

# **GVA**

**Lecciones Aprendidas  
Semana 12**

**Versión 1.5**

## Historia de revisiones

Fecha	Versión	Descripción	Autor
29/08/2014	1.0	Lecciones aprendidas en las primeras dos semanas de proyecto.	Todo el equipo.
27/09/2014	1.1	Lecciones aprendidas.	Todo el equipo.
24/10/2014	1.2	Agregado de lecciones aprendidas de verificación	Christopher Quincke
25/10/2014	1.3	Lecciones aprendidas de SCM y responsable de integración	Nicolás Greising
26/10/2014	1.4	Lecciones aprendidas de Diseño e Implementación	Martín Santagata
26/10/2014	1.5	Lecciones aprendidas SQA	Alejandro Casco

# Contenido

- [1. Introducción](#)
- [2. Por Disciplinas](#)
  - [2.1. Requerimientos](#)
  - [2.2. Diseño](#)
  - [2.3. Implementación](#)
  - [2.4. Verificación](#)
  - [2.5. Implantación](#)
  - [2.6. Gestión de Proyecto](#)
  - [2.7. Gestión de Configuración y Control de Cambios](#)
  - [2.8. Gestión de Calidad](#)
  - [2.9. Comunicación](#)
  - [2.10. Formación y Entrenamiento](#)
- [3. Otras lecciones](#)

# 1. Introducción

Lección Aprendida: Experiencia positiva o negativa obtenida durante la realización de alguna actividad. Se trata del registro de mejores prácticas, problemas recurrentes o experiencias exitosas, durante la implantación del proceso.

La siguiente lista de verificación es actualizada (agregando, modificando y/o eliminando items, cada 15 días en la reunión de equipo).

## 2. Por Disciplinas

### 2.1. Requerimientos

1. Siempre llevar una guía de preguntas.
2. Llevar la conversación hacia los objetivos.
3. Hacer sugerencias cuando no se tienen cosas definidas.
4. Cuando se le consulta algo al cliente y este habla demasiado sobre el tema, sin llegar a nada muy concreto, tratar de cortarlo cambiando el tema o quizás preguntando nuevamente.

### 2.2. Diseño

1. Tener un buen conocimiento del patrón MVC, el cual es un tanto diferente a lo aprendido previamente en la facultad.
2. Diseñar con ayuda de un especialista técnico para lograr un diseño más certero.
3. Diseñar la interfaz tempranamente ayudó a validar tanto pantallas como funcionalidades.
4. Un buen diseño de la estructura de la base de datos es fundamental para el desarrollo y la estabilidad del producto.

### 2.3. Implementación

1. Para poder llevar a cabo una integración exitosa de los casos de uso que van a ser de la liberación es importante tener un grupo bien definido de casos de uso. Una vez definido este grupo, si al momento de integrar, alguno de los casos de uso especificados no se encuentra terminado, se debe proceder a integrar dejando el mismo, en caso de ser posible, para una liberación posterior.
2. Ser cuidadoso al utilizar herramientas de versionado.
3. Con respecto a los métodos de trabajo se probó implementar de a pares, de forma individual y en grupo, siendo el último método el más efectivo pero difícil de llevar a cabo.
4. Conocer lo mejor el lenguaje a trabajar, en este caso C# y .Net, para así sacarle el máximo provecho de las funcionalidades que posee.

## **2.4. Verificación**

1. A la hora de realizar la verificación es importante que el ambiente de verificación sea independiente del de desarrollo y que no se realicen cambios en el ambiente de verificación mientras se está verificando.
2. Para diseñar los casos de prueba es necesario tener un conocimiento profundo de los requisitos, por lo tanto en la etapa de relevamiento de requisitos la participación de algún integrante del equipo de verificación resulta muy apropiada y puede generar que la verificación sea de mejor calidad repercutiendo directamente en la calidad del producto final.
3. El ambiente de verificación de performance debe ser un ambiente controlado, donde no intervengan agentes externos que puedan alterar los resultados de las pruebas.
4. Se debe controlar que las pruebas unitarias tengan una calidad apropiada, de otra forma puede ocurrir que se encuentren errores en la etapa de verificación del sistema que debieron ser detectados en las pruebas unitarias.
5. Es de suma importancia definir en conjunto con el cliente criterios de aceptación que especifiquen lo más claro posible los requisitos funcionales y no funcionales que debe tener el producto, así como también los atributos de calidad deseados. De esta forma se evitan problemas de “malos entendidos” que puedan surgir al entregar el producto al cliente y se cuenta con un documento que facilita y respalda la decisión de si el producto se encuentra apto o no para ser entregado al cliente.

## **2.5. Implantación**

1. Manejar fechas más holgadas con el cliente que con el equipo.

## **2.6. Gestión de Proyecto**

1. Plantear objetivo de la fase, las metas que se deben alcanzar, y cómo se logrará.
2. Plantear viabilidad, identificando y mitigando los riesgos. Por ejemplo: la realización de prototipos para el estudio de la tecnología como meta a alcanzar.
3. Realizar planes que aseguren la calidad del producto.
4. Darle al cliente todo lo que pida, siempre y cuando
5. Fijar objetivos a corto plazo a los implementadores para así identificar los retrasos lo antes posible.
6. [Estimar las actividades de ... con una holgura de ...]

## **2.7. Gestión de Configuración y Control de Cambios**

1. Para poder llevar un buen seguimiento de la línea base es importante verificar semanalmente que el contenido del documento “Registro de versiones” esté coherente con la línea base del proyecto en ese momento.
2. Al momento de documentar la gestión de cambio, es importante que el formulario de solicitud de cambio tenga la información necesaria para llevar a cabo un buen registro de lo que se solicita.

## **2.8. Gestión de Calidad**

1. Mantener de forma organizada los documentos que se entregan y los que no
2. Para revisar los casos de prueba, chequear la coherencia de los mismos con el documento de modelo de casos de uso.
3. Definición de criterios de aceptación con el cliente.

## **2.9. Comunicación**

1. Para las reuniones tanto con el cliente como con el docente, se debe establecer de antemano un responsable de realizar el acta. Esto permite que la misma sea entregada en un plazo corto de a lo sumo 48 horas, evitando así omisiones en los temas tratados.
2. Antes de cualquier reunión, enviar a los convocados los temas a tratar para establecer un orden y lograr un mejor aprovechamiento del tiempo.
3. Nos resultó de mucha utilidad el uso de herramientas rápidas, como ser whatsapp que permite la comunicación constante e instantánea con todo el equipo.

## **2.10. Formación y Entrenamiento**

1. Es una buena medida para la transmisión de conocimientos la programación de a pares, principalmente en el comienzo del proyecto.