



Universidad Nacional Autónoma de México

Programación Orientada a Objetos

Requerimientos de Software

V0.5

Fecha:15/12/2022

Integrantes

- **Aranda Solis Ricardo**
- **García González Alejandro**
- **Rodríguez Rodríguez Arturo**
- **Sosa Cortez Misael Ivan**



Ingeniería de la Computación, Universidad Nacional Autónoma de México
Programación Orientada a Objetos - Requerimientos de Software

Contenido

Histórico de Cambios	3
1. Descripción general	4
1.1. Objetivos del software	4
1.2. Glosario de términos	4
2. Requisitos funcionales	5
2.1. Identificación de los casos de uso	5
2.2. Detallado de los casos de uso	5
2.2.1 Pedir datos de alumnos	5
2.2.2 Actualizar datos académicos	6
2.2.3 Exportar datos del alumno	6
2.3. Prototipo de la interfaz del usuario	6
2.4. Plan de pruebas del sistema	8
3. Requisitos no funcionales	9
3.1. Usabilidad	9
3.2. Eficiencia	9
3.3. Variabilidad	9
3.4. Restricciones de diseño y construcción	9
3.5. Reusabilidad	9



Ingeniería de la Computación, Universidad Nacional Autónoma de México
Programación Orientada a Objetos - Requerimientos de Software

Histórico de Cambios

Versión	Descripción	Responsable de Actualización	Fecha de actualización
v0.1	Creación del documento y Relleno de requerimientos (Entregables 1, 2 y 3)	Aranda Solís Ricardo	24/10/2022
v0.2	Modificación del documento y requisitos funcionales (Entregables 1, 2 y 3)	García González Alejandro	24/10/2022
v0.3	Modificación de prototipo de Interfaz de usuario y plan de pruebas del sistema	Aranda Solís Ricardo	25/10/2022
v0.4	Modificación de requisitos funcionales	García González Alejandro	26/10/2022
v0.5	Modificación de requisitos funcionales	Aranda Solís Ricardo	26/10/2022



Ingeniería de la Computación, Universidad Nacional Autónoma de México
Programación Orientada a Objetos - Requerimientos de Software

1. Descripción general

1.1. Objetivos del software

En la facultad de ingeniería se debe de hacer la gestión de los datos de los alumnos, para esto es necesario desarrollar un programa que permita crear datos personales como lo son: nombre, apellidos, edad, dirección particular. Además, que, para la buena implementación en la facultad de ingeniería, debe de generar datos académicos (semestres, créditos, números de inscripción) de la carrera que estamos cursando, con las restricciones que existen en la inscripción de materias. Por otra parte, es importante poder hacer la modificación de los registros académicos, exceptuando los números de inscripción, todo lo anterior siguiendo el patrón de diseño MVC, es decir que el programa reciba peticiones del usuario y mediante una comunicación del controlador y el modelo creado para evaluar los datos, se le dé al usuario una salida de su petición. Finalmente, todos los datos deberán de ser exportados a un archivo .csv para ser visualizado en Excel por los directivos.

1.2. Glosario de términos

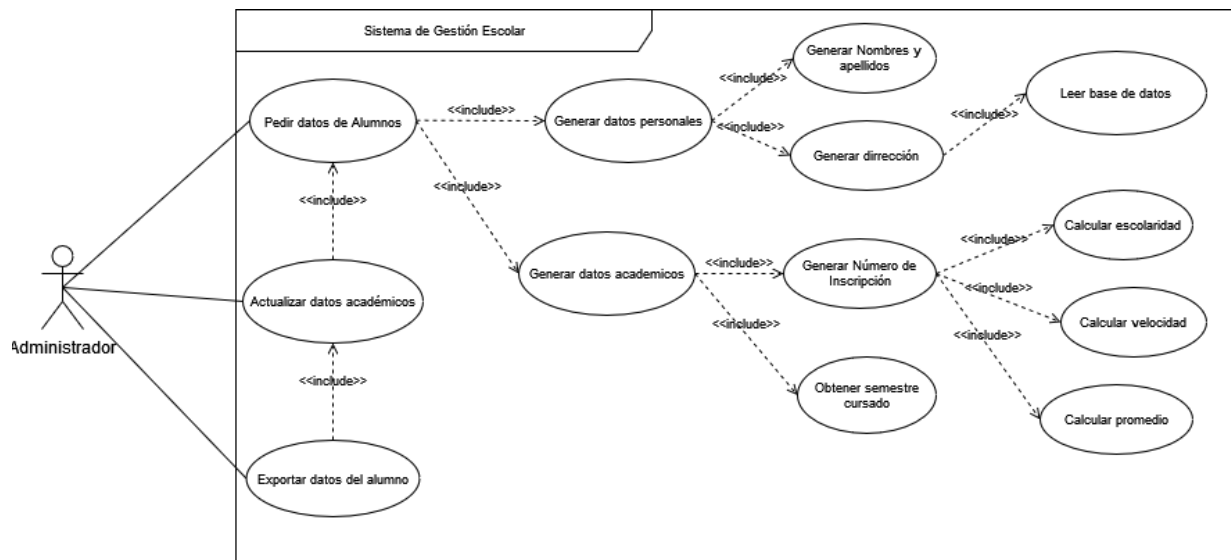
Término	Significado en el software
Datos del alumno	Nombre (nombre de pila y apellidos), edad, dirección, créditos, número de inscripción.
Créditos	Valor numérico de cada una de las materias en un solo semestre.
Número de inscripción	Cálculo numérico que se calcula como: $Promedio * \frac{Asignaturas aprobadas en ordinario}{Asignaturas inscritas en ordinario} * \frac{Créditos del alumno}{Créditos desde el ingreso}$
Dirección	Ubicación del domicilio del alumno.
Gestión de datos del alumno	Registrar, modificar, y en este caso generar datos acerca del alumno.
Sistema generador de datos personales	Sistema que permite generar al menos 50 nombres y una estructura de 50 apellidos.
Sistema generador de número de inscripción	Sistema que hace el cálculo de los números de inscripción de acuerdo a la historia académica.
Sistema generador de datos académicos	Sistema que genera los datos del alumno que tienen que ver con la facultad.
Sistema de modificación de datos académicos	Sistema que permite actualizar la información de los alumnos, exceptuando el número de reinscripción.

Ingeniería de la Computación, Universidad Nacional Autónoma de México

Programación Orientada a Objetos - Requerimientos de Software

2. Requisitos funcionales

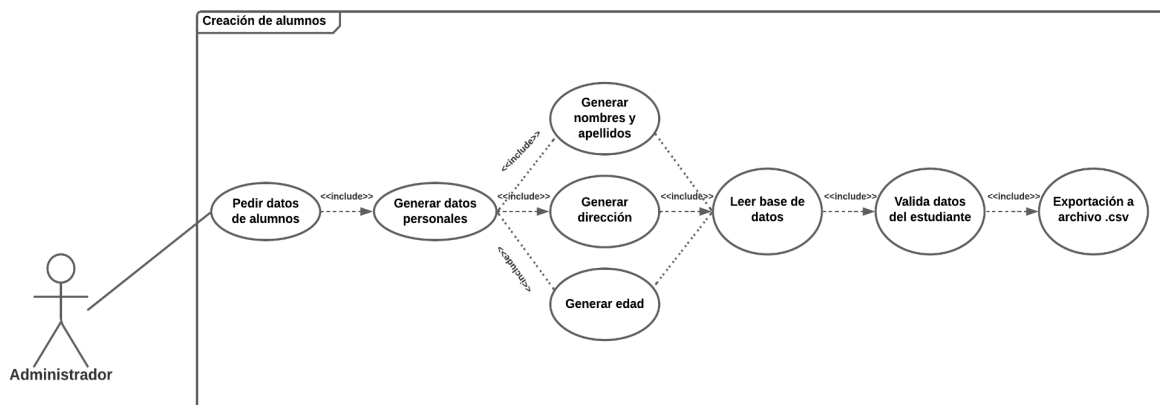
2.1. Identificación de los casos de uso



2.2. Detallado de los casos de uso

2.2.1 Pedir datos de alumnos

En este apartado se generarán datos para crear 500 alumnos para esto, anteriormente de crear un alumno, existirá un generador de nombres, un generador de apellidos, un generador de direcciones y un generador de edades, con esta información el programa se encargará de crear 500 alumnos de manera aleatoria cuidando que se cumplan cada una de las restricciones para dichos datos, y por último estos datos se guardarán en un archivo csv.





2.2.3 Exportar datos del alumno

2.3. Prototipo de la interfaz del usuario

Gestor de Datos del Alumno

Opciones

No se ha seleccionado ninguna opción

Iniciar

Nombre

Apellido 1

Apellido 2

Edad

No. de insc...


Dirección

Tabla sin contenido




Ingeniería de la Computación, Universidad Nacional Autónoma de México

Programación Orientada a Objetos - Requerimientos de Software

 Gestor de Datos del Alumno

Generar datos del Alumno ▾ Generar datos del Alumno seleccionado

Iniciar	Nombre	Apellido 1	Apellido 2	Edad	No. de insc...	Dirección
Tabla sin contenido						

 Gestor de Datos del Alumno

Generar datos del Alumno ▾ Iniciando

- Generar datos del Alumno
- Calcular números de inscripción
- Modificar datos del Alumno
- Exportar datos a csv

Iniciar	Nombre	Apellido 1	Apellido 2	Edad	No. de insc...	Dirección
Tabla sin contenido						



Ingeniería de la Computación, Universidad Nacional Autónoma de México
Programación Orientada a Objetos - Requerimientos de Software

2.4. Plan de pruebas del sistema

Pruebas	Acciones	Prueba a superar
Registro/Generación de Datos Académicos	Registrar/Generar	Escribir un float con un rango de 5 a 10, además de calificaciones nulas como "NP". Además que generar las materias del plan de estudios y el avance de créditos de los mismos
	Actualizar	Tener la capacidad de sobrescribir un dato del registro académico, exceptuando el número de inscripción y respetando las restricciones de "registrar"
	Eliminar	Hacer una calificación Nula (NULL)
	Lectura	Poder tomar los valores académicos para poder operar con ellos (Para poder sacar promedio, escolaridad y velocidad)
Cálculo de número de inscripción	Operar datos	Poder operar de manera correcta con la función: $No. \text{ de inscripción} = \frac{Promedio * \frac{Asignaturas \text{ aprobadas en ord}}{Asignaturas \text{ inscritas en ord}} * \frac{Créditos \text{ del alumno}}{Créditos \text{ desde el ingreso}}}{1}$
Registro/Generación de Datos Personales	Registrar/Generar	Tener la capacidad de generar nombres de pila y apellidos(solo dos), así como su edad(int).
	Actualizar	Sobreescribir alguno de los datos anteriores, incluyendo la dirección
	Eliminar	Poder suprimir los datos anterior mencionados, actualizando el número de alumnos en caso de eliminar a algún alumno completamente
	Leer	En este caso ser capaz de leer una base de datos de direcciones, para finalmente asignar aleatoriamente a una dirección a cada estudiante.
MVC	Implementación	Lograr separar la lógica, datos, y la interfaz de usuario, con el fin de



Ingeniería de la Computación, Universidad Nacional Autónoma de México
Programación Orientada a Objetos - Requerimientos de Software

		mantener actualizado al usuario, recibir las entradas del mismo y del acceso y operación de datos para la correcta implementación del modelo
--	--	--

3. Requisitos no funcionales

3.1. Usabilidad

Se seguirá la ISO/IEC 9126-1:2001 que define la usabilidad como la capacidad que tiene un producto software para ser entendido, aprendido, operable, atractivo para el usuario y conforme a estándares/guías, cuando es utilizado bajo unas condiciones específicas.

Por ello la usabilidad se divide en las siguientes subcaracterísticas:

- Fácil de aprender: capacidad para ser aprendido en su manejo
- Fácil de entender: capacidad del producto de ser entendido por nuevos usuarios en términos de su propósito y cómo puede ser usado en tareas específicas.
- Fácil de operar: capacidad del producto de ser operado y controlado en cualquier momento.
- Atractivo: capacidad del producto de ser atractivo para sus usuarios.
- Conformidad: capacidad del producto para adherirse a estándares, convenciones, guías de estilo o regulaciones relacionadas con la usabilidad.

3.2. Eficiencia

El programa debe de ser capaz de generar los datos que se piden en un tiempo razonable, permitiendo que el administrador pueda realizar eficientemente su trabajo

3.3. Variabilidad

El programa debe de poder tener datos diferentes (dentro de su rango) en cada ejecución.

3.4. Restricciones de diseño y construcción

El sistema deberá estar codificado en algún lenguaje orientado a objetos.

3.5. Reusabilidad

El sistema será desarrollado sin comprar o reutilizar componentes de otro software.