



Universidad Nacional Autónoma de México

Programación Orientada a Objetos

Requerimientos de Software

Versión 0.2

Fecha: 27/Octubre/2022

Integrantes:

- **Hernández Ruiz Leny Javier**
- **Ochoa Lozano Miguel Angel**
- **Silverio Martínez Andrés**
- **Trujillo Salazar Wendy Jazmin**



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

Contenido

Histórico de Cambios	3
Descripción general	4
Objetivos del software	4
Glosario de términos	4
Requisitos funcionales	5
Identificación de los casos de uso	5
Detallado de los casos de uso	5
Prototipo de la interfaz del usuario	6
Plan de pruebas del sistema	6
Requisitos no funcionales	7
Usabilidad	7
Eficiencia	8
Mantenimiento	8
Restricciones de diseño y construcción	8
Reusabilidad	8



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)	Trujillo Salazar Wendy Jazmin	26/10/2022
v0.2	Modificación del documento y requisitos funcionales (Entregables 1, 2 y 3)	Hernández Ruiz Leny Javier Ochoa Lozano Miguel Ángel Silverio Martínez Andrés Trujillo Salazar Wendy Jazmin	27/10/2022



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

1. Descripción general

1.1. Objetivos del software

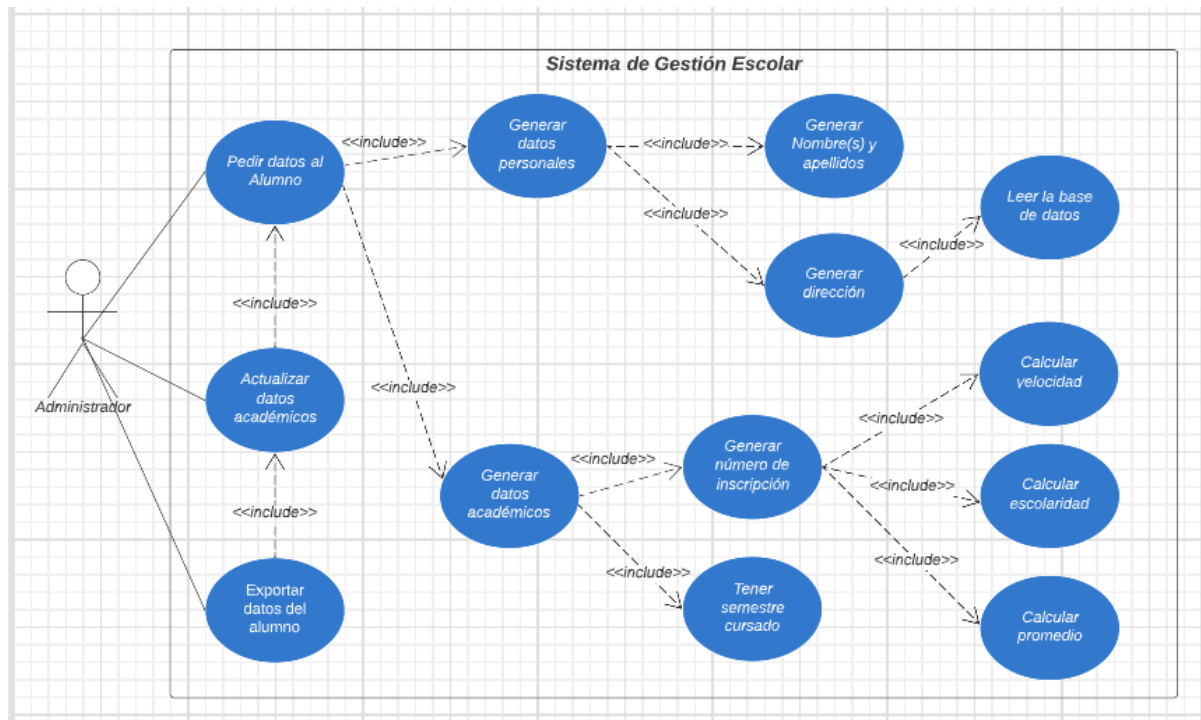
En la facultad de ingeniería se debe de realizar la gestión de los datos de varios alumnos, por lo que se necesita crear un programa que sea capaz de almacenar datos personales como: nombre, apellidos, edad, dirección y para el caso de datos académicos como: semestres, créditos y números de inscripción de la carrera a la que pertenecen, siguiendo las restricciones que existen durante la inscripción de materias. Cabe mencionar que es importante la modificación de los registros académicos, a excepción de los números de inscripción, siguiendo el patrón de diseño MVC (que el programa reciba peticiones del usuario) a través del controlador y el modelo creado para evaluar los datos, se le pueda dar al usuario la salida de su petición de modo que todos los datos sean exportados a un archivo .csv y ser visualizado en Excel por el usuario.

1.2. Glosario de términos

Término	Significado en el software
Alumno	Usuario que ingresa sus datos (nombre(s), apellidos, edad, dirección, créditos y número de inscripción).
Dirección	Ubicación donde reside el alumno.
Créditos	Valoración determinada en cada una de las materias que especifican en el plan de estudios.
Número de inscripción	$Promedio * \frac{Asignaturas aprobadas en ordinario}{Asignaturas inscritas en ordinario} * \frac{Créditos del alumno}{Créditos desde el ingreso}$
Gestión de datos del alumno	Registrar, consultar y cancelar una cita para asesoría.
Sistema generador de datos personales	Consultar y rechazar o aceptar una cita para asesoría.
Sistema generador de número de inscripción	Persona que da asesorías a alumnos respecto a una ciencia, un arte, una técnica, etc.
Sistema generador de datos académicos	Proceso mediante el cual un alumno realiza un registro para solicitar a un profesor una cita para asesoría.
Sistema de modificación de datos académicos	Sistema de programación flexible que permite que el código reorganice los elementos de la web, adaptándolos a la resolución de la pantalla donde están siendo visualizados.

2. Requisitos funcionales

2.1. Identificación de los casos de uso



2.2. Detallado de los casos de uso

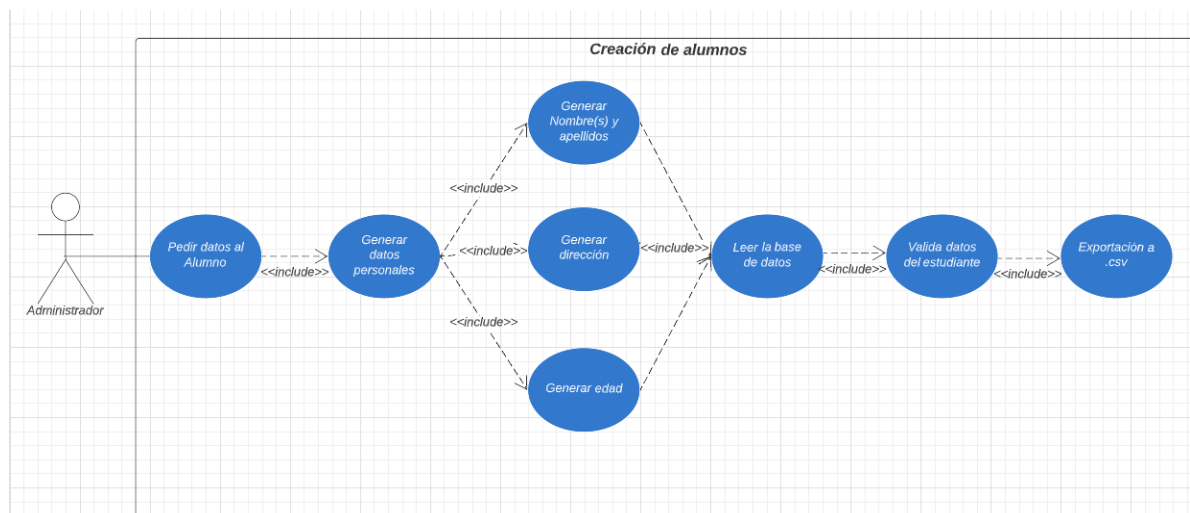
2.2.1. Pedir datos del alumno

En este apartado se generarán datos para crear 500 alumnos, para esto existirá un generador de nombres, apellidos, direcciones y edades. Con esta información el programa podrá crear 500 alumnos de forma aleatoria siempre y cuando se cumplan cada una de las restricciones y a su vez se guardarán en un archivo csv.



Ingeniería de la Computación, Universidad Nacional Autónoma de México

Programación Orientada a Objetos - Requerimientos de Software



2.2.2. Actualizar datos académicos

Para actualizar los datos se deberá ingresar a esta opción en la interfaz y después de esto se seleccionará en la lista de opciones el dato que el usuario desea modificar, ya sean datos personales o académicos; como es el caso de las calificaciones, que se podrán modificar exceptuando el número de inscripción y dichos datos serán exportados a un archivo csv.

2.2.3. Exportar datos del alumno

Se seleccionará esta opción y se podrá iniciar, exportando la tabla que se mostrará en la interfaz, pero se deben de recopilar todos los nodos y transcribirlos en un archivo que se creará posteriormente, y al escribirlo en archivo .csv se separarán los datos por comas pero obteniendo los datos de forma individual, y finalmente coordinarse con la acción iniciar del usuario.

2.3. Prototipo de la interfaz del usuario

Salida en Apache NetBeans.

2.4. Plan de pruebas del sistema

Pruebas	Acciones	Prueba a superar
Registro de los datos académicos del alumno	Registrar	Añadir un rango de calificaciones, por ejemplo de 6 a 10, además de calificaciones no aprobatorias.
	Actualizar	Sobreescribir algún dato que sea incorrecto o que esté en otro lugar al que no pertenezca
	Eliminar	Que exista una calificación nula (NULL)



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

	Lectura/Impresión	Tomar los valores académicos para leerlos e imprimirlos más adelante
Número de inscripción	Operar datos	Calcular número de inscripción operando de la siguiente manera: $Promedio * \frac{Asignaturas aprobadas en ordinario}{Asignaturas inscritas en ordinario} * \frac{Créditos del alumno}{Créditos desde el ingreso}$
Generación de datos personales del alumno	Registrar	Generar nombres y apellidos, así como su edad
	Actualizar	Sobreescribir algún dato de los que se registraron
	Eliminar	Suprimir los datos que estén equivocados o se encuentren en un lugar incorrecto
	Leer	Leer los datos dentro de la base de datos de direcciones, para asignar una dirección a cada estudiante
MVC	Implementación	Lograr separar la lógica, los datos y la interfaz de usuario, esto con el fin de que nuestro programa ejecute a la perfección

3. Requisitos 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 tiempo promedio de respuesta del programa debe de ser rápido, para que no genere contratiempos al usuario



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

3.3. Mantenimiento

Una vez entregado el programa, el soporte y mantenimiento periodico dependerá de las veces que el usuario así lo requiera (Se cobrará menos por esto).

3.4. Restricciones de diseño y construcción

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

3.5. Reusabilidad

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