recibido con la confirmación de un alumno para un proyecto propuesto por un alumno

Al igual que en el paso anterior este mensaje se activa cuando un alumno confirma un grupo, que en este caso es de un alumno autor. Su objetivo es que dicho alumno no pueda generar otro grupo hasta que el que aceptó previamente se confirme o se cancele.

25. Marcar alumno como confirmado para un grupo

Esta tarea es la encargada de ejecutarse al recibir un mensaje del *paso 23* o del *paso 24*. Aquí se crea una variable de tipo booleana que le impedirá al alumno crear otro grupo.

26. Mensaje de cancelación de proyecto de alumno autor

Este mensaje se ejecuta cuando se cancela una propuesta de grupo en el proyecto de un alumno autor en el *paso 6* del subproceso de la *figura 24*.

27. Mensaje de cancelación de proyecto

Al igual que en el paso anterior este paso es llamado cuando se cancela la propuesta de un grupo en el *paso 6* del subproceso de la *figura 20*.

28. Marcar alumno como confirmado para un grupo

En esta tarea se desmarca al alumno que confirmó el proyecto que le fue propuesto pero que terminó cancelandose por otro alumno. Esto le permite a este alumno poder proponer un proyecto si lo desea.

4.5 Confirmación o cancelación de proyectos

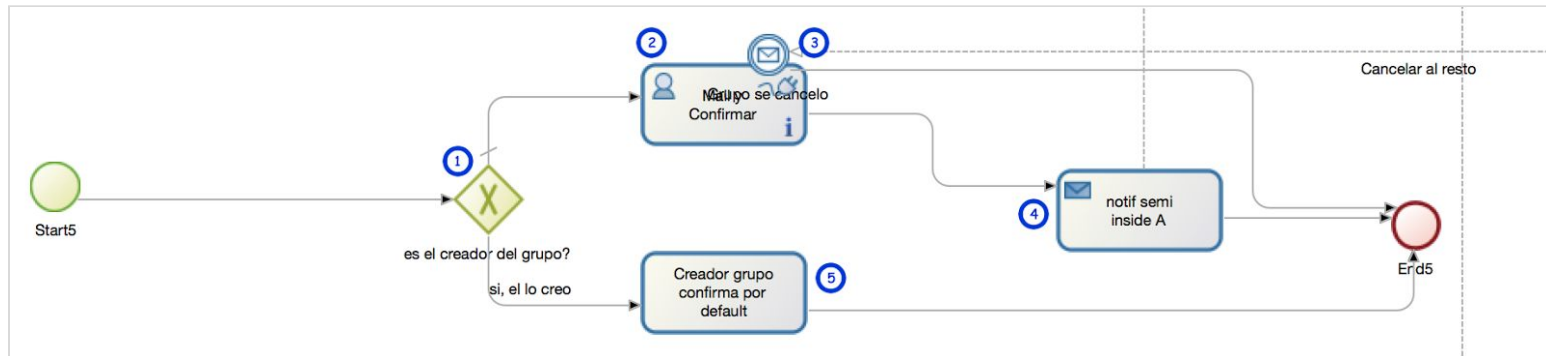


Figura 16: Diagrama del flujo de confirmación o cancelación de una propuesta de grupo.

1. Condicional - es creador del grupo

En esta etapa se divide el flujo en dos, el primero relacionado a si este proceso está asignado al alumno que propuso el grupo y el segundo al caso contrario.

2. Email de confirmación o cancelación

Aquí se le envía un email a los alumnos del grupo (tal como se muestra en la *figura 17*) pidiéndoles que confirmen o cancelen el proyecto que les fue propuesto. En dicho mail se les solicita ingresar a la plataforma y completar el formulario de aceptación o cancelación que se muestra en la *figura 18*.

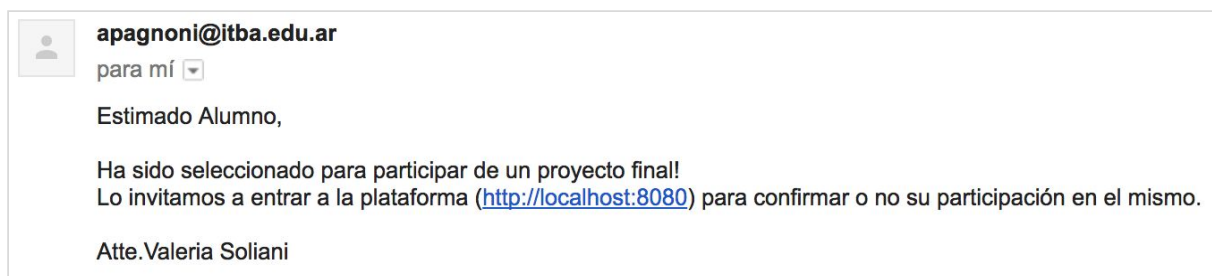


Figura 17: alumno es invitado a participar de un proyecto final.

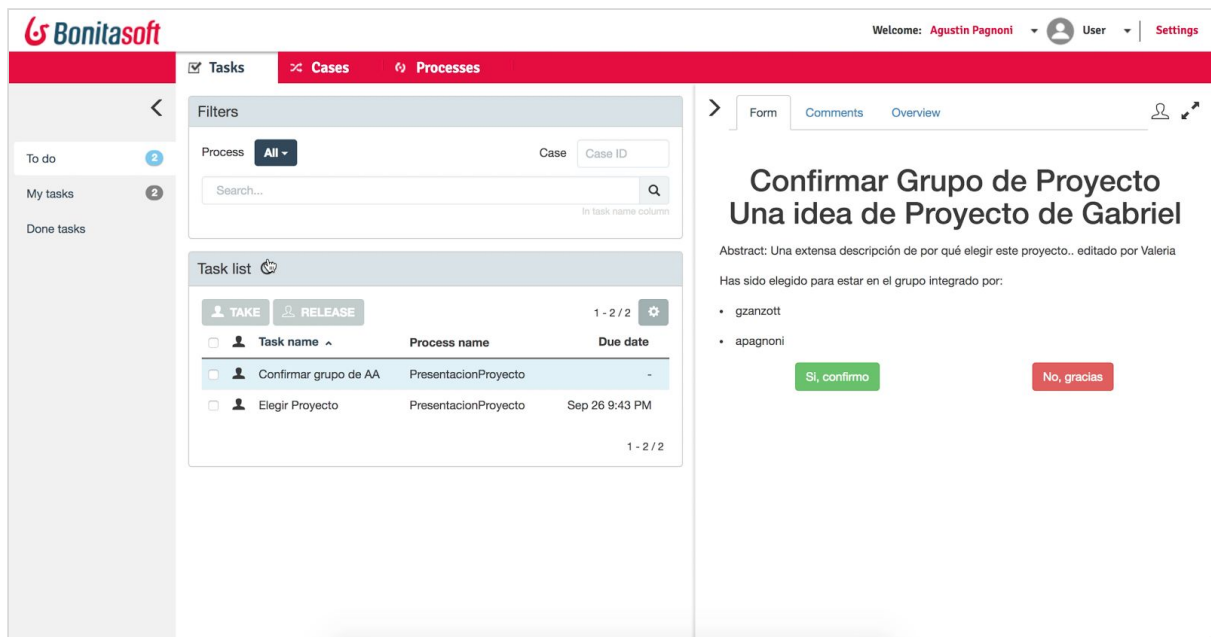


Figura 18: formulario de confirmación de grupo.

3. Mensaje finalización del subprocesso por cancelación

Este evento se ejecuta cuando un alumno cancela el grupo a través del *paso 8* del proceso de la *figura 24*. Dicha acción repercute en que el resto del grupo deje de tener esta tarea y finalice este subprocesso.

4. Lanzamiento de mensaje - alumno confirmó el proyecto

En esta tarea se envía un mensaje para notificar que el alumno ya aceptó un proyecto y que este no puede proponer otro grupo hasta que alguno de los otros integrantes de este grupo cancele el proyecto propuesto. Como se puede ver en la *figura 19* no se puede avanzar del primer paso de propuesta de grupo ya que este alumno aceptó una propuesta que todavía no fue confirmada o cancelada.

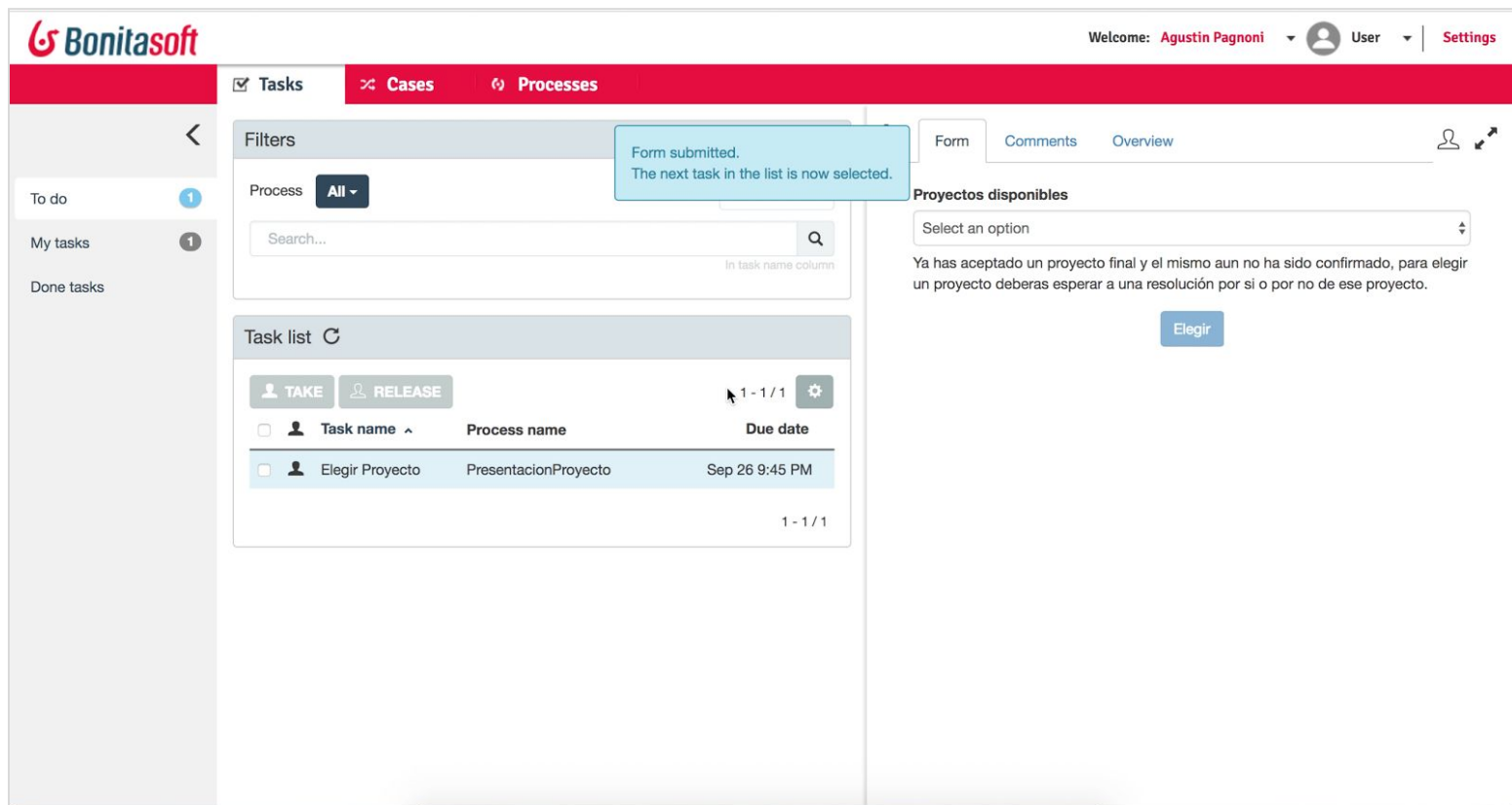


Figura 19: Lanzamiento de mensaje - alumno confirmó el proyecto

5. Confirmación del alumno creador del grupo

Quien crea el grupo confirma automáticamente, la acción que realiza esta tarea es marcar como confirmado al alumno que creó el grupo.

4.6 Notificar confirmación o cancelación de proyecto

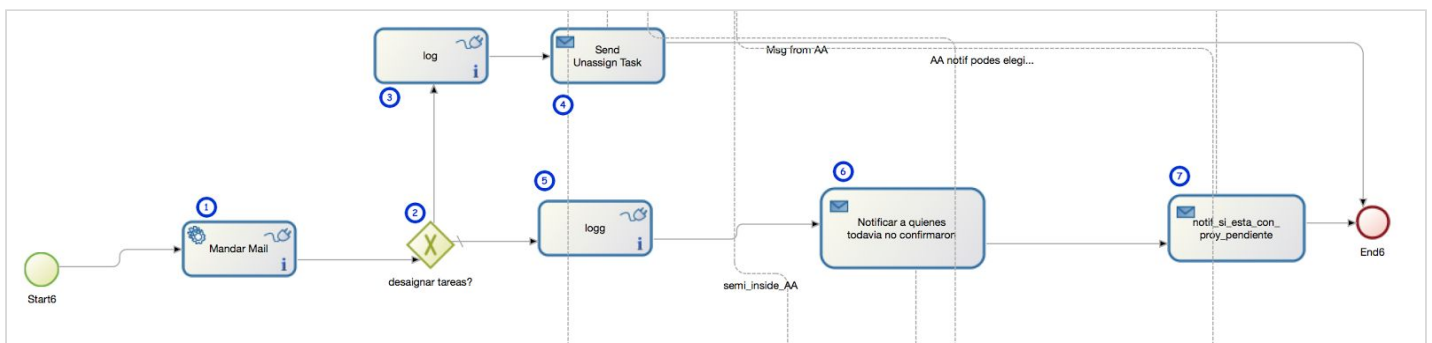


Figura 20: Diagrama del flujo de notificación a miembros de una propuesta de grupo tras una confirmación o cancelación del mismo.

1. Email notificando aceptación o cancelación del grupo

En este momento dependiendo si el grupo fue aceptado o cancelado se envía un mail informándole al alumno que su grupo fue consolidado o que algún alumno rechazó la propuesta de grupo. En la *figura 21* puede verse un ejemplo de una confirmación de grupo mientras que en la *figura 22* puede verse un ejemplo de un grupo que fue cancelado.

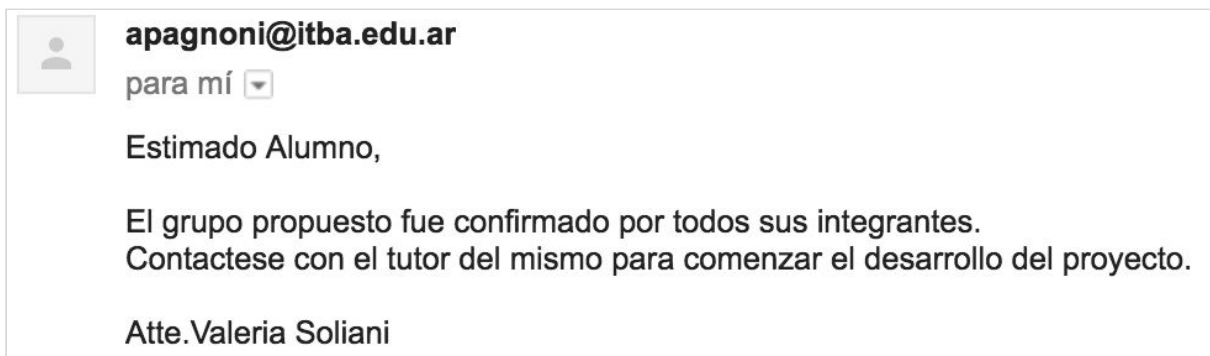


Figura 21: Email de grupo confirmado.

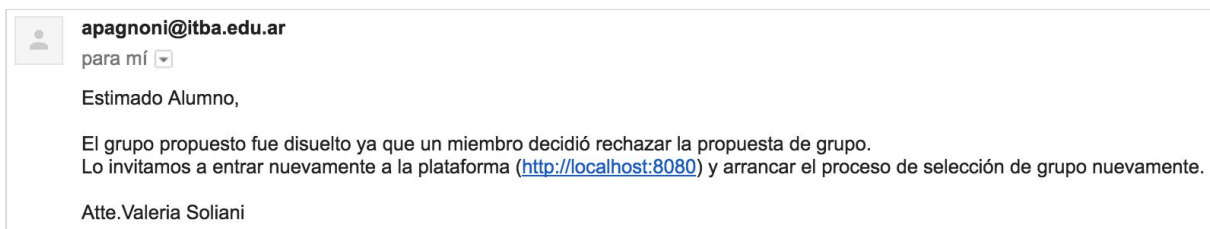


Figura 22: Email de grupo cancelado.

2. Condicion grupo confirmado

aquí se evalúa si el grupo fue aceptado o cancelado para determinar el flujo a seguir a seguir.

3. Log grupo confirmado

Esta tarea se encarga de generar un log informando que el grupo fue consolidado.

4. Enviar mensaje de confirmación de grupo

En este paso se envía un mensaje al subproceso de la *figura 11*, su finalidad es ejecutar el evento del *paso 20* de dicho

subproceso y hacer que este concluya ya que el grupo fue confirmado.

5. Log grupo cancelado

Aquí se genera un log con el objetivo de registrar el grupo que fue cancelado.

6. Cancelar confirmaciones

En esta tarea se envía un mensaje al subproceso de la *figura 11* con el objetivo de cancelar las confirmaciones de los alumnos que fueron invitados al grupo pero que todavía no confirmaron ya que la propuesta de grupo fue declinada.

7. Volver a habilitar la creación de proyectos

Dado que el grupo en cuestión fue cancelado se envía un mensaje al subproceso de la *figura 11* para que los alumnos de dicho grupo puedan volver a tener la posibilidad de crear un grupo.

4.7 Confirmación o cancelación de proyecto de un alumno

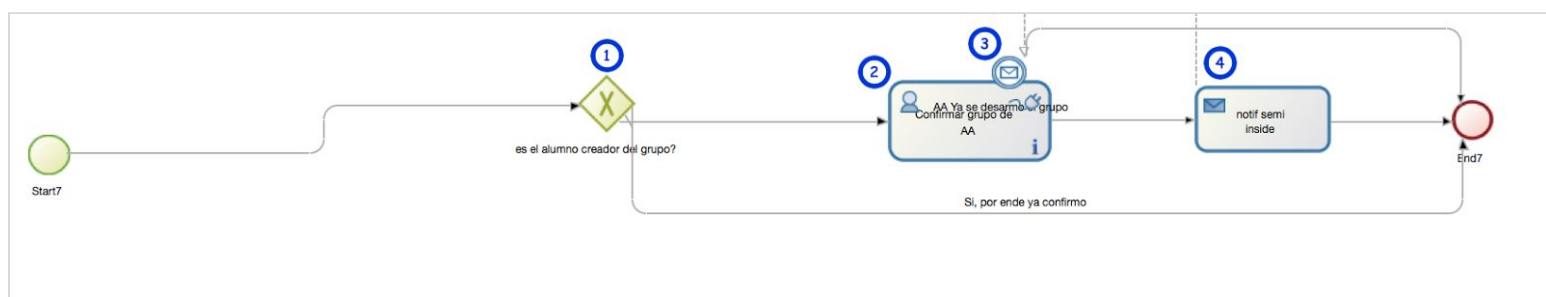


Figura 23: Diagrama del flujo de confirmación o cancelación de una propuesta de grupo en el proyecto de un alumno.

1. Condición es creador de grupo

Similar a lo que ocurre en el subproceso de la *figura 16*, se define qué camino del flujo se tomará dependiendo si el

usuario que se encuentra en este subproceso es quien propuso el grupo (el alumno autor del proyecto) o no.

2. Email de confirmación de grupo en un proyecto de alumno autor

Se envía un email a los alumnos del grupo propuesto (excepto al alumno autor del proyecto). Este email le pide a los alumnos que confirmen o cancelen el grupo para continuar el proceso.

3. Mensaje de grupo ya cancelado

En caso de que un alumno decline la propuesta de grupo la tarea del *paso 5* de la *figura 24* envía un evento que hace que este subproceso concluya para todos aquellos alumnos del grupo propuesto que todavía no hayan contestado.

4. Lanzamiento de mensaje - alumno confirmó el proyecto

Aquí, al igual que en el *paso 4* del subproceso de la *figura 16* se envía un mensaje notificando que el alumno confirmó el grupo, la finalidad nuevamente es que dicho alumno no pueda proponer otro grupo nuevo hasta que el que acaba de aceptar no se confirme o cancele.

4.8 Notificar confirmación o cancelación de proyecto de un alumno

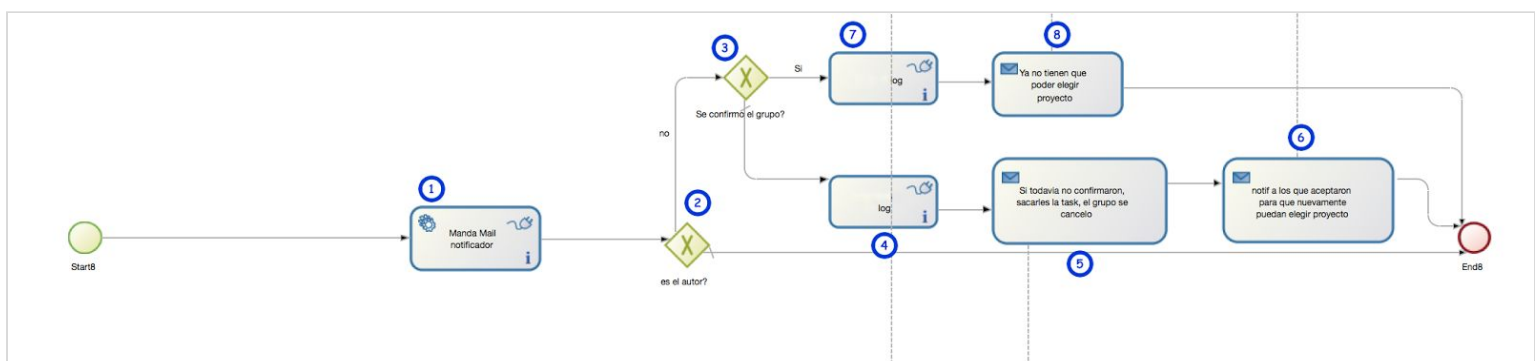


Figura 24: Diagrama del flujo de notificación de confirmación o cancelación de una propuesta de grupo en el proyecto de un alumno.

1. Envío de email de confirmación o cancelación de la propuesta de grupo

Se envía un email a los alumnos del grupo propuesto (excepto al alumno autor del proyecto). Este email contiene un mensaje notificándoles si el grupo propuesto fue confirmado por todos los alumnos que lo conforman o si alguno canceló la propuesta.

2. Separación en casos dependiendo de si es el autor o no

Debido a que el alumno autor del proyecto es quien creó la propuesta y la finalización de este subproceso directamente llevará a que el alumno pueda volver a proponer un grupo se plantea esta condición para que se realicen ciertas tareas en caso de que el alumno que se encuentra en este subproceso no sea el alumno autor (es decir el alumno que propuso el grupo).

3. Condicion grupo confirmado

Se contempla si es el caso de que el grupo fue confirmado o cancelado para realizar las acciones correspondientes.

4. Log - grupo cancelado

Se genera un log notificando la cancelación del grupo por cualquier eventualidad que lo requiera.

5. Enviar mensaje para que los alumnos no tengan que confirmar

Debido a que el grupo ya fue cancelado no tiene sentido que el resto de los miembros del grupo propuesto tengan que confirmar. Es por eso que aquí se envía un mensaje al *paso 3* del subproceso de la *figura 23* para que se termine esta tarea automáticamente.

6. Desbloquear alumnos que ya aceptaron

En esta tarea se envía un mensaje al *paso 26* del subproceso de la *figura 11*. Este mensaje hace que los alumnos que ya habían

aceptado el grupo propuesto y se les bloqueó la posibilidad de proponer otro grupo vuelvan a tener dicha posibilidad dado que la propuesta actual fue cancelada.

7. Log - confirmación de grupo

Se busca generar un log del equipo que fue confirmado por si esta informacion es requerida para algun control en el futuro.

8. Enviar mensaje de finalización de selección de grupo

Como este grupo ya fue confirmado esta tarea se encarga de que se envíe un mensaje al *paso 21* del subproceso de la *figura 11* para que este subproceso termine.

5. Casos de uso

Plantearemos 4 casos de uso comunes de la aplicación definiendo para este ejemplo una organización tiene:

- ❖ 1 usuario con rol de BPM Admin
- ❖ varios usuarios con rol de profesores (que a su vez tendran tambien rol de tutores)
- ❖ varios usuarios con rol de alumno

Todos los casos comienzan con un email que recibe el usuario con rol de BPM Admin en el cual se le pide que ingrese a definir las fechas de inicio del proceso. Una vez que el usuario con rol de BPM Admin define la fecha de presentación de proyectos de alumnos y profesores se espera a esa fecha para dar comienzo a cada subproceso. Cuando la presentación de proyectos por parte de profesores comienza, todos los usuarios con rol de profesor reciben un email invitándolos a proponer proyectos. El mismo email se le envía a los alumnos cuando se cumple la fecha de inicio de alumnos. Una vez finalizado el periodo de presentación de proyectos el usuario con rol de BPM Admin define cuales de estos proyectos serán elegibles este cuatrimestre, pudiendo editar cualquier dato de

estos previamente. Luego de definir los proyectos el proceso se amplía a varios casos. A continuación explicaremos los 4 casos mas comunes.

5.1 Alumno creador de proyectos

Si el alumno no propuso ningún proyecto que haya sido elegido recibirá un email notificándole los proyectos que hay disponibles para ser realizados este cuatrimestre. Al iniciar sesión en la plataforma podrá elegir armar una propuesta de grupo. Para esto eligirá el proyecto que sea de su interés. Una vez elegido, éste no podrá ser seleccionado por el resto de los alumnos durante los próximos 15 minutos. Esto le permitirá al alumno elegir al resto de los integrantes para realizar la propuesta de grupo. La cantidad de alumnos a invitar estará delimitada por el proyecto elegido. Una vez realizada la selección de grupo el alumno que envió la propuesta no tiene más tareas para realizar y se le envía un email pidiendo confirmación a los alumnos elegidos. Una vez que todos los alumnos aceptan el grupo queda confirmado y cuando finaliza el armado de grupos se le envía al BPM Admin un email con los grupos armados.

5.2 Alumno autor de un proyecto

Si el proyecto que un alumno propuso es aceptado este alumno recibirá un email diferente al del resto notificándole que su proyecto fue aceptado. Cuando este alumno entre a la plataforma no tendrá que elegir un proyecto ya que él está automáticamente agregado a su proyecto y es la única persona que puede armar el grupo para su proyecto. Una vez elegido el grupo el flujo continua de la misma manera que en el caso 5.1.

5.3 Alumno que participa en un grupo

Un alumno es invitado a participar de un grupo cuando recibe un email notificándole acerca de la invitación. Una vez aceptada la misma el alumno invitado no tiene que realizar ninguna otra acción.

5.4 Grupo cancelado y armado de un nuevo grupo

Retomando el caso 5.1 hasta el punto en el que el alumno realiza la propuesta de grupo, si algún otro alumno rechaza la propuesta se le enviará un email a todos los participantes de la propuesta informándoles que el grupo fue cancelado. Esto le permitirá a todos los participantes de esta propuesta poder tener la posibilidad de armar un nuevo grupo y que el ciclo se repita hasta que el equipo se consolide.

6. Conclusiones

Llevar a cabo este proyecto nos dimos cuenta de la complejidad intrínseca en utilizar software que tiene tanto una versión gratis para la comunidad y otra reservada para clientes pagos. En experiencias pasadas, la diferencia de funcionalidades y/o comunidad activa no era un impedimento, pero con Bonita BPM el caso fue distinto. Comenzamos diciendo esto ya que la falta de comunidad y ciertas restricciones que detallaremos más adelante complejizaron más de la cuenta una funcionalidad relativamente simple de implementar en otras tecnologías, no necesariamente orientadas a BPM.

Más allá de las dificultades, estamos de acuerdo en el beneficio de utilizar BPMN (hoy materializado en Bonita), para estandarizar la tecnología y la documentación de los procesos activos de una organización grande, con desarrolladores distintos, como lo es una

universidad. Es importante destacar que una persona administrativa o de negocio pueda visualizar el flujo de un proceso dentro de su organización, y así detectar posibles optimizaciones, o puntos de falla/cuellos de botella.

Algunas notas curiosas de la versión de comunidad de Bonita BPM:

- ❖ No permite visualizar los archivos de texto con el código que entra en cada tarea. Por lo que no admitía el uso de herramientas de versionado de código tal como *git*.
- ❖ No era posible el uso común de herramientas de debugging, por lo que se recurrió a un extensivo uso de logging para los fines prácticos. El sistema de logging “standard” de Bonita no funcionaba sin una extensa configuración por lo que también hubo que armar un sistema de logging en paralelo, lo cual no fue difícil pero no resultó un trabajo extra que no debería ser necesario.
- ❖ A lo largo de este proyecto, cambiaron muchas cosas en Bonita, con un backlog de errores conocidos muy extenso, llegando a la versión 7.5 con un aproximado de 16531 issues/bugs.
- ❖ Durante 2 años desarrollando el sistema, Bonita en su versión gratuita no te permitía interactuar con la base de datos interna, ya que no contábamos con el usuario ni la contraseña. Luego en versiones posteriores se agregó una funcionalidad para enviar *queries* a la base de datos desde un panel web, esto generó mucha incertidumbre en un principio (ocasionada por la ineficiencia) a la hora de ver los cambios en el estado de las tablas utilizadas de la base de datos.
- ❖ Nos encontramos en varias ocasiones con impedimentos cuya resolución no estaba de nuestro lado. Luego de *extensas horas* de intentos y al ver que los resultados no eran los esperados en función de la documentación, intuíamos que tenían que ser

errors de Bonita lo que explicase los hechos. Esta hipótesis luego se confirmaba al ver las notas de los cambios de versión que resolvían aquellos problemas que habíamos detectado.

- ❖ Encontrar soluciones a problemas similares en los foros de comunidad no era opción, dado la *nula actividad* dentro de los foros de la versión gratuita. Esto claramente es un factor que ralentiza el desarrollo de manera considerable.

Por otro lado, se rescatan los siguientes puntos de BPMN y Bonita:

- ❖ Nuevas integraciones con servicios, ej Google Calendar, o interfaces a lo largo del avance de versiones. La mayoría no eran requeridas para la funcionalidad requerida, pero se ve claramente la utilidad de estar integrados a APIs de terceros.
- ❖ La facilidad para establecer un nuevo flujo, es decir flujos troncales al negocio, sin escribir 1 sola línea de código.
- ❖ Portabilidad, refiriéndonos a que se puede dejar de usar Bonita y utilizar distinto software tal como Signavio, Activiti o Camunda, ya que todas son compatibles con BPMN 2.0.

En retrospectiva, el hecho de que existan previos procesos automatizados utilizando BonitaSoft Comunidad en ITBA no debería ser razón suficiente para continuar con una versión gratuita. Dicha versión entorpece el uso de servicios básicos tal como la BBDD, el filesystem de los snippets de código interno de los procesos y hasta del sistema de logging/salida estándar que no funciona simplemente. Con lo cual creemos que debería evaluarse el uso de la versión paga para futuras implementaciones.