



Departamento de Informática
Universidad Técnica Federico Santa María



Entregable III
Análisis y Diseño de Software
120-COGNITIO
NONAME CORP.

Integrantes:

Nombre	Email	Teléfono
Eduardo Borgoño	eduardo.borgono.13@sansano.usm.cl	+56 9 5125 5845
Juan Pablo Muñoz	Juan.munozm.13@sansano.usm.cl	+56 9 8677 6990
Francisca Ramírez	Francisca.ramirez.12@sansano.usm.cl	+56 9 8678 4013

1- Listado de requerimientos

REQUERIMIENTOS FUNCIONALES:

ID requerimiento	Requerimiento	Obligatoriedad
FR000	El estudiante debe poder registrarse.	Obligatorio
FR001	El estudiante debe poder autenticarse.	Obligatorio
FR002	El estudiante debe poder responder u omitir el test.	Obligatorio
FR003	El sistema debe asignar un perfil a cada estudiante que responda el test o mostrar una vista estándar si un estudiante omite el test.	Obligatorio
FR004	El estudiante debe poder elegir las unidades y tópicos disponibles.	Obligatorio
FR005	El sistema debe mostrar los contenidos disponibles en los tópicos según el tipo de estudiante con el que interactúe.	Obligatorio
FR007	El estudiante debe poder entregar un feedback del contenido en visualización.	Obligatorio
FR008	El sistema debe mostrar al coordinador o profesor los feedbacks existen.	Deseable
FR010	El profesor debe poder autenticarse.	Obligatorio
FR011	El profesor debe poder ver, añadir, eliminar y editar una unidad.	Obligatorio
FR012	El profesor debe poder ver, añadir, eliminar y editar un tópico.	Obligatorio
FR013	El profesor debe poder subir, ver, eliminar y editar un contenido.	Obligatorio
FR014	El coordinador debe poder autenticarse.	Obligatorio
FR015	El sistema mostrará un panel de control de contenidos y usuarios.	Obligatorio
FR016	El coordinador debe poder ver, añadir, eliminar y editar una unidad.	Obligatorio
FR017	El coordinador debe poder ver, añadir, eliminar y editar un tópico.	Obligatorio
FR018	El coordinador debe poder subir, ver, eliminar y editar un contenido.	Obligatorio

FR019	El coordinador debe poder crear una cuenta para un profesor, coordinador.	Obligatorio
FR020	El coordinador debe poder suspender la cuenta de un estudiante, profesor o coordinador.	Deseable

REQUERIMIENTOS NO FUNCIONALES:

ID NF requerimiento	Requerimiento no funcional	FR asociados
NFR1	La autenticación estará restringida a usuarios definidos por correos institucionales y contraseñas.	FR001, FR020, FR032
NFR2	El sistema presentará los contenidos a los estudiantes organizados por unidades, que contendrán determinados tópicos y estos a su vez contendrán determinados contenidos.	FR006, FR007, FR008, FR022, FR023, FR024, FR025, FR026, FR027, FR033, FR034, FR035, FR036, FR038, FR039, FR043, FR044
NFR3	El sistema manejará contenidos a través de links, imágenes y documentos en general, los cuales estarán limitados a un tamaño determinado.	FR008, FR009, FR010, FR028, FR029, FR030, FR040, FR041, FR042, FR045
NFR4	El sistema aceptará un único feedback por contenido y podrá ser modificado por un tiempo determinado.	FR011, FR012

2- Casos de uso

El diagrama de casos de uso se encuentra en el archivo de modelo de Visual Paradigm CE. Diagramas.vpp (ver github).

3- Diagramas de secuencia del sistema

Los diagramas de secuencia se encuentran en el archivo de modelo de Visual Paradigm CE. Diagramas.vpp (ver github).

4- Modelo relacional de la base de datos

El modelo relacional de la base de datos se encuentra en el archivo de modelo de Visual Paradigm CE. Diagramas.vp (ver github).

5- Modelo de clases

El modelo de clases se encuentra en el archivo de modelo de Visual Paradigm CE. Diagramas.vp (ver github).

6- Diagrama de secuencia de componentes del sistema

El diagrama de secuencias de componentes del sistema se encuentra en el archivo de modelo de Visual Paradigm CE. Diagramas.vp (ver github).

7- Bosquejo MVC

El modelo de clases MVC se encuentra en el archivo de modelo de Visual Paradigm CE. Diagramas.vp (ver github).

8- Lecciones aprendidas

Como grupo aprendimos a usar las herramientas de trabajo (Github y Trello). Trello nos ayudó en la organización para asignar tareas, evitando así que más de una persona trabajase en la misma tarea; por otra parte Github facilitó mucho el desarrollo colaborativo, permitiéndonos juntar los avances de código de cada uno en uno solo de forma automática.

Trabajar con framework (AngularJs) y patrones de diseño (principalmente MVC) facilitó el desarrollo de la aplicación web. AngularJs nos permitió un desarrollo mas dinámico de la página y los patrones de diseño por su parte nos ayudaron a comprender y solucionar problemas ya conocidos en el desarrollo de software y en el diseño de éste. El uso inicial de estas herramientas y también de NodeJS fue complejo ya que la documentación existente era de difícil comprensión, al ser herramientas desconocidas para nosotros también nos fue difícil saber buscar adecuadamente información al respecto.

Para proyectos futuros creemos que es importante organizar mejor el tiempo para desarrollar en equipo, ya sea fijando reuniones u horarios de trabajos en equipo, sin embargo fue difícil organizarnos con tiempo debido a la carga académica de otros ramos y a que las pautas no se entregaban con el suficiente tiempo.