



UNIVERSIDAD FASTA

Ingeniería en Informática – FIM 42 – Proyecto Final

**Retrospectiva de Proyecto**

# **CommonJobs**

**(Sistema de Recursos Humanos CommonSense)**

**Alumnos:**

- \* Andrés Moschini.
- \* Matías José.
- \* Juan Diego Raimondi.

**Director Funcional:** Gabriel Buyatti.

**Director Técnico:** Ing. Alejandro Fantini.

**Auditora:** Ing. Gabriela Degiampietro.

**Cátedra:**

- \* Profesor titular: AS. Hilario Fernando Schechtel
- \* Profesor asociado: Ing. Roberto Giordano Lerena
- \* Profesor asociado: Lic. Alejandro Nikolic

**Fecha de presentación:** 11/12/2012

## Tabla de Contenidos

Tabla de Contenidos .....	1
Retrospectiva de Proyecto .....	2
El problema .....	2
Las acciones .....	2
Cómo seguir .....	2
Tareas pendientes .....	2
Postulantes .....	2
Administración de los almuerzos diarios .....	3
Empleados .....	3
Detalles de terminación .....	3
Ayuda y documentación .....	3

## Retrospectiva de Proyecto

---

### El problema

Alrededor del `sprint 9` (cerca de la mitad del proyecto) detectamos que el sistema no se estaba utilizando y que muchas de las funciones que estábamos implementando no iban a ser utilizadas, al menos no en el corto plazo. En un enfoque ágil como el que intentamos seguir, esto es un problema grave desde el punto de vista del producto como utilidad para la empresa, ya que es importante que en cada `sprint` agreguemos valor al sistema. Por otro lado, el método ágil nos permite cambiar ese rumbo y corregir esos problemas, incluso en una fase tan avanzada del proyecto. Proveer la funcionalidad apropiada mejorará el uso del sistema, generando más interacción que a su vez alimentará el uso (actualmente planificado) de esas funcionalidades no utilizadas.

De este periodo anterior se rescata la construcción de la estructura, el aprendizaje, y por supuesto algo de funcionalidad.

El área de RRHH también estaba (y está) organizándose y modificando sus métodos internos. Ahora mismo, su principal necesidad está relacionada con la administración del legajo e información de los empleados.

La propuesta de proyecto inicial estaba enfocada en la selección de postulantes, actualmente el área de RRHH maneja eso con una relativa informalidad, sin seguir procesos rigurosos, pero con gran flexibilidad y eficiencia. Si bien desde el punto de vista organizacional, es importante generar un proceso, no podemos forzar la definición del mismo, ni imponerlo nosotros ya que en el sistema cae en desuso.

### Las acciones

Intentando adaptarnos a esta situación, se decidió enfocar las decisiones al corto plazo, con el nuevo sub-objetivo de hacer útil la plataforma, esperando que su uso generara necesidades y eventualmente se descubriera la funcionalidad real necesaria, sin chocar con los procesos informales realizados.

Las primeras acciones correctivas fueron revisar completamente el Product Backlog y repriorizar las tareas. El mismo dinamismo que decidimos aceptar, nos impidió planificar a largo plazo ya que las prioridades cambiaban de `sprint` a `sprint`: la línea de fin de proyecto se perdió en un Backlog poco definido.

Decidimos dedicar tiempo de los `sprints` a analizar y planificar las siguientes funciones a implementar en el corto plazo (~2 `sprints`).

Esto nos alejó de la funcionalidad original propuesta, pero permitió que el sistema comience a ser útil y sea utilizado.

### Cómo seguir

Alcanzamos el `sprint 13` (~2/3 partes del proyecto) y vimos la necesidad de prepararnos para finalizar el proyecto.

Decidimos que debemos volver a definir una línea de fin de proyecto sólida, de manera de evitar sorpresas. Para ello evaluamos los distintos requerimientos necesarios para poder considerar el proyecto terminado y lo consideramos con el director técnico y con la cátedra.

Documentaremos los cambios sobre la propuesta original en un documento que presentaremos a la cátedra un resumen de la funcionalidad implementada hasta ahora y el Product Backlog que incluya los aspectos considerados en este documento.

Debemos resolver varios aspectos de terminación que el director funcional no considera importantes, pero hacen a la calidad del producto (ej. Crear un empleado y no guardarlo genera un empleado vacío). Para esto, prefijaremos tareas técnicas o terminaciones de UI con un espacio importante en los `sprints`.

Creemos que puede haber funcionalidad que el director funcional no considera importante para el día a día pero que puede ser valiosa para un producto mejor acabado y completo a ojos de la cátedra, y útil para la empresa (ej. Flujo de postulantes). Con asesoramiento de la cátedra y del director funcional seleccionaremos esa funcionalidad y la definiremos con ayuda de algún interesado que esté involucrado en cuanto a las funciones que son útiles para la empresa.

### Tareas pendientes

#### Postulantes

Se investigará si es importante para la empresa perfeccionar el flujo de postulantes, en caso contrario convenimos con la cátedra que se harán los siguientes cambios para darle un pequeño cierre:

- Convertir postulante en empleado.
- Asociar postulante a búsqueda.
- Permitir marcar postulante como descartado.
- Limitar las búsquedas en el tiempo.

Otras funciones a considerar si la empresa lo considera útil:

- Separar el rol de edición de postulantes del de empleados de manera de permitir a otros empleados editar los postulantes sin acceder a información confidencial de los empleados.
- Indicar que tan reciente es la información de los postulantes.
- Indicar qué empleado realizó una entrevista.
- Indicar las entrevistas de idioma
- Sistematizar la carga de las entrevistas de manera que pueda ser útil para seleccionar postulantes.
- Medir efectividad de las búsquedas.
- Publicación de búsquedas en redes sociales. Que el que siga el link caiga en página para registrar sus datos como postulante, quede automáticamente asociado a esa búsqueda y se envíe notificación por mail a RRHH.

## Administración de los almuerzos diarios

Si bien no es una tarea puntual del área de RRHH, es una responsabilidad de esa area y ahora mismo no está funcionando bien. Considerando el tiempo invertido por cada empleado en mantener su menú actualizado y las confusiones que constantemente ocurren, creemos que sería bueno sistematizarlo. CommonJobs es la plataforma ideal para hacerlo. Eso además permitirá que el sistema sea utilizado más ampliamente en la empresa y que se proyecte como plataforma (uno de los objetivos originales).

## Empleados

Se agregará la matricula a los datos del empleado y en la página de legajo.

Continuaremos con el informe gráfico de las ausencias solicitado por el cliente.

Otras funciones a considerar:

- Bajas
- Períodos discontinuos de contratación
- Promociones/cambios de puesto
- Más de una universidad o varios títulos
- Cursos
- Nivel de estudios en personal sin estudios universitarios
- Edición de algunos datos por el propio empleado
- Pedido de vacaciones desde CommonJobs o integración con Jira

## Detalles de terminación

Es importante mejorar la estética y experiencia de usuario para causar una buena impresión a primera vista por lo que se realizarán mejoras en detalles de la UI y de usabilidad.

Considerar:

- Agregar nuevo postulante, no guardo pero queda grabado aunque no modifico ningún dato (dejo todo en blanco).
- Calendario debería tener una forma fácil de navegar entre años, porque se usa para fechas no muy recientes también, como el caso de experiencia previa.
- En experiencia previa el nombre del puesto del postulante podría estar separado del nombre de la empresa.
- No alerta que hay cambios sin grabar cuando se hace clic en un link que se abre en la misma página (ejemplo: clic en link de acceso junto a la estrella)
- Agrego Notas en un postulante sin llenar datos y grabo. Se graban aunque no tengan más datos que la fecha actual que se sugiere.
- Agrego Link de acceso, edito el texto y hago clic en el link y sale mensaje: "Sorry, an error occurred while processing your request." Si el link no existe hasta que se graba, todavía no debería verse como un link.

## Ayuda y documentación

Al sitio de documentación se agregará el manual de usuario, con una sección para cada página del sitio y para algunos conceptos como ser los "Slots".

También se incluirá la siguiente información de sistema:

- Descripción del dominio
- Descripción de la arquitectura
- Descripción de la “estructura” de base de datos (por ejemplo índices)

De ser posible se redactará un *Abstract* describiendo las características del sistema y resaltando aquellas que no se aprecian fácilmente a simple vista.

Considerar documentar:

- Críticas a la metodología de trabajo utilizada
- Pros y contras de RavenDB
- Plan de Implantación (incluyendo deployment, requerimientos de hardware, soporte y migración de datos).