

Especificación de Requerimientos

Descripción del Diseño

Education Software

Software facilitador para el manejo de notas académicas

Apellidos, Nombres	Correo electrónico	Rol
Camacho, Diana Marcela	dmcamachom@outlook.com	Tester
Henao Cano, Jefferson	jefferhenao911@gmail.com	Database Manager
Henao García, Ana Maria	anamariahenaogarcia@gmail.com	Backend Developer
Portilla Araujo, Keneth Danilo	kenethportilla@gmail.com	Project Manager
Sierra Calderón, Yesica Katherine	sierracyesica@gmail.com	Frontend Developer

Fecha de presentación: XX/XX/2021

Contenido

1	INTRODUCCIÓN	3
1.1	PROPÓSITO	3
1.2	ALCANCE O ÁMBITO DEL SISTEMA	3
1.3	DEFINICIONES, ACRÓNIMOS Y ABREVIATURAS	3
1.3.1	<i>Definiciones</i>	3
1.3.2	<i>Acrónimos</i>	3
1.3.3	<i>Abreviaturas</i>	3
1.3.4	<i>Referencias</i>	4
1.4	PERSPECTIVA GENERAL DEL DOCUMENTO	4
2	DESCRIPCIÓN GENERAL DE LA APLICACIÓN	4
2.1	PERSPECTIVA DE LA APLICACIÓN	4
2.2	FUNCIONES DE LA APLICACIÓN	4
2.3	CARACTERÍSTICAS DE LOS USUARIOS	5
2.4	RESTRICCIONES	5
2.5	SUPOSICIONES Y DEPENDENCIAS	5
2.6	REQUERIMIENTOS DIFERIDOS	5
3	REQUERIMIENTOS ESPECÍFICOS	5
3.1	REQUERIMIENTOS	6
3.1.1	<i>Product Backlog</i>	6
3.1.2	<i>Ciclo de Sprints del proyecto</i>	6
3.1.3	<i>Sprint Backlog</i>	6
3.1.4	<i>Historias de usuario (Tareas y Subtareas)</i>	6
3.1.5	<i>Mecánica de organización del grupo. (Reuniones, evidencias/artefactos)</i>	6
3.2	MODELO DE REQUERIMIENTOS	7
3.2.1	<i>Modelo de Casos de Uso</i>	7
4	DESCRIPCIÓN DEL DISEÑO	9
4.1	INTERFAZ GRÁFICA (MOCKUPS)	9
5	GESTIÓN DE LA CONFIGURACIÓN	9
6	PRUEBAS	9
6.1	DESCRIPCIÓN DE PRUEBAS UNITARIAS	9
6.2	DESCRIPCIÓN DE PRUEBAS DE ACEPTACIÓN	9
7	GLOSARIO	10
8	ANEXO(S)	10

1 INTRODUCCIÓN

[La introducción debe proporcionar una perspectiva de todo el documento. Debe contener los apartados que se indican a continuación.]

1.1 Propósito

- a. En este documento se encuentra la descripción del proceso y las herramientas utilizadas para el desarrollo del aplicativo web Education Software, como también información específica de uso, la población objeto y las metas de futuras actualizaciones.
- b. Este documento le proporciona información para sacar el mayor provecho posible del aplicativo y/o la metodología y procesos que llevó al primer prototipo y las proyecciones que se tienen para el aplicativo.

1.2 Alcance o Ámbito del Sistema

- a. Education Software es un aplicativo web.
- b. El aplicativo Education Software tiene como funciones almacenar, modificar y mostrar notas propias y personales de docentes vinculados a entidades académicas haciendo el cálculo requerido por la/el usuaria, facilitando así el manejo de un amplio número de estudiantes y disminuyendo significativamente el tiempo empleado en dichas tareas; actualmente está diseñado para docentes, aunque se espera en las próximas actualizaciones ofrecer también el rol de estudiante.
- c. El aplicativo Education Software posee varios tipos de visualización de las notas para la comodidad y necesidad del docente: por curso, por materia y por estudiante, el docente puede asignar el porcentaje de la nota que está suministrando al sistema y a su vez subdividir el total de ese porcentaje según la necesidad, ejemplo ese 100% de notas representa el 80% total esperando la nota del 20% de la evaluación por semestre y calcular así la nota final real. Para las primeras actualizaciones se tiene contemplado incluir el rol de estudiante dentro de la interfaz de inicio, el autocompletado de notas según casillas seleccionadas por el usuario, resaltado de los estudiantes que se encuentran en riesgo de perder el ciclo en la vista de curso.

1.3 Definiciones, Acrónimos y Abreviaturas

[Este apartado debe proporcionar las definiciones de términos, siglas y abreviaturas necesarios para interpretar adecuadamente el documento. Esta información puede proporcionarse por referencia a uno o más anexos de este documento o por referencia a otros documentos.]

1.3.1 Definiciones

1.3.2 Acrónimos

1.3.3 Abreviaturas

1.3.4 Referencias

- Las secciones y apartados que conforman la estructura de esta plantilla se adaptaron de los estándares: IEEE Std-830-1998 (IEEE Recommended Practice for Software Requirements Specifications); IEEE Std-1016-2009 (IEEE Standard for Information Technology — Systems Design — Software Design Descriptions); ISO/IEC/IEEE 29148:2011(E) (Systems and software engineering — Life cycle processes — Requirements engineering).
- Además, se consideraron los estándares: ISO/IEC 23026 (IEEE Std 2001-2002) (Software Engineering — Recommended Practice for the Internet — Web Site Engineering, Web Site Management, and Web Site Life Cycle); IEEE/EIA 12207.1-1997 (ISO/IEC 12207) (Standard for Information Technology—Software life cycle processes—Life cycle data).
- Se tuvieron en cuenta también los estándares: OMG Unified Modeling Language (OMG UML), Infrastructure, Version 2.4.1 (OMG Document Number: formal/2011-08-05; Standard document URL: <http://www.omg.org/spec/UML/2.4.1/Infrastructure>) y OMG Unified Modeling Language (OMG UML), Superstructure, Version 2.4.1 (OMG Document Number: formal/2011-08-06; Standard document URL: <http://www.omg.org/spec/UML/2.4.1/Superstructure>).

1.4 Perspectiva General del Documento

[Este apartado debe:

- a) describir lo que contiene el resto del documento;
- b) explicar cómo está organizado el documento.]

2 DESCRIPCIÓN GENERAL DE LA APLICACIÓN

[Esta sección del documento debe describir los factores generales que afectan a la aplicación y sus requerimientos. Esta sección no establece requerimientos específicos, los que se describen en detalle en la sección 3, sino que proporciona un contexto para dichos requerimientos.

Generalmente, esta sección consiste en seis apartados, que son los que siguen.]

2.1 Perspectiva de la Aplicación

[Este apartado debe poner la aplicación en perspectiva con otros productos relacionados. Si el producto es totalmente autónomo e independiente, se debe indicar aquí. Si se define una aplicación que es un componente de un sistema más grande, como ocurre con frecuencia, entonces en este inciso se debe relacionar los requerimientos del sistema más grande a la funcionalidad de la aplicación y debe identificar las interfaces entre ese sistema y la aplicación. Puede ser útil un diagrama de bloques que muestre los principales componentes del sistema más grande, las interconexiones y las interfaces externas.

Si son necesarios más detalles, recurrir al IEEE Std-830-1998.]

2.2 Funciones de la Aplicación

[Este apartado debe proporcionar un resumen de las principales funciones que ejecutará la aplicación, sin indicar la gran cantidad de detalles que pueda requerir cada una de esas funciones.

A veces el resumen de funciones que se necesita para este apartado puede tomarse directamente de una especificación de nivel superior (si existe) que asigna funciones particulares para la aplicación. Téngase en cuenta que por motivos de claridad

a) Las funciones deben organizarse de una manera que haga que la lista de funciones sea comprensible para el cliente o para cualquier otra persona que lea el documento por primera vez.

b) Pueden utilizarse formas textuales o gráficas para mostrar las diferentes funciones y sus relaciones. Con un esquema tal, no se pretende mostrar el diseño de un producto, sino simplemente las relaciones lógicas entre las funciones.]

2.3 Características de los Usuarios

[Este apartado debe describir las características generales de los usuarios previstos de la aplicación incluyendo el nivel educativo, experiencia y conocimientos técnicos. No debe utilizarse para establecer requerimientos específicos.]

2.4 Restricciones

[Este apartado debe proporcionar una descripción general de cualquier otra cuestión que limite las opciones del desarrollador. Se podría incluir: a) políticas regulatorias; b) limitaciones de hardware (por ejemplo, requerimientos de sincronización de señales); c) interfaces a otras aplicaciones; d) operación paralela; e) funciones de auditoría; f) funciones de control; g) requerimientos de lenguajes de alto nivel; h) protocolos; i) requerimientos de confiabilidad; j) criticidad de la aplicación; k) consideraciones de seguridad.]

2.5 Suposiciones y Dependencias

[Este apartado debe enumerar cada uno de los factores que afectan los requerimientos establecidos. Estos factores no son restricciones de diseño sino que, por el contrario, cualquier cambio en ellos que podría afectar los requerimientos. Por ejemplo, una hipótesis puede ser que un sistema operativo específico estará disponible para la aplicación. Si, de hecho, el sistema operativo no está disponible, el documento tendrá que cambiar en consecuencia.]

2.6 Requerimientos Diferidos

[Este apartado debe identificar los requerimientos que pueden postergarse para versiones futuras de la aplicación.]

3 REQUERIMIENTOS ESPECÍFICOS

[Para el desarrollo de toda esta sección 3 utilizar como bibliografía de soporte:

(1) Schwinger, W.; Koch, N. "Modeling Web Applications", Chapter 3 en: Kappel, G.; Pröll, B.; Reich, S.; Retschitzegger, W. (Editors) *Web Engineering. The Discipline of Systematic Development of Web Applications*, John Wiley & Sons Ltd., 2006.

(2) Koch, N.; Knapp, A.; Zhang, G.; Baumeister, H. "UML-Based Web Engineering. An Approach Based on Standards", Chapter 7 en: Rossi, G.; Pastor, O.; Schwabe, D.; Olsina, L. (Editors) *Web Engineering: Modelling and Implementing Web Applications*, Springer-Verlag London Limited, 2008.]

3.1 Requerimientos

[Este apartado debe contener una lista de los requerimientos surgidos de las necesidades e ideas aportadas por los usuarios/clientes, desarrolladores y demás participantes en el proceso de desarrollo.]

3.1.1 Product Backlog

[Lista con todos los requerimientos iniciales del producto que se va a desarrollar que identifique las necesidades del producto para lograr su máxima utilidad. Asimismo, contiene la descripción de las tareas y subtareas que se van a realizar para la ejecución de cada requisito, mismas que se organizarán en función de sus prioridades. Además, la pila de producto también indica una estimación del tiempo en la que cada tarea se va a desarrollar y el valor que cada una le da al producto.]

3.1.2 Ciclo de Sprints del proyecto

[Listar los sprints contemplados desde el proyecto para la generación de valor al cliente]

3.1.3 Sprint Backlog

[Lista de elementos seleccionados previamente del Product Backlog para ser desarrollados en el día a día en los diferentes Sprints del proyecto. Tras crear esta lista, el equipo del proyecto tendrá que identificar las funcionalidades y priorizar las que se entregarán en el Sprint.]

3.1.4 Historias de usuario (Tareas y Subtareas)

[Es una explicación general e informal de una función de software escrita desde la perspectiva del usuario final. Su propósito es articular cómo proporcionará una función de software valor al cliente. Asimismo, deben utilizar un lenguaje no técnico ya que supone la descripción de una función de software descrita por los usuarios finales].

3.1.5 Mecánica de organización del grupo. (Reuniones, evidencias/artefactos)

3.2 Modelo de Requerimientos

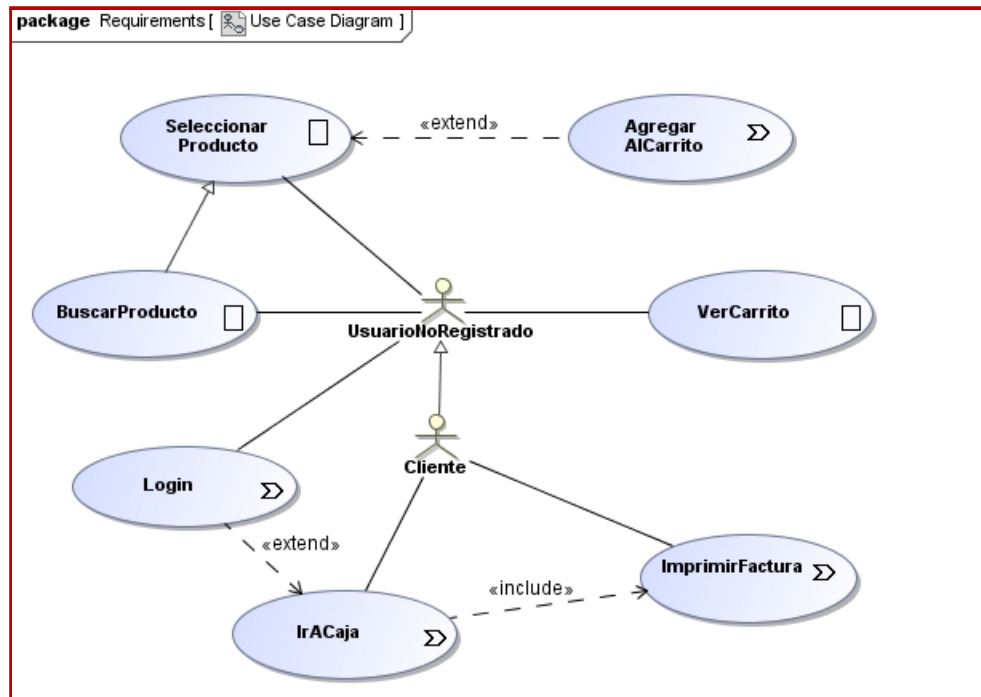
[El *modelo de requerimientos* se basa en dos modelos principales: el *modelo de casos de uso* y el *modelo del dominio*.]

3.2.1 Modelo de Casos de Uso

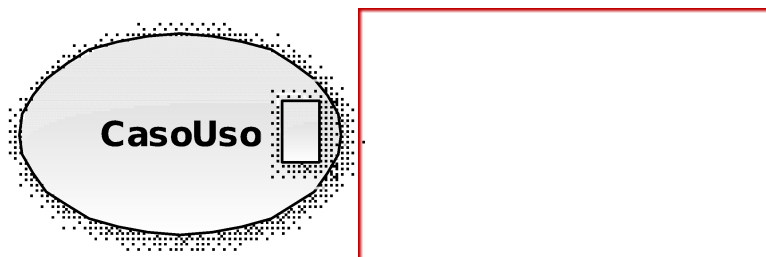
[La funcionalidad de una aplicación Web se modela como un conjunto de *casos de uso*, que describen los requerimientos de la aplicación desde las perspectivas de los *actores* (personas y otros sistemas).

Una particularidad de los requerimientos de una aplicación Web es la funcionalidad de navegación, que permite al usuario navegar por el hipertexto y encontrar nodos. El enfoque de UWE es crear un único modelo de casos de uso, que utiliza el estereotipo <<navigation>> para denotar la diferencia entre casos de uso funcionales y casos de uso específicos de hipertexto.

OBSERVACIÓN: Si la cantidad de casos de uso en un mismo diagrama es grande (una cantidad adecuada podría ser 7 ± 2), sería conveniente aplicar el antiguo principio de "divide et impera". Es decir, dividir el diagrama de casos de uso en varias partes agrupando elementos que tengan algún tipo de coincidencia entre sí. El elemento de UML que se utiliza para estas situaciones es el *paquete*. Si todavía algún paquete tuviera demasiados casos de uso, entonces se podría subdividir ese paquete en varios paquetes, y dentro de estos estarían los casos de uso. Se debe asegurar que los paquetes sean cohesivos, es decir, que manejen información o funcionalidad relacionada. Luego, a partir del diagrama de paquetes se derivan los casos de uso; cada paquete implica un diagrama de casos de uso (o de paquetes).]



[Para el estereotipo <<navigation>>, en lugar de la etiqueta se puede usar el ícono (€).]



[A continuación, se describen los casos de uso utilizando tablas y diagramas.]

CU-01: "Nombre del Caso de Uso"

[Se debe realizar una descripción de los escenarios del caso de uso.]

Iniciador	Actor que inicia el caso de uso.
Otros actores	Otros actores, si existen.

Precondiciones	Condiciones que deben cumplirse para que pueda realizarse el caso de uso.	
Flujo básico		
Actor	Sistema	
1. ...		
	2. ...	
	3. ...	
4. ...		
Flujo alternativo 1	Indicar otro camino por el cual se puede realizar el caso de uso, aparte del flujo básico	
Flujo alternativo 2		
...		
Flujo alternativo n		
Poscondiciones	Indicar los cambios que se producirán o los datos que deben quedar registrados.	

[Cuando se ejecuta una instancia de un caso de uso, éste interactúa con instancias de actores y realiza una secuencia de acciones de acuerdo a lo que se describió en la especificación del caso de uso. Para mostrar la realización de un caso de uso se puede emplear: un *diagrama de clases* con todas las clases que participan en él; *diagramas de interacción* (secuencia y/o comunicación); o, como indica UWE, *diagramas de actividad*.

OBSERVACIÓN: UWE sugiere lo siguiente "Siguiendo el principio de utilizar UML siempre que sea posible para la especificación, se refinan los requerimientos con diagramas de actividad UML. Para cada caso de uso no trivial, se construye al menos un diagrama de actividad para el flujo principal de tareas a realizar para proporcionar la funcionalidad indicada por el caso de uso correspondiente. Opcionalmente, pueden utilizarse diagramas adicionales para excepciones y variantes."]

CU-02: "Nombre del Caso de Uso"

...

...

CU-nn: "Nombre del Caso de Uso"

4 DESCRIPCIÓN DEL DISEÑO

[Para el desarrollo de toda esta sección 4 utilizar como bibliografía de soporte:

(1) Schwinger, W.; Koch, N. "Modeling Web Applications", Chapter 3 en: Kappel, G.; Pröll, B.; Reich, S.; Retschitzegger, W. (Editors) *Web Engineering. The Discipline of Systematic Development of Web Applications*, John Wiley & Sons Ltd., 2006.

(2) Koch, N.; Knapp, A.; Zhang, G.; Baumeister, H. "UML-Based Web Engineering. An Approach Based on Standards", Chapter 7 en: Rossi, G.; Pastor, O.; Schwabe, D.; Olsina, L.

(Editors) *Web Engineering: Modelling and Implementing Web Applications*, Springer-Verlag London Limited, 2008.]

4.1 Interfaz gráfica (Mockups)

5 GESTIÓN DE LA CONFIGURACIÓN

Debe especificar los lenguajes de programación y el stack tecnológico que debe ser utilizado en la construcción de la solución.

especificar el uso y estructura de los repositorios de código

acceso y uso de la plataforma tecnológica en la nube con la que cuenta cada equipo a configuración y uso de los ambientes de desarrollo y producción.

6 PRUEBAS

6.1 Descripción de pruebas unitarias

6.2 Descripción de pruebas de aceptación

7 GLOSARIO

•

8 ANEXO(S)

[ES OPCIONAL]

[Se pueden agregar anexos, si se consideran necesarios para obtener mayor claridad en el contenido del documento.]