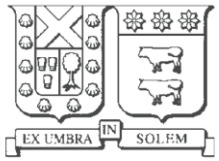
Departamento de Informática
Universidad Técnica Federico Santa María



Entregable III
Análisis y Diseño de Software / Fundamentos de Ingeniería de Software

Integrantes:

Nombre	Email	Teléfono
Francisco García M	francisco.garciam.13@sansano.usm.cl	+56962585319
Simón Contreras M	simon.contreras.13@sansano.usm.cl	+56984297665

Entregable 3

1- Listado de requerimientos

Requerimientos Funcionales:

Id	Requerimiento	Obligatoriedad
FR1	Los alumnos deben responder una encuesta para poder acceder a la información de la plataforma.	Obligatorio
FR2	La encuesta debe clasificar al alumno en uno de los cuatro tipos de estudiante definidos por D.Kolb.	Obligatorio
FR3	Los estudiantes deben tener acceso a la información de cada asignatura en el orden correspondiente a su clasificación.	Obligatorio
FR4	Los estudiantes deben poder inscribir asignaturas.	Obligatorio
FR5	Los alumnos NO deben conocer la clasificación que entrega la encuesta.	Obligatorio
FR6	Los alumnos deben poder inscribir asignaturas para poder acceder a la información de estas.	Obligatorio
FR7	Los alumnos deben poder cambiar el orden de la información que se muestra en pantalla y esta debe ser guardada.	Obligatorio
FR8	Los alumnos deben poder inscribir asignaturas para poder acceder a la información de estas, también las pueden desinscribir.	Obligatorio
FR9	Los usuarios deben poder editar su información personal y contraseña. (correo de acceso NO).	Obligatorio
FR10	Los usuarios deben tener un perfil de donde puedan interactuar con todas las funcionalidades de la plataforma.	Obligatorio
FR11	Debe existir un sistema de feedback, para poder reportar inconsistencias en la información y la organización de esta.	Obligatorio
FR12	Debe existir un canal de comunicación entre los usuarios	Obligatorio

	y el soporte de esta.	
FR13	La plataforma debe poseer un sistema de autentificación (Log-in).	Obligatorio
FR14	La plataforma debe proporcionar al administrador de facultades para crear, actualizar, visualizar y eliminar (CRUD) todo tipo de contenido (cuentas, información, etc.)	Obligatorio
FR15	Los profesores a cargo de cada asignatura deben poder subir material en distintos formatos (video, imágenes, gráficos, texto.)	Obligatorio
FR16	Los profesores deben poder crear módulos para cada unidad de aprendizaje, dentro de cada asignatura.	Obligatorio
FR17	Debería existir un tipo de evaluación dentro de la plataforma, para poder acceder al siguiente módulo de aprendizaje (definir una taxonomía sería lo óptimo).	Deseable
FR18	Los alumnos deberían poder agregar comentarios a los textos de información que estén estudiando y estos deben ser guardados.	Deseable
Nuevos		
FR19	Los alumnos deben poder ver el progreso de su encuesta en tiempo real.	Obligatorio
FR20	Debe existir una persona que se encargue de problemas técnicos de la plataforma.	Obligatorio

Requerimientos No Funcionales:

Id	**Requerimiento no funcional**	**FRs asociados**
NFR1	Encriptar contraseñas para el inicio de sesión en la plataforma.	FR9, FR13
NFR2	El desarrollo back-end y front-end deben ser con NodeJS y AngularJS, respectivamente. Y motor de base datos a usar es MySQL.	Todos

NFR3	La plataforma debe soportar grandes volúmenes de usuarios conectados simultáneamente.	Todos
NFR4	Los usuarios deben solicitar al administrador de la página la creación de sus cuentas.	FR13

2- Casos de uso

El diagrama de casos de uso final se encuentra en el archivo modelosEntregable.vpp en la carpeta modelos.

3- Diagramas de secuencia del sistema

Ver archivo modelosEntregable.vpp en carpeta modelos.

Observación:
-Todos los casos de uso parten de la premisa que la persona usando el sistema está en la página principal de su cuenta.
- Si en el diagrama no hay clases instanciadas significa que fueron instanciadas en un paso pre-requisito del caso de uso en cuestión. Generalmente al iniciar sesión se genera la instancia del usuario por lo que no es necesario que vuelva a aparecer en el diagrama.
-Por último, como se considera al sistema como caja negra, sólo interesa la respuesta directa que recibe el usuario y no como el sistema realiza la petición.

4- Modelo relacional de la base de datos

Ver archivo modelosEntregable.vpp en carpeta modelos.

5- Modelo de clases

Ver archivo modelosEntregable.vpp en carpeta modelos.

6- Diagrama de secuencia de componentes del sistema

Ver archivo modelosEntregable.vpp en carpeta modelos.

Observaciones:
- Los controladores de Controlador.js sólo controlan el despliegue de información de las Views, por lo tanto, no interaccionan directamente con las clases del modelo, por lo cual no serán incluidas en los diagramas.

7- Bosquejo MVC

Ver archivo modelosEntregable.vpp en carpeta modelos.

Patrones Utilizados:

Page Controller: este patrón fue escogido ya que permite encapsular en una sola clase la lógica del control y flujo entre las distintas Views de la plataforma. El patrón está implementado en el archivo routes.js, este controla, invoca y determina que vista se debe desplegar bajo cierta acción del usuario.

Transaction Script: este patrón fue escogido debido a que busca encapsular la lógica de negocio y la conexión con la base de datos en clases según el contexto de la operación. Este se ve reflejado en los controladores de la plataforma los cuales poseen ciertos procedimientos específicos los cuales son parte de la lógica del negocio, por lo cual con Transaction script se puede encapsular de manera eficiente este tipo de transacciones, además todas las clases del modelo de datos poseen ciertas operaciones encapsuladas bajo la misma lógica que brinda este patrón.

Active Record: este patrón fue escogido debido a que las operaciones más comunes a realizar sobre la base de datos son operación CRUD, por lo tanto, es evidente el agregar las operaciones insert, update, create, delete, etc dentro las clases del modelo de datos. Como se acaba de indicar este patrón es visible en las operaciones implementadas en cada clase del modelo de datos.

Singleton: el poder asegurar la integridad de la sesión del usuario y de las posibles transacciones desde una cuenta hacia la base de datos es esencial, por lo que mediante el uso del middleware Passport para NodeJS, se implementa este patrón, lo cual permite mantener una instancia por usuario dentro de la plataforma.

Template View: como se escogió usar EJS para las Views, es evidente que este patrón está presente, ya que EJS permite el uso de Javascript sobre la plantilla de forma nativa, lo que permite marcar y desplegar información de manera mucho más eficiente, además el uso de AngularJS y su data binding ({{data}}), expresa fielmente el

patrón de diseño Template View, en donde mediante marcas se despliega la información dinámica.

8-Lecciones Aprendidas

El poder integrar tecnologías distintas (Nodejs, Angularjs, Mysql) facilita enormemente el desarrollo de plataformas en cortos períodos de tiempo.

El uso de patrones de diseño es primordial a la hora de diseñar plataformas, en general las soluciones que se desean implementar son problemas recurrentes en el desarrollo de software, por lo que seguir los patrones estándar de las industrias asegura obtener un software funcional, el cual es más fácil de entender, debuggear y modificar.

La documentación es primordial a la hora de explicar el sistema otro informático, ya que la representación gráfica de los sistemas en interacción, usando estándares internacionales (UML) facilita la comprensión de este, y una posterior implementación.

Las herramientas de control de versiones y kanban, permiten gestionar de buena manera el trabajo en equipo.

Futuras mejoras:

Mayor uso del Issue Tracker y un uso más exhaustivo del control de versiones (GitHub).

Mejor organización tanto en las carpetas como los archivos del código de la plataforma.

Mejor encapsulamiento de las reglas del negocio críticas.

Mejorar los perfiles de los accesos a la base de datos, abusar más de las propias funcionalidades del gestor de base de datos.

Errores cometidos:

Trabajar en la plataforma muy encima de la fecha de entrega de los entregables.

No conversar lo suficiente con el cliente, si se hubiera hecho un feedback más constante el producto final hubiera sido aún más atractivo para él.

Entregable 3