



DEPARTAMENTO
DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

Trabajo práctico Nro. 2

Programming in the large

21 de noviembre de 2011

Ingeniería de Software II

Integrante	LU	Correo electrónico
Facundo Carillo	221/07	fedepousa@gmail.com
Rodrigo Castaño	221/07	fedepousa@gmail.com
Brian Curcio	61/07	bcurcio@gmail.com
Federico Pousa	221/07	fedepousa@gmail.com
Felipe Schargorosdsky	221/07	fedepousa@gmail.com



Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

Ciudad Universitaria - (Pabellón I/Planta Baja)

Intendente Güiraldes 2160 - C1428EGA

Ciudad Autónoma de Buenos Aires - Rep. Argentina

Tel/Fax: (54 11) 4576-3359

<http://www.fcen.uba.ar>

Índice

1. Plan de proyecto	3
1.1. Iteraciones	3
1.1.1. Gantt	3
1.2. Work Breakdown Structure	3
2. Escenarios	7
2.1. Disponibilidad	7
2.2. Performance	8
2.3. Seguridad	8
2.4. Modificabilidad	9
2.5. Usabilidad	9
3. Análisis de Riesgos	11
3.1. Riesgos	11
3.2. Planes de contingencia y mitigación	12

$$\begin{aligned}
y &= x^4 + 4 \\
&= (x^2 + 2)^2 - 4x^2 \\
&\leq (x^2 + 2)^2
\end{aligned} \tag{1}$$

1. Plan de proyecto

1.1. Iteraciones

La primera iteración comprenderá principalmente tareas de gestión y capacitación de los desarrolladores. Se tomó esta decisión porque identificamos varios intereses en conflicto y consideramos que es necesario establecer una prioridad entre ellos tan tempranamente como sea posible. Para hacer esto correctamente será necesario en primer lugar capacitar al equipo de desarrollo y de comunicación con los stakeholders en temas de seguridad. Priorizamos este tema en particular por haber sido considerado un atributo de calidad prioritario en el QAW y por la falta de capacitación del equipo en el mismo.

Una vez realizada la capacitación será necesario identificar arquitecturas alternativas priorizando de manera diferente los distintos intereses en conflicto.

Contando con distintas alternativas arquitectónicas se realizará una serie de reuniones con los distintos stakeholders informándoles por qué los distintos intereses entran en conflicto desde un punto de vista tecnológico y en qué medida beneficia cada posible decisión arquitectónica a la concreción de un objetivo u otro.

Es crucial que la totalidad del equipo tenga conocimientos suficientes de seguridad para garantizar que no sean introducidas vulnerabilidades en módulos aparentemente no relacionados con el tema, por lo que la capacitación de los desarrolladores continuará y se solapará con las reuniones con los stakeholders.

1.1.1. Gantt

1.2. Work Breakdown Structure

Utilizando la dinámica sugerida en las clases teóricas, generamos un WBS híbrido. En el primer nivel dividimos el sistema en 5 productos y procesos que capturan las distintas actividades a realizarse para la construcción del sistema completo. Cada uno de los productos y procesos del primer nivel agrupan funcionalidades relacionadas, excepto Actividades de gestión, que se diferencia de los demás por relacionarse con el sistema en su totalidad.

Esta división no muestra todas las dependencias entre productos y procesos, muchas de las cuales terminarán de definirse en el diagrama de Gantt de la iteración que corresponda.

Los cinco elementos del primer nivel serán Interfaz, Sistema Facultad, Comunicación, Actividades de gestión y Fiscalizador.

Interfaz comprende los productos y procesos relacionados con el relevamiento de requerimientos, casos de uso, codificación y prueba de la interfaz. Este producto seguramente podrá ser construido independientemente de los demás.

Sistema Facultad incluye productos y procesos orientados a la construcción del sistema autónomo de cada Facultad, que se encargará de recibir y fiscalizar la totalidad de los votos correspondientes a la misma. Los requerimientos funcionales y atributos de calidad prioritarios definitivos para este producto dependerán de varias tareas de gestión a realizarse en las primeras iteraciones, con el fin de consensuar soluciones de compromiso a los varios intereses que actualmente se encuentran en conflicto, como ser la posibilidad de contar con resultados parciales y un registro de votantes en tiempo real, en contraposición con la idea de garantizar el voto secreto.

Comunicación está conformado por productos y procesos con el fin de proveer de comunicación entre los distintos sistemas de cada Facultad. Este producto, tal como Sistema Facultad, depende de la definición de varios aspectos de los atributos de calidad y requerimientos funcionales, por lo que dependerá de los procesos de Actividades de gestión.

Actividades de gestión reúne diversos procesos que pueden ser separados en unos pocos ejes. En primer lugar está la capacitación de los stakeholders, explicando por qué los intereses expresados entran en conflicto y cuáles alternativas arquitectónicas permiten priorizar uno u otro y en qué medida. En segundo lugar habrán gran cantidad de procesos destinados a la validación intermedia del avance de la construcción del sistema, que consistirán en reuniones pequeñas con los principales interesados y algunas reuniones más grandes donde se muestren funcionalidades de interés general. También habrán actividades de planificación en general, que comprenderá, por ejemplo, la confección de los diagramas de Gantt de cada iteración. Por último será necesario capacitar al equipo en temas de seguridad, para lograr que sea tomada en cuenta en todos los niveles de desarrollo y elicitación de requerimientos. Esta capacitación será una de las primeras actividades a realizarse.

Ingresando al sistema.

Actor	Votante
Pre condición	Ninguna
Pos condición	El usuario se encuentra autenticado.
Curso normal	Curso alternativo
1. El votante indica su nombre de usuario y contraseña	
2. Si los datos son correctos el usuario recibe una confirmación de que su nombre de usuario y contraseña son correctos	Si los datos son incorrectos el usuario recibe un mensaje de error indicándole que la combinación de nombre de usuario y contraseña es incorrecta
3. Se muestra la interfaz de voto electrónico	Ir al fin del caso de uso
4. Fin del caso de uso	

Votando

Actor	Votante
Pre condición	El usuario se encuentra autenticado
Pos condición	Se registra correctamente el voto del usuario
Curso normal	Curso alternativo
1. Se muestra ante el usuario la lista de candidatos	
2. El usuario selecciona el candidato de su preferencia	
3. Se pide confirmación al usuario mostrando claramente qué candidato ha elegido	
4. El votante confirma su selección	Si el usuario no confirma, se descarta la selección del usuario
5. Se le informa al usuario que se ha enviado a su email el certificado del voto	Ir al fin del caso de uso
6. Fin del caso de uso	

Fiscalizando resultados parciales:

En este caso de uso se accede desde la terminal de una agrupación política a una interfaz gráfica que permite ver los resultados disponibles en tiempo real.

Oberservando resultados parciales

Actor	El usuario del rectorado
Pre condición	El usuario se encuentra autenticado
Pos condición	Se entregan resultados parciales

	Curso normal	Curso alternativo
1.	Se muestra opciones de resultados parciales por facultad con votaciones abiertas	
2.	El usuario selecciona el resultado parcial a observar	
3.	Fin del caso de uso	

2. Escenarios

Tras presenciar el QAW y releer su resumen decidimos plasmar todos los escenarios separados por categorías. Gracias a esto pudimos tomar decisiones arquitectónicas a fin, satisfaciendo los atributos de calidad relevados.

2.1. Disponibilidad

1.
 - *Es de esperar que haya problemas de conectividad, por lo que se espera que el sistema pueda funcionar también en modalidad offline. Bajo esta modalidad, los votos serán almacenados de manera temporal y segura en la sede correspondiente, para luego ser transmitidos al momento de recuperar la conexión. De la misma manera, esta modalidad deberá funcionar para la comunicación entre el rectorado y las facultades.*
 - **Fuente:** Interno al sistema.
 - **Estímulo:** Hay problemas de conectividad, se cayó la conexión entre el sistema y GUI-onlines
 - **Entorno:** Operación normal
 - **Artefacto:** GUI-offline
 - **Respuesta:** La GUI-offline detectan que el sistema está offline y habilitan la votación local en modalidad offline
 - **Medición:** La GUI-offline detecta que el sistema perdió conectividad y habilita votación offline en menos de 0.314159265358979323846264338327950 segundos
2.
 - *A fin de cumplir con el Estatuto Universitario, el Rectorado mantendrá la potestad de definir los requisitos que deben cumplir los electores, junto con el calendario de elecciones y otras normas que sirvan para preservar cierta igualdad entre unidades académicas. Asimismo, el Rectorado espera contar con información actualizada al instante de todos los procesos simultáneos que se desarrollen.*
 - **Fuente:** conexión con otra facultad/rectorado
 - **Estímulo:** error en la conexión
 - **Entorno:** Normal
 - **Artefacto:** Sistema
 - **Respuesta:** Utilizar una conexión alternativa
 - **Medición:** El sistema estará conectado nuevamente en menos de 15 segundos
3.
 - *Algunos no están convencidos de utilizar el voto offline y afirman que el rectorado debería garantizar conectividad*
 - **Fuente:** Votante
 - **Estímulo:** El usuario emite su voto exitosamente
 - **Entorno:** Normal
 - **Artefacto:** Sistema
 - **Respuesta:** El voto es registrado correctamente
 - **Medición:** Se garantiza una disponibilidad de 99,99999 %

2.2. Performance

1.
 - *Al finalizar el acto electoral el recuento de votos debera ser inmediato*
 - **Fuente:** Rectorado
 - **Estímulo:** Piden recuento de votos
 - **Entorno:** Normal
 - **Artefacto:** Sistemas en rectorado y sistemas en cada facultad
 - **Respuesta:** El sistema en rectorado le pide una copia de los votos a cada facultad y hace el recuento
 - **Medición:** El envío de los votos de cada facultad se hace en menos de 1.1235813213455 segundos y el recuento de votos total se hace en 2.71828182845904523 segundos
2.
 - *Debe publicarse en internet una copia de todo el código fuente, más la firma digital de la versión en uso, como así también debe ser posible que cada agrupación fiscalice el funcionamiento correcto de la aplicación por medio de la auditoría en tiempo real y ejecución paralela de copias del mismo software, corriendo en los equipos propios de cada fuerza política.*
 - **Fuente:** Usuario
 - **Estímulo:** Emite un voto
 - **Entorno:** Sobrecargado
 - **Artefacto:** Sistema
 - **Respuesta:** Se registra el voto y se envia a los fiscalizadores
 - **Medición:** En menos de 5 segundos los bunkers estan informados del voto
3.
 - *Los resultados deben recibirse al mismo tiempo que se van generando y para luego ser contrastados con el escrutinio definitivo*
 - **Fuente:** Votante
 - **Estímulo:** El usuario emite su voto exitosamente
 - **Entorno:** Normal
 - **Artefacto:** Sistema
 - **Respuesta:** El voto es registrado correctamente
 - **Medición:** En menos de 1 hora los resultados de votacion son recibidos por todas las agrupaciones políticas.

2.3. Seguridad

1.
 - *Debe bancarse procesos de auditorías y recuentos*
 - **Fuente:** Fuente: Pedido de cualquier persona
 - **Estímulo:** Requiere auditar los votos para recontarlos
 - **Entorno:** Normal
 - **Artefacto:** Sistema
 - **Respuesta:** Le entrega online o en rectorado, una copia en formato digital de los votos emitidos, todo el material estar certificado
 - **Medición:** Debe remitir información en menos de una semana
2.
 - *El sistema debe asegurar el derecho al voto anónimo*
 - **Fuente:** Atacante con acceso privilegiado al sistema

- **Estímulo:** El atacante intenta identificar a quién votó un elector en particular
- **Entorno:** Normal
- **Artefacto:** Sistema
- **Respuesta:** El sistema guarda registro de todos los accesos realizados por el atacante
- **Medición:** En ningún caso es posible, a partir de la información obtenida, asegurar que un votante en particular votó a un candidato en particular en un tiempo menor a 100 años con las capacidades de cómputo actuales.

2.4. Modificabilidad

1.
 - *Quieren usar el sistema también para otro tipo de elecciones, por ejemplo, plebiscitos*
 - **Fuente:** Rectorado
 - **Estímulo:** Se desea confeccionar un plebiscito
 - **Entorno:** Normal
 - **Artefacto:** Sistema
 - **Respuesta:** Se configura el sistema en para que pueda ofrecer participar en un plebiscito sin ninguna cambio negativo al sistema actual
 - **Medición:** Se invierte menos del 12,345678 de las horas empleadas para diseñar el sistema original
2.
 - *Que se pueda votar desde diversas plataformas, celulares, todo.*
 - **Fuente:** Un votante
 - **Estímulo:** Se intenta votar desde una Tablet conectada con 4G por internet.
 - **Entorno:** Normal
 - **Artefacto:** Sistema web
 - **Respuesta:** El sistema permite votar siendo transparente para este la plataforma.
 - **Medición:** El sistema funciona correctamente en los 5 dispositivos mas utilizados.

2.5. Usabilidad

1.
 - *Francisco quiere que el sistema sea sencillo de usar, para que no se confunda al electorado al momento de emitir el voto*
 - **Fuente:** Usuario
 - **Estímulo:** El usuario ingresa el voto para el candidato que eligió
 - **Entorno:** Operación normal
 - **Artefacto:** Sistema
 - **Respuesta:** Se registra el voto del usuario
 - **Medición:** El 99,99 % de los usuarios efectivamente votó al usuario que cree haber votado.
2.
 - *La interfaz debe ser simple, para que no consuma mucho tiempo del usuario*
 - **Fuente:** Usuario
 - **Estímulo:** El usuario ingresa el voto para el candidato que eligió
 - **Entorno:** Operación normal
 - **Artefacto:** Sistema

- **Respuesta:** Se registra el voto del usuario
 - **Medición:** El 99 % de los usuarios logran emitir su voto en dos minutos.
3. ▪ *El sistema debera brindar ayuda para aprender a ser usado, de manera que no sea necesario leer un manual*
- **Fuente:** Usuario que no sabe usar el sistema
 - **Estímulo:** El usuario quiere elegir un candidato
 - **Entorno:** Operación normal
 - **Artefacto:** Sistema
 - **Respuesta:** Se registra el voto del usuario
 - **Medición:** Todos los usuarios logran emitir un voto.
4. ▪ *Interfaz intuitiva, y clara! (usabilidad)*
- **Fuente:** Un votante
 - **Estímulo:** Se ingresa por primera vez al sistema intentando votar
 - **Entorno:** Normal
 - **Artefacto:** Sistema web
 - **Respuesta:** El sistema permite registrar la votación
 - **Medición:** El usuario puede registrarsu voto en menos de 4 minutos en la primera vez.
5. ▪ *Usable en otros idiomas!, fácil de configurar el idioma (modificabilidad, usabilidad)*
- **Fuente:** Un votante
 - **Estímulo:** Un votante que domina mal el español ingresa el sitio para votar y desea cambiar el idioma a inglés
 - **Entorno:** Operación Normal
 - **Artefacto:** Sistema
 - **Respuesta:** El sistema permite elegir el idioma y cambiar sus contendo al idioma correspondiente.
 - **Medición:** El usuaro encuentra como cambiar el idioma en menos de 3 segundos.
6. ▪ *Desea que se instalen terminales en todas las sedes para que pueda votar todos los alumnos sin conexión a internet.*
- **Fuente:** Un votante
 - **Estímulo:** Un votante sin conexión a internet se acerca a la facultad a emitir su voto en una termna
 - **Entorno:** Normal
 - **Artefacto:** Terminal de votación
 - **Respuesta:** La terminal registrar elvoto
 - **Medición:** El 100 % de los votantes pueden votar en las terminales de la facultad

3. Análisis de Riesgos

En esta sección se analizarán los riesgos pertinentes al proyecto.

El objetivo de este análisis es poder crear un plan de mitigación y contingencia para poder controlar los riesgos que puedan afectar al proceso del desarrollo del producto, de manera que se pueda minimizar el impacto que los posibles riesgos produzcan en caso de manifestarse.

Los pasos más importantes en este análisis son:

- Identificar los riesgos que puedan manifestarse.
- Analizarlos individualmente.
- Documentarlos. En este paso se utilizará la representación de Glutch para especificarlos y luego se utilizará la Matriz de Magnitudes del SEI para analizar el nivel del riesgo en base a su probabilidad de ocurrencia y a su severidad.
- Generar un plan de mitigación y contingencia para cada uno de los riesgos identificados.

3.1. Riesgos

A continuación se muestran los riesgos más relevantes identificados por el grupo, especificados y analizados como se explicitó anteriormente.

- Dificultades para funcionar co-operativamente con los sistemas de los partidos políticos: Dada la heterogeneidad natural en los sistemas de los partidos políticos, y que no hay experiencias previas de cooperatividad con estos sistemas, es posible que se encuentren dificultades para garantizar la comunicabilidad a todos los fiscalizadores pertinentes.
 - Probabilidad: Probable.
 - Severidad: Crítica.
 - Nivel: Alto
- Problemas con las tecnologías a utilizar: Dado los atributos de calidad relevados, se necesitará trabajar sobre muy diversas plataformas, al desconocer varias de estas plataformas de última generación es posible que se retrasen los tiempos de producción al necesitar invertir tiempo en investigación de los diferentes dispositivos soportados por el software.
 - Probabilidad: Muy probable.
 - Severidad: Media.
 - Nivel: Alto.
- Reducción de recursos humanos en la Software Factory: Dado que los recursos humanos del grupo de desarrollo estan sujetos a posibles renuncias, despidos o licencias por enfermedad o motivos personales es posible que el plan de trabajo no se pueda adaptar a la realidad al tener distintas horas hombre de trabajo a las planificadas.
 - Probabilidad: Poco probable.
 - Severidad: Crítica.
 - Nivel: Medio.
- Cambios en los requerimientos: Dado que la cantidad de Stakeholders es alta, es posible que algún Stakeholder cambie algún requerimiento durante fases más tardías del proceso productivo.

- Probabilidad: Probable.
 - Severidad: Crítica.
 - Nivel: Alto.
-
- Incompatibilidades entre los requerimientos no funcionales: Dado que la cantidad de Stakeholders es alta, y sobretodo es un grupo muy heterogeneo, esto genera que los atributos de calidad relevados tengan difentes procendencias y, luego, es posible que los atributos conflictuen entre sí de alguna forma no prevista durante el QAW.
 - Probabilidad: Probable.
 - Severidad: Media.
 - Nivel: Medio.

 - Dificultades con las capacidades de los recursos tecnológicos disponibles: Dado que no hay antecedentes de este tipo de sistemas en la UBA, no hay estimación de cuales van a ser las distribuciones de usuarios en los diferentes horarios. Luego, es posible que la tecnología disponible no sea suficiente para soporta la carga total del sistema.
 - Probabilidad: Probable.
 - Severidad: Crítica.
 - Nivel: Alto.

 - Robo de claves del sistema: Dado que el propósito del sistema tiene fines políticos relevantes, es posible que existan intentos de obtener las claves que permiten autentificarse a los votantes para poder manipular las votaciones.
 - Probabilidad: Poco probable.
 - Severidad: Crítica.
 - Nivel: Medio.

 - Recorte de presupuesto: Dado que el presupuesto para el sistema todavía no esta aprobado porque depende de todo el presupuesto de la UBA ya que lo paga directamente rectorado, es posible que se generen recortes al presupuesto que pongan en riesgos la disponibilidad de recursos humanos y tecnológicos para el proyecto.
 - Probabilidad: Probable.
 - Severidad: Crítica.
 - Nivel: Alto.

3.2. Planes de contingencia y mitigación

A continuación se presentarán las acciones a tener en cuenta en caso de que se presenten los riesgos anteriormente planteados.

- Dificultades para cooperar con los sistemas de los partidos políticos:
 - Priorizar el relevamiento de las especificaciones de los sistemas a tener en cuenta.
 - Mantener un contacto periódico con los administradores de dichos sistemas.
 - Generar reuniones con los administradores para poner en consenso el hecho de que los sistemas tengan una comunicación compatible.

- Problemas con las tecnologías a utilizar:
 - Relevar todas las diferentes plataformas que se quieren soportar en el sistema.
 - Relevar las formaciones de los recursos humanos en cada una de las plataformas pertinentes.
 - Generar un plan de capacitación en base a los relevamientos hechos anteriormente.
- Reducción de recursos humanos en la Software Factory:
 - Relevar situaciones personales antes de comenzar con el desarrollo.
 - Tener contacto de empresas de outsourcing competentes ante la reducción.
- Cambios en los requerimientos:
 - Tener un moderador de QAW capacitado para que el mismo sea exitoso y queden bien claros los requerimientos.
 - Generar reuniones periódicas con los Stakeholders para revisar que se esten satisfaciendo sus necesidades conforme avanza el proyecto.
- Incompatibilidades entre los requerimientos no funcionales:
 - Permitir que los arquitectos y los analistas formen parte del QAW para una detección temprana de incompatibilidades.
 - Una vez identificada una incompatibilidad, ponerla en evidencia a los Stakeholders pertinentes para generar una votación de prioridades.
 - Si la incompatibilidad se detecta en una etapa tardía actuar en consecuencia a la votación de prioridades realizada en el QAW.
- Dificultades con las capacidades de los recursos tecnológicos disponibles:
 - Relevar datos estadísticos de sistemas con características similares.
 - Generar test de stress en base a los datos relevados y en base a datos *pesimistas*, como podría ser todo el padrón de una facultad utilizando el sistema en la misma franja horario.
 - En caso de no tener la capacidad suficiente o de que se presenten problemas en el momento de una votación, generar un plan de división del padrón para sugerir horarios de utilización del sistema.
- Robo de claves del sistema:
 - Mantener una cantidad mínima de recursos humanos abocados a la tarea de generación de claves.
 - Tener un sistema de claves de emergencia en caso de que se detecte una filtración de información para asignar rápidamente las nuevas formas de autenticación.
- Recorte de presupuesto:
 -