



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

**Sistema web de apoyo para la
asignatura *Cómputo Móvil***

TESIS

Que para obtener el título de
Ingeniero en Computación

P R E S E N T A

Juan Carlos Espinosa Ortiz

DIRECTOR DE TESIS

Ing. Marco Antonio Martínez Quintana



Ciudad Universitaria, Cd. Mx., 2019

Dedicatoria ...

*a mi hermano,
quien ha estado conmigo
desde el primer minuto de vida.*

Agradecimientos

A mis padres por darme la oportunidad de concluir este ciclo y por aconsejarme cuando ha sido necesario.

A la Universidad Nacional Autónoma de México, a la Facultad de Ingeniería y a mis profesores por darme las herramientas técnicas y humanas para desempeñarme profesionalmente.

Índice general

Agradecimientos	II
Introducción	IX
Planteamiento del problema	XI
Objetivos	XII
1 Marco teórico	1
1.1 Introducción al mundo móvil e inalámbrico	1
1.1.1 Hardware	1
1.1.2 Software	2
1.1.3 Redes	3
1.1.4 Proveedores de servicio	4
1.1.5 Proveedores de Telefonía Móvil	4
1.1.6 Proveedores de Servicios de Internet	5
1.1.7 Comercio Móvil	5
1.2 Dispositivos móviles	6
1.2.1 Clasificación de los teléfonos móviles	8
1.2.2 Sistemas Operativos	9
1.3 Arquitecturas para aplicaciones móviles y envío de mensajes	12
1.3.1 Envío de mensajes	13
1.3.2 Seguridad móvil	14
1.4 Construcción de aplicaciones de clientes inteligentes	16

<i>ÍNDICE GENERAL</i>	IV
1.4.1 Arquitectura Cliente-Servidor	16
1.4.2 Desarrollo de clientes inteligentes	18
1.5 Construcción de aplicaciones en Internet inalámbrica	19
1.5.1 Clientes Ligeros	19
1.6 Administración de datos en sistemas móviles	24
1.6.1 Bases de datos móviles	24
1.6.2 Sincronización de bases de datos	25
2 Desarrollo	27
2.1 Fase de análisis de requerimientos	27
2.1.1 Historias de usuario para rol Administrador	27
2.1.2 Historias de usuario para rol Profesor	28
2.1.3 Historias de usuario para rol Alumno	30
2.1.4 Diagramas de Casos de Uso	32
2.2 Fase de diseño y construcción	35
2.2.1 Diseño de la Base de Datos	36
2.2.2 Diseño del Web API REST	38
2.2.3 Diseño de la Capa de Presentación	43
2.3 Fase de implementación y pruebas	45
2.3.1 Preparación del Grupo de Recursos en <i>Microsoft Azure</i>	45
3 Pruebas de la plataforma	48
4 Resultados	50
5 Conclusiones	52
6 Apéndices	53
6.1 Repositorio de Código	53
6.2 Casos de Prueba	54
6.2.1 Casos de prueba Módulo Administrador	54

6.2.2	Casos de prueba Módulo Profesor	58
6.2.3	Casos de prueba Módulo Alumno	65
6.3	Manual Técnico	68
6.3.1	Configuración de la Base de Datos	69
6.3.2	Configuración del Servidor Web API	74
6.3.3	Configuración del Servidor Web	77
6.4	Manuales de Usuario	77
6.4.1	Módulo Administrador	78
6.4.2	Módulo Profesor	81
6.4.3	Módulo Alumno	92
7	Referencias	97

Índice de figuras

1.1	DynaTAC 8000X Motorola	2
1.2	Aparece software especial y variado para dispositivos móviles.	3
1.3	Participación por operador en suscripciones móviles en México (2018). Fuente: Telconomia. 2018	4
1.4	Ejemplos de empresas de M-Commerce en México.	5
1.5	Ejemplo de teclado integrado en pantallas táctiles.	6
1.6	Ejemplo de micrófono para dispositivos móviles.	7
1.7	Ejemplo de cámara para dispositivos móviles.	7
1.8	Ejemplo de pantalla táctil para dispositivos móviles.	7
1.9	Representación gráfica de las Capas del S.O.	10
1.10	Sistemas Operativos móviles. Fuente: impulsonegocios.com	11
1.11	Arquitectura Cliente-Servidor. Fuente: brianjcoleman.com	18
1.12	Ejemplo del desarrollo de un cliente ligero.	21

1.13	Uso de micronavegadores móviles en México (2018). Fuente: gs.statcounter.com	24
1.14	Representación gráfica del almacenamiento-reenvío	25
1.15	Representación gráfica de sincronización continua	26
2.1	Diagrama de Caso de Uso de inicio de sesión y registro	32
2.2	Diagrama de Caso de Uso del módulo de Administrador	32
2.3	Diagrama de Caso de Uso del módulo de Profesor	33
2.4	Diagrama de Caso de Uso del módulo de Alumno	34
2.5	Representación gráfica la arquitectura del proyecto.	36
2.6	Modelo relacional de la Base de Datos.	38
2.7	Estructura de la solución API.	39
2.8	Estructura del proyecto ComputoMovil.DTOS.	39
2.9	Ejemplo de Enum.	40
2.10	Ejemplo de clase DTO.	40
2.11	Estructura del proyecto ComputoMovilAPI.	41
2.12	Ejemplo de servicio API.	42
2.13	Proceso de generación de token al iniciar sesión.	43
2.14	Estructura de la Capa de Presentación.	44
2.15	Ejemplo de interfaz de Profesor.	44
2.16	Grupo de recursos en Microsoft Azure.	45
2.17	Perfil de publicación en Visual Studio para Azure App Service.	46
2.18	Transferencia de archivos de ambiente de desarrollo a ambiente de producción.	47
6.1	Estructura del repositorio de código.	53
6.2	Historial de cambios efectuados en el repositorio.	54
6.3	Configuración de credenciales de correo electrónico.	74
6.4	Configuración de la cadena de conexión de la Base de Datos.	74
6.5	Configuración del archivo WebApiConfig.cs.	75
6.6	Configuración CORS en servicios web.	75
6.7	Configuración CORS en Azure.	76
6.8	Configuración de la zona horaria del Web App en Azure.	76

6.9	Ubicación del botón “Alta Profesor” en el panel de Administrador.	79
6.10	Ventana emergente “Alta Profesor” para registrar a un nuevo profesor. . . .	79
6.11	Ubicación del botón “Editar Profesor”.	80
6.12	Ventana emergente “Actualizar Profesor”.	80
6.13	Ubicación del botón “Eliminar Profesor”.	81
6.14	Pantalla de la sección de “Perfil de Profesor”.	82
6.15	Ubicación del botón “Alta Curso”.	83
6.16	Ventana emergente “Alta Curso” para registrar un nuevo curso.	83
6.17	Ubicación del botón “Editar Curso”.	83
6.18	Ventana emergente “Editar Curso”.	84
6.19	Ubicación del botón “Eliminar Curso”.	84
6.20	Mensaje de confirmación de eliminación de curso.	85
6.21	Botón para seleccionar curso en la sección de administrar actividades. . . .	85
6.22	Ventana emergente “Nueva Actividad”.	86
6.23	Ventana emergente “Editar Actividad”.	87
6.24	Mensaje de confirmación “Eliminar Actividad”.	88
6.25	Apartado “Calificaciones de Actividades”.	89
6.26	Ventana emergente de detalles de Tarea.	89
6.27	Pantalla de “Calificaciones Finales” antes de asignar la calificación final. . .	90
6.28	Pantalla de “Calificaciones Finales” después de asignar la calificación final.	90
6.29	Ventana emergente “Nuevo Aviso”.	91
6.30	Ventana emergente “Editar Aviso”.	91
6.31	Ubicación del botón “Eliminar Aviso”.	92
6.32	Pantalla de la sección de “Perfil de Alumno”.	93
6.33	Listado de cursos del sistema antes de la inscripción.	93
6.34	Listado de cursos del sistema después de la inscripción.	94
6.35	Ventana emergente “Entregar Tarea”.	94
6.36	Ventana emergente “Entregar Tarea” con datos precargados.	95
6.37	Pantalla de calificaciones antes de recibir evaluaciones.	96
6.38	Pantalla de calificaciones después de recibir evaluaciones.	96

6.39 Pantalla de la sección de avisos.	96
--	----

Índice de tablas

1.1 Generaciones del Cómputo Móvil	4
1.2 Tipos de dispositivos móviles.	9
1.3 Bases de datos móviles más populares. Fuente: CMAP Foundation Level Course.	25
3.1 Ejemplo de Caso de Prueba.	48

Introducción

El Cómputo Móvil es un área de la tecnología que ha ido adquiriendo cada vez más importancia a lo largo de las últimas décadas, los avances tecnológicos han dado lugar a un ambiente propicio para la evolución y creación de nuevas formas de computación móvil.

Para poder tener un mayor entendimiento del Cómputo Móvil es importante tener presente el significado de cada una de las palabras que lo componen -*Cómputo* y *Móvil*-, en primer lugar; la palabra *Cómputo* rápidamente la asociamos al objeto computadora, es decir, al aparato físico que cotidianamente utilizamos para realizar trabajos informáticos, desarrollar y utilizar software, navegar en Internet, etcétera. Sin embargo, computar, en el sentido más estricto de la palabra, se refiere a la acción de calcular o contar, así entonces podemos definir una computadora como un objeto que realiza cálculos, pero no sólo las PC o las laptops son computadoras, como ejemplo se puede mencionar el ábaco, un aparato pionero en la computación ya que a través del uso de este objeto era más fácil y rápido realizar cálculos numéricos, sin embargo no se le puede considerar una computadora como las conocemos actualmente ya que este instrumento no ejecuta acciones de manera automática, otro ejemplo es el reloj digital el que calcula la hora realizando un conteo de pulsos a partir de una señal periódica, este es capaz de almacenar información al establecer una alarma y ejecutarla de manera automática.

Por otro lado el término *Móvil*, se le atribuye a aquellos objetos que de acuerdo a sus características pueden funcionar mientras se desplazan por el espacio, es decir son autónomos. Volviendo al ejemplo del reloj digital, podemos decir que es una computadora móvil ya que lo llevamos libremente con nosotros, ahora ¿qué le permite ser móvil y autónomo? la movilidad la permite la batería en su interior, ya que le brinda autonomía para desplazarse y seguir funcionando.

Además de la autonomía de una fuente de energía, las computadoras móviles también necesitan autonomía para poder estar comunicadas, la solución a este problema fueron las ondas de radio las cuales permiten la comunicación entre computadoras de forma inalámbrica. Para llevar a cabo esto es necesario la implementación de protocolos y estándares para entablar canales de comunicación. Es por ello que la materia de *Cómputo Móvil* en la carrera de Ingeniería en Computación es importante pues la humanidad está inmersa en un sinnúmero de computadoras móviles por ello la necesidad de tener bases fuertes en dicha asignatura y de esta manera ser fuertes competidores a nivel profesional.

La intención de este trabajo es implementar un sistema web que sirva de apoyo para el aprendizaje de la materia *Cómputo Móvil*, abarcando cada uno de los tópicos definidos en ella, el sistema se enfocará principalmente en servir de plataforma (dicho sea de paso, móvil) para un mejor aprovechamiento por parte del alumnado de la asignatura y una mejor administración docente de los grupos inscritos en ella.

Planteamiento del problema

Actualmente la tecnología avanza a una gran velocidad y esto trae consigo el desarrollo de diferentes herramientas, pero también la obsolescencia de algunas otras a corto plazo, por lo que es importante mantener a los alumnos actualizados en cuanto a las tendencias del mercado y tengan con ello una formación competitiva e integral. La Facultad de Ingeniería actualiza cada cierto tiempo los temarios de cada asignatura, así como los planes de estudio, sin embargo, este lapso es en años y en este tiempo, específicamente en el área del cómputo móvil pueden surgir nuevos conceptos, metodologías, herramientas y he aquí la importancia de mantener al alumnado actualizado en la materia.

Es importante mencionar que Cómputo Móvil es una asignatura con un total de 100 % horas teóricas por lo que contar con un sistema web complementará el enfoque práctico de la asignatura, dando así cumplimiento a las necesidades de los estudiantes.

Objetivos

El objetivo principal de este trabajo es desarrollar un sistema web que dé soporte al aprendizaje de los tópicos relacionados con la asignatura Cómputo Móvil y de esta forma se puedan complementar las clases presenciales. El sistema web contará con un módulo para cada uno de los temas que se imparten en la asignatura de Cómputo Móvil, los cuales son:

- Introducción al mundo Móvil e Inalámbrico
- Dispositivos Móviles
- Arquitecturas para aplicaciones móviles y envío de mensajes
- Construcción de aplicaciones de clientes inteligentes
- Construcción de aplicaciones en Internet inalámbrica
- Administración de datos en sistemas móviles

En donde para cada módulo se determinará qué subtemas y materiales son los más adecuados para el sistema y que sean un verdadero apoyo para la labor docente y el correcto aprendizaje en los alumnos. El sistema también contará con un módulo que permitirá al profesor llevar una inspección del avance de los alumnos más de cerca.

El trabajo tratará de apegarse a los lineamientos establecidos por las mejores prácticas de desarrollo y de análisis del negocio al pretender incluir las fases de análisis de requerimientos, de diseño-construcción y de implementación-pruebas, de tal forma que se obtenga

un trabajo de buena calidad que satisfaga la problemática planteada en este escrito. Finalmente, se contará con manuales técnicos y de usuario para que sea posible dar continuidad al sistema web.

Capítulo 1

Marco teórico

1.1. Introducción al mundo móvil e inalámbrico

El Cómputo Móvil involucra dispositivos capaces de realizar tareas de cómputo de manera inalámbrica. Por lo tanto, tienen que ver dos avances tecnológicos importantes: las redes inalámbricas para transmisión de datos y la miniaturización electrónica, al grado de portar dispositivos como un accesorio más de vestir. Gracias a estos avances y al continuo desarrollo de aplicaciones y sistemas operativos (*software*) cada vez más sofisticados y especializados es posible satisfacer tanto necesidades personales como laborales.

1.1.1. Hardware

La miniaturización de los dispositivos electrónicos, es decir la fabricación de hardware de dimensiones cada vez más reducidas, se ha logrado a través del refinamiento de las técnicas de producción. El desarrollo de nuevas tecnologías de almacenamiento y los avances en la comunicación inalámbrica han abierto un nuevo campo en el área de la computación, el computo móvil. Esta rama de la computación ha cobrado relevancia en los últimos años debido a sus bastas aplicaciones.

Durante los primeros años de la existencia de las computadoras, éstas eran de dimensiones considerablemente grandes y de características bastante humildes. Fue hasta finales de la década de los 70's que los ordenadores personales fueron una realidad, acercándose

por primera vez las tecnologías de la computación al uso “cotidiano”.

Uno de los primeros ejemplos de computadoras móviles fueron las calculadoras lanzadas al mercado por *Casio* en 1957, este dispositivo usaba 342 relevadores eléctricos para resolver problemas de adición, sustracción, multiplicación y división, de hasta 14 dígitos.

Años más tarde, en 1983 *Motorola* presenta el *DynaTAC 8000X*, el primer teléfono celular de la historia. La batería sólo daba para una hora de conversación u 8 horas en *stand-by*. La calidad de sonido del teléfono móvil era muy mala, era pesado y poco estético. A pesar de todo marcó el inicio de la telefonía móvil.

En la actualidad existen tecnologías sofisticadas que permiten integrar dispositivos electrónicos en un espacio limitado a un costo razonable lo que ha propiciado que, junto con los avances en la transmisión de datos, se desarrollen nuevos dispositivos de cómputo que satisfagan diversas necesidades humanas, ejemplos de estos son: la telefonía celular, las calculadoras, los relojes digitales, etcétera.



Figura 1.1: DynaTAC 8000X Motorola

1.1.2. Software

El desarrollo de *software* ha evolucionado casi en conjunto con el *hardware* de los dispositivos móviles. Las primeras calculadoras portátiles contaban con un pequeño mecanismo que mostraba las operaciones, pero con el paso de los años, al aparecer nuevas tecnologías como las pantallas LCD, se tuvieron que desarrollar pequeños sistemas de *software* para ofrecer una mejor experiencia de usuario que a su vez permitieran explotar las nuevas

capacidades que ofrecía la era digital.

En el ámbito móvil, el cambio fue radical a finales del siglo pasado. Los avances tecnológicos dotaron de una amplia gama de servicios que empezaron a aparecer en los celulares como juegos, calculadoras, agendas, calendarios, etcétera.

Actualmente el desarrollo de sistemas de *software* se encuentra presente en un sinnúmero de dispositivos, desde el sistema que controla la combustión en un cohete espacial hasta teléfonos inteligentes de un uso muy extendido a lo largo del planeta.



Figura 1.2: Aparece software especial y variado para dispositivos móviles.

1.1.3. Redes

En 1969 se creó la red ARPANET, desarrollada para la comunicación militar, marcaría el comienzo de lo que conocemos actualmente como Internet, interconectado dispositivos para compartir servicios o información.

En 1990 se consolida el *World Wide Web* como una red de comunicación disponible para todos y nacen servicios como el acceso remoto a otras máquinas (SSH y Telnet), transferencia de archivos (FTP), correo electrónico (SMTP), conversaciones en línea, transmisión de archivos (P2P, P2M, descarga directa), etc.

El Internet se ha consolidado como uno de los principales medios de intercomunicación en los dispositivos móviles. Sin embargo, existen otras tecnologías de transmisión de extensas aplicaciones como las microondas, el *Bluetooth*, satélites, etcétera.

Generación	Año	Descripción
1G	1980	Comunicación analógica
2G	1990	Digitalización, reducción de tamaño, estándar GSM y servicio SMS. Primera versión del protocolo WAP para el uso de Internet.
2.5G	Finales de los 90'	Mejora de velocidad de transmisión y mensajería multimedia.
3G	2001	Inclusión de servicios de Internet de manera más cómoda y fluida.
4G	2013	Uso de datos y de voz de manera extensiva que permiten incluso transmisiones de video en formato HD.

Tabla 1.1: Generaciones del Cómputo Móvil

1.1.4. Proveedores de servicio

Un Proveedor de Servicio (*Service Provider*) es una empresa que se encarga de prestar servicios a otras empresas o individuos mediante suscripciones o contratos. Ejemplos de los servicios que se ofrecen en el campo del cómputo móvil son: la telefonía celular, mensajería SMS, acceso a Internet, etcétera.

1.1.5. Proveedores de Telefonía Móvil

Este tipo de empresas se encargan de prestar servicios de telefonía inalámbrica a sus clientes. Existen dos tipos de proveedores de telefonía móvil, los Operadores de Red Móvil (MNO) y los Operadores de Red Móvil Virtuales (MVNO). Los primeros son aquellos que son dueños de la infraestructura que permite la comunicación móvil, mientras que los segundos compran al por mayor los servicios de un MNO para ofrecer los servicios de telefonía móvil a sus clientes.

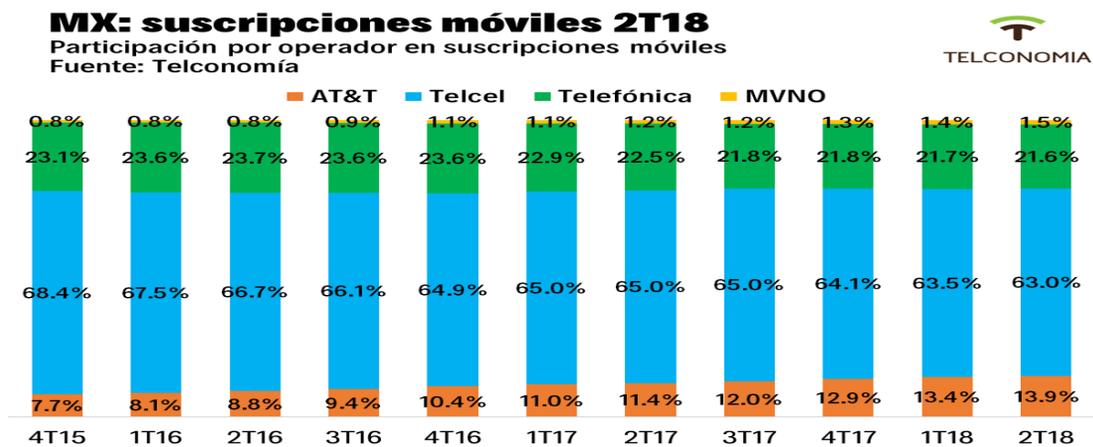


Figura 1.3: Participación por operador en suscripciones móviles en México (2018). Fuente: Telconomía. 2018

1.1.6. Proveedores de Servicios de Internet

Este tipo de empresas proporcionan a sus usuarios el acceso a los servicios de Internet. Existen diversos tipos de Proveedores de Servicios de Internet (ISP).

- ISP de Acceso.
- ISP de Buzón de Correo.
- ISP de Servidores.
- ISP de Transito.
- ISP Virtuales.
- ISP Gratuitos.

1.1.7. Comercio Móvil

El comercio móvil, comúnmente conocido como *M-Commerce*, hace referencia a la compra de productos y/o servicios a través de un dispositivo móvil. Por medio del *M-Commerce* nos es posible adquirir productos tangibles como ropa, calzado, autos e incluso casas, pero además también es posible adquirir productos intangibles como software, música, imágenes y videos. Por otra parte, a través del *M-Commerce* es posible acceder a servicios como la Banca Móvil, servicios de *streaming* de audio y video, servicios de información, etcétera.

Recientemente, se han incorporado también los sistemas de posicionamiento en los dispositivos móviles como lo son los Servicios Basados en la Localización (GPS), los cuales proveen un servicio de localización de personas u objetos en el espacio y tiempo haciendo uso del *hardware* y *software* presente en el dispositivo rastreado.

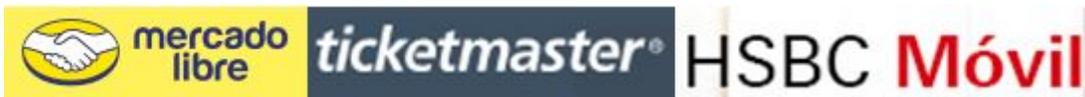


Figura 1.4: Ejemplos de empresas de M-Commerce en México.

1.2. Dispositivos móviles

Los dispositivos móviles son aquellos artefactos capaces de realizar sus funciones de manera relativamente autónoma y que, en la mayoría de los casos tienen la posibilidad de comunicarse con otros dispositivos ya sea móviles o fijos.

Como se ha mencionado, no sólo las laptops, los *smartphones* o las tabletas son computadoras móviles, sin embargo, la asignatura de Cómputo Móvil está centrada en este tipo de dispositivos. Para fines prácticos se contemplarán aquellos dispositivos móviles más cotidianamente utilizados.

La evolución de los dispositivos móviles no ha sido repentina, y es que con el avance tecnológico a través de los últimos años ha sido posible dotar de mejores elementos de *software* y *hardware* a estos dispositivos.

Los elementos de comunicación presentes en los dispositivos móviles son sumamente importantes para brindarles esa autonomía de comunicación que necesitan para interactuar con otros dispositivos, así mismo estos dispositivos cuentan con diversos mecanismos de entrada que les permiten operar e interactuar con los usuarios. A continuación, se mencionan algunos de estos mecanismos:

Teclado. Es el medio físico que sirve de interfaz entre el usuario y el dispositivo para ingresar a él información con el fin de poder ser procesada. Actualmente es común encontrar la interfaz del teclado integrada en las pantallas táctiles.



Figura 1.5: Ejemplo de teclado integrado en pantallas táctiles.

Micrófono. Es un dispositivo que transforma sonidos en impulsos eléctricos capaces de ser analizados por el dispositivo móvil.



Figura 1.6: Ejemplo de micrófono para dispositivos móviles.

Cámara. Es aquel dispositivo capaz de captar imágenes que luego son digitalizadas para poder ser entendidas de manera eléctrica.



Figura 1.7: Ejemplo de cámara para dispositivos móviles.

Pantalla Táctil. Mecanismo que proyecta imágenes en una superficie cristalina y que funciona como medio de entrada de información mediante la interacción táctil por parte del usuario.



Figura 1.8: Ejemplo de pantalla táctil para dispositivos móviles.

1.2.1. Clasificación de los teléfonos móviles

Los teléfonos móviles son dispositivos que nos permiten básicamente realizar comunicaciones telefónicas de manera inalámbrica haciendo uso de los servicios de un MNO (*Mobile Network Operator*) o bien un MVO (*Mobile Virtual Operator*), este tipo de prestadores de servicios nos brindan la posibilidad de entablar comunicaciones a través del uso de su infraestructura de red. No obstante, las posibilidades de los teléfonos actuales no se limitan al envío y recepción de mensajes o a la realización de llamadas telefónicas, ya que las capacidades de *software* y *hardware* presentes hoy en día han facilitado la ejecución de muchas tareas más en estos dispositivos.

Dada la evolución y las diversas necesidades que satisfacen los teléfonos móviles, la variedad de estos dispositivos en la actualidad es muy grande y aunque algunas clasificaciones que se presentan a continuación han quedado prácticamente en desuso, en su tiempo fueron las tecnologías más novedosas y requeridas a nivel mundial, de ahí la importancia de su mención.

Web Enabled Phones. Estos teléfonos móviles permiten el envío y recepción tanto de llamadas telefónicas como de mensajes SMS, presentan características de *software* y *hardware* discretas que les permiten realizar sus funciones sin problema alguno.

Low End Smartphones. Los teléfonos móviles de esta categoría permiten la comunicación vía texto o voz y adicionalmente permiten la instalación y ejecución de aplicaciones creadas por terceros.

High End Smartphones. Al igual que las categorías anteriores, este tipo de teléfonos móviles permiten la comunicación mediante mensajes de texto, voz, instalación de aplicaciones de terceros y se distinguen por el uso intensivo de datos.

Esta clasificación de los teléfonos móviles han ido aumentando sus prestaciones y de hecho las tecnologías que les dieron vida como SMS o llamadas de telefonía celular cada vez han sido menos solicitadas por los usuarios. De igual forma, no todas las necesidades de un usuario móvil actual pueden ser cubiertas por un teléfono móvil, es por esta razón que los avances en la computación personal móvil también han dado de que hablar en los últimos años. A continuación, se presenta una pequeña clasificación de los dispositivos

móviles.

Tablets. Son dispositivos comúnmente más pequeños que una laptop y de mayor tamaño que un teléfono celular. Generalmente estos dispositivos carecen de comunicación celular convencional, en su lugar se valen de las redes inalámbricas para conectarse a Internet y establecer comunicaciones basadas en los protocolos propios de esta tecnología.

Wearables. Son dispositivos de dimensiones tan pequeñas que son fáciles de portar en el cuerpo, están enfocadas a la interacción directa con el usuario que las porta. Ejemplos de estos dispositivos son relojes, pulseras, lentes digitales, etcétera.

Laptops, notebooks y ultrabooks. Son computadoras compactas que disfrutan la autonomía de movilidad adquirida gracias a una batería y tarjetas de red que les permiten ser utilizadas de manera *wireless*. Poseen prácticamente las mismas características que una PC de escritorio.

Tipo de dispositivo móvil	Descripción
Web Enabled Phones	Funcionalidades básicas de envío de SMS, llamadas y navegación web.
Low End Smartphones	Permiten la instalación de aplicaciones de terceros.
High End Smartphones	Uso extensivo de datos.
Tablets	Ofrecen prestaciones similares a los High End Smartphones, pero son de dimensiones más grandes
Wearables	Dispositivos de dimensiones pequeñas que ofrecen información directa al usuario

Tabla 1.2: Tipos de dispositivos móviles.

1.2.2. Sistemas Operativos

Además de las características físicas de los dispositivos móviles, estos cuentan con otra característica muy importante que les permite operar y brindarle una mejor experiencia al usuario que los porta, se trata del Sistema Operativo.

El Sistema Operativo de los dispositivos móviles, al igual que el de los dispositivos fijos, es el intermediario entre el *hardware* y el usuario, hace las veces de administrador de los recursos del aparato y además pone a la disposición del usuario las capacidades del dispositivo de la forma más amigable posible.

En el mundo móvil, los Sistemas Operativos han sido muchos y muy variados desde su aparición, pero todos comparten dos características: ser ligeros y estar enfocados a la movilidad.

La interacción entre la máquina y el usuario es por el Sistema Operativo, pero este proceso de comunicación no se realiza de forma directa pues para que ambas partes puedan comunicarse (usuario y máquina) el Sistema Operativo recorre las siguientes capas:

Capa de *Kernel*. También llamado Núcleo del Sistema, es el encargado de la administración del *hardware* disponible en el dispositivo.

Capa de *Middleware*. En esta capa se encuentran los programas o módulos que les permiten a las aplicaciones comunicarse con los recursos de *hardware* presentes en el dispositivo.

Capa de Administración de Aplicaciones. Es la encargada de la ejecución, detención y finalización de las aplicaciones alojadas en el Sistema Operativo.

Capa de Interfaz. Esta es la intermediaria entre el usuario y los recursos del dispositivo a través de los diversos mecanismos de entrada y salida presentes en el aparato electrónico.

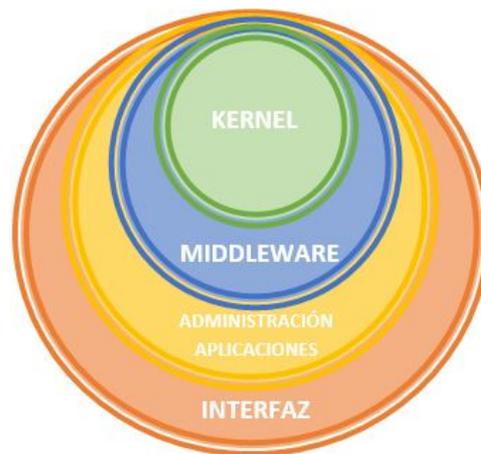


Figura 1.9: Representación gráfica de las Capas del S.O.

Los sistemas operativos móviles de mayor popularidad han sido los siguientes:

Windows CE. La versión *Compact Edition* de *Windows* fue diseñada específicamente para operar en dispositivos con recursos limitados. *Windows CE* es un sistema operativo de 32 bits con capacidad de trabajar multitareas, además se trata de un sistema operativo modular, es decir, se pueden agregar únicamente las partes funcionales que requiera el dispositivo en el que trabajará.

Android. Es un sistema operativo basado en *Linux* creado por la compañía estadounidense *Google*. Se caracteriza por su fácil personalización, interfaz intuitiva y por la versatilidad en cuanto los dispositivos en los que puede implementarse ya que abarca desde televisores, *smartphones*, relojes, autos, etcétera.

Windows Phone. Es un sistema operativo desarrollado por *Microsoft* enfocado a optimizar el consumo de aplicaciones y servicios para teléfonos inteligentes.

IOS. Es el sistema operativo desarrollado por *Apple* para ser utilizado en sus productos móviles, está enfocado principalmente en ofrecer una interfaz agradable al usuario a través de la cual se pueda acceder a todas las funcionalidades del dispositivo utilizando una pantalla táctil.

Palm OS. El sistema operativo pionero en hacer de las pantallas táctiles el medio de entrada principal de los primeros dispositivos móviles. Se trataba de un sistema operativo que ofrecía características sencillas, pero cotidianas para la vida empresarial como la integración de una agenda personal, calendario, notas e incluso galería de fotos y juegos sencillos en las últimas versiones.

Java ME. *Java Platform Micro Edition* ofrece una solución para aquellos dispositivos móviles de características limitadas, brindándoles un entorno flexible y sólido para la ejecución de aplicaciones *Java*. *Java ME* está presente en teléfonos móviles, reproductores multimedia, impresoras, etcétera.



Figura 1.10: Sistemas Operativos móviles. Fuente: impulsonegocios.com

1.3. Arquitecturas para aplicaciones móviles y envío de mensajes

Las arquitecturas utilizadas para las aplicaciones móviles están basadas en las diferentes características ofrecidas por los dispositivos móviles. Según la información con la que operan se pueden dividir de la siguiente manera:

Autocontenidas. La característica principal de esta arquitectura se basa en información que permanece estática a través del tiempo, es decir, esta arquitectura no requiere de conectividad a Internet o a algún tipo de fuente de información externo para operar. Ejemplo de una aplicación que sigue esta arquitectura es una calculadora, la cual no requiere información de Internet para funcionar correctamente.

Internet Inalámbrico. Esta arquitectura está comprendida por tres partes: un *micro-browser*, un servidor web y una base de datos. La primera realiza una petición a través de una URL para consumir algún servicio, la segunda parte escucha e interpreta las peticiones para poder enviar una respuesta y la tercera almacena la información que las aplicaciones requieren para funcionar. Ejemplo de una aplicación de esta arquitectura puede ser una cartelera de cine, la cual actualiza la información a través de peticiones a un servidor web y almacena las preferencias del usuario o realiza la compra de boletos en una base de datos remota.

Clientes Inteligentes. Esta arquitectura permite a las aplicaciones tener cierto grado de libertad respecto a la conectividad con la que cuente el dispositivo. Cuando no existe conexión las aplicaciones son capaces de almacenar la información de manera local, de esta forma pueden sincronizarse con el servidor posteriormente cuando la conectividad se establece. Ejemplo de una aplicación inteligente es una *app* de mensajería capaz de poner en lista de espera los mensajes que deseamos enviar cuando no existe conexión a Internet y de igual forma puede recuperar los mensajes que nos enviaron mientras estuvimos desconectados.

1.3.1. Envío de mensajes

El envío de mensajes a través de los dispositivos móviles, particularmente a través de los *smartphones*, ha sido uno de los principales motores en la mejora de las tecnologías que permiten la comunicación. A fin de ofrecer productos más competitivos, los fabricantes móviles han incluido y generado diferentes tecnologías para el intercambio de mensajes entre sus usuarios, este tipo de recursos que permiten la comunicación comúnmente son puestos a disposición de los clientes a través de una renta o suscripción a uno o varios servicios.

Email

El correo electrónico es un servicio que permite el intercambio de mensajes entre usuarios a través de la red. Su funcionamiento es similar al correo postal convencional, en donde el emisor redacta un mensaje de texto, adicionando a su consideración imágenes, documentos o cualquier otro tipo de archivos, posteriormente puede enviar el contenido del mensaje a una o varias personas utilizando la dirección electrónica de cada una, esta dirección es única para cada usuario del servicio de correo electrónico.

Personal Information Management (PIM)

Un gestor de información personal es un conjunto de servicios y/o aplicaciones que sirven para organizar la información cotidiana de una persona, así como herramientas que le permitan comunicarse con otros individuos. Actualmente, muchas de las herramientas de un PIM están alojadas en la nube como un conjunto de servicios puestos a disposición del usuario final, de esta forma es más fácil tener toda la información en cualquier lugar a través de diferentes dispositivos móviles que cuenten con una conexión a Internet. Las herramientas que forman parte de un PIM incluyen calendarios, notas, documentos, agendas, galerías de fotos, correo electrónico, etcétera.

SMS, MMS y mensajería sobre Internet.

El servicio SMS (*Short Message Service*) ha sido de los canales más populares de mensajería móvil, a través de este servicio es posible enviar y recibir mensajes de hasta 160 caracteres. Por otra parte, el Servicio MMS (*Multimedia Messaging Service*) permite además el envío de imágenes y videos cortos de hasta 300KB. A pesar de su popularidad y bajo costo, el uso de los servicios de envío de mensajes mencionados anteriormente ha ido decayendo gracias al surgimiento de aplicaciones móviles para el envío de mensajes a través de Internet ya que estas permiten el envío de texto, videos, documentos, fotografías y otros tipos de archivos de manera inmediata a un costo significativamente más bajo pues no requiere de la contratación explícita de un proveedor de servicio de mensajería sino que se vale del uso de datos a través de Internet.

1.3.2. Seguridad móvil

Creación de ambientes seguros

Debido a que muchos dispositivos móviles son parte del día a día se han convertido en una de nuestras herramientas principales para la generación y consumo de información, dicha información a menudo involucra datos personales o confidenciales, esto les ha costado a los dispositivos móviles convertirse en blancos para aquellas personas que buscan robar este tipo de información.

Es importante tomar las medidas necesarias para evitar que la información privada no sea utilizada inadecuadamente, es por ello que se deben generar ambientes seguros para el manejo de la información en los dispositivos móviles.

La primer y más importante barrera de seguridad debe ser levantada por el usuario, pues a pesar de que una aplicación móvil cuente con sistemas de seguridad muy avanzados, a final de cuentas los resultados dependerán en gran medida de que los usuarios móviles conozcan e implementen buenas prácticas para la protección de su información. Para evitar ser víctimas del robo de información algunas recomendaciones son las siguientes:

- Generar contraseñas fuertes.

- Desactivar conexiones inalámbricas cuando no se usan.
- Evitar conexiones wi-fi gratuitas.
- No confiar en aplicaciones de fuentes desconocidas.
- Instalar sistemas antirrobo para poder rastrear nuestro teléfono.
- No compartir información confidencial por medios no seguros.

Adicionalmente, la creación de ambientes seguros incluye:

- **Autenticación.** Comprobar que se es quién se dice ser.
- **Confidencialidad.** Aseguramiento de la privacidad.
- **Autorización.** Nivel de acceso a determinada información basado en el uso de credenciales.
- **No rechazo.** Asegura que los usuarios no puedan rechazar ciertas acciones identificando quien las realiza.

Hilos seguros

Con el uso del cómputo paralelo es cada vez más frecuente el desarrollo de aplicaciones móviles que utilizan la programación multihilo. Este tipo de programación proporciona una mayor velocidad en las tareas ejecutadas por varios hilos a la vez, éstos pueden acceder a la información de la memoria al mismo tiempo, es por ello que se debe asegurar que este acceso se realice de la manera más segura posible.

Para la creación de aplicaciones multihilo seguras, es importante considerar que múltiples hilos pueden acceder sólo a aquellos datos compartidos dentro de la aplicación mientras, que un bloque de código sólo puede ser accedido por un único hilo a la vez, de esta forma se evitan condiciones de carrera o sobre escritura de datos que puedan generar información errónea, incompleta o inexistente en los sistemas multihilo.

Tecnologías de seguridad

A medida que los sistemas móviles evolucionan también lo han hecho las diversas tecnologías de seguridad, como lo son:

Criptografía. Consiste en entablar comunicaciones entre dos puntos de manera tal que la información que fluye en un canal inseguro no pueda ser entendido por un tercero.

Encriptación de datos. Los datos encriptados aseguran la privacidad de datos sensibles mediante la transformación de la información en datos que no puedan ser entendidos.

Certificados digitales. Garantizan que una persona es quien dice ser.

Firmas digitales. Son utilizadas para verificar que la información proviene de un emisor confiable y que además los datos contenidos en el mensaje no han sido alterados desde su envío.

1.4. Construcción de aplicaciones de clientes inteligentes

1.4.1. Arquitectura Cliente-Servidor

La arquitectura Cliente-Servidor se basa en el consumo de servicios a través de un canal de comunicación, la construcción de aplicaciones de clientes inteligentes se apoyan en esta arquitectura, la cual realiza la mayor parte del procesamiento del lado del servidor haciendo más ligero el cómputo del lado del cliente. Es clara la preferencia de esta arquitectura para la construcción de aplicaciones de clientes inteligentes móviles, pues generalmente estos dispositivos cuentan con recursos de cómputo limitados, pero buscan ofrecer información actualizada a través de las aplicaciones que residen en ellos.

Las aplicaciones de cliente inteligente se distinguen porque consumen información del servidor para mostrar información al cliente, pero no necesariamente tienen que estar todo el tiempo *online* para operar correctamente, sino que realizan sus funciones con normalidad aún cuando no cuentan con conexión a Internet y envían los datos locales generados cuando se establece nuevamente la conectividad con el servidor.

Arquitectura del Cliente

El cliente inteligente está conformado por la interfaz de usuario, un módulo lógico y un módulo de almacenamiento de datos los cuales se describen a continuación:

Interfaz de Usuario. Es la forma en que la aplicación luce para los usuarios finales y además es una representación gráfica y amigable de lo que se capaz de realizar con la aplicación móvil.

Módulo lógico. Es el encargado de realizar las peticiones al servidor. En este módulo se define la lógica y el uso de protocolos para el consumo de servicios.

Módulo de almacenamiento de datos. A pesar de que gran parte de la información que utiliza una aplicación de cliente inteligente es por medio de peticiones al servidor es importante tener una fuente local de datos para ofrecer algunos servicios de manera *offline* o simplemente para almacenar información que es constante durante la vida de la aplicación móvil de cliente inteligente.

Arquitectura del Servidor

El servidor está formado generalmente por un módulo de sincronización y un módulo de almacenamiento de datos.

En el módulo de sincronización se realiza la recepción de peticiones y el envío de respuestas y es el intermediario entre el cliente y el almacenamiento de datos. El servidor cuenta con una API para ofrecer el acceso a los servicios ofrecidos por la aplicación móvil, además en el servidor se establece una capa de negocio en donde se definen aquellas necesidades propias del negocio que satisface un servicio determinado, por último, el servidor utiliza un adaptador que mapea una fuente de datos de manera que la información pueda ser consumida a través de un lenguaje de programación.

Por otra parte, el módulo de almacenamiento de datos es el encargado de organizar toda la información requerida por la aplicación móvil para operar correctamente.

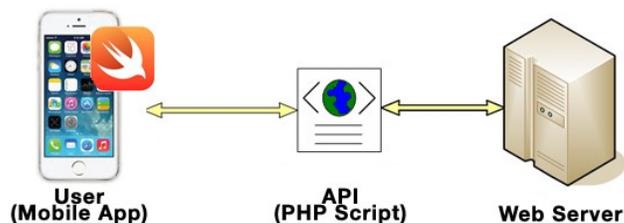


Figura 1.11: Arquitectura Cliente-Servidor. Fuente: brianjcoleman.com

1.4.2. Desarrollo de clientes inteligentes

Fase del análisis de requerimientos

El análisis de requerimientos es la fase más importante para la construcción de cualquier sistema, pues en esta etapa se definen aquellas necesidades que un usuario busca satisfacer mediante el uso del sistema.

Basados en la premisa de “el cliente no sabe lo que quiere”, es importante realizar una investigación detallada para averiguar qué es lo que el usuario realmente necesita y que tipo de solución puede resolver sus necesidades.

Fase de diseño

En esta etapa se definen las tecnologías que se van a utilizar en la construcción de la solución, el diseño preliminar de la interfaz de usuario y la planeación del proyecto. Es vital la comunicación continua con el usuario final para tener su visto bueno y sobre todo para que tenga una idea más aterrizada de lo que se espera de la aplicación a desarrollar.

Fase de implementación y pruebas

En esta etapa se ponen manos a la obra, se construye la aplicación basada en los requerimientos y el diseño previamente aceptados por el usuario. Es altamente recomendable que las pruebas de lo que se está construyendo acompañen en todo momento al desarrollo del software para evitar retrasos en el proyecto y para asegurar la calidad del software.

Fase de implantación

Es la última etapa en el desarrollo del *software*, en esta fase se pone en marcha el proyecto en el ambiente final y todas las características definidas al inicio del proyecto se ponen a disposición del usuario. En esta última parte del proyecto aún se pueden realizar pruebas y mantenimiento de la aplicación móvil.

1.5. Construcción de aplicaciones en Internet inalámbrica

1.5.1. Clientes Ligeros

Los clientes ligeros no requieren la instalación de ningún software a excepción de un navegador de Internet. Al igual que los clientes inteligentes, los clientes ligeros realizan peticiones a un servidor para la obtención de información, y la mayor parte de la lógica reside y es ejecutada del lado del servidor, de esta forma el cliente ligero obtiene los mensajes provenientes del servidor y los muestra al usuario final por medio de un lenguaje de marcado interpretado por el *browser*.

Cliente

El cliente está representado por la interfaz de usuario mostrada por el *browser*, en esta interfaz hace uso de algún lenguaje de marcado para la representación gráfica de la aplicación de cliente inteligente y en algunas ocasiones se incluye algún lenguaje de *script* para proveer de dinamismo al usuario final.

Middleware

En esta capa se encuentran los programas o módulos que les permiten a las aplicaciones comunicarse con los recursos de *hardware* presentes en el dispositivo. El *middleware* actúa como un traductor para permitir la comunicación y administración de datos en las aplicaciones.

Servicios de envío de mensajes

Las aplicaciones de cliente ligero requieren de servicios estandarizados para el envío de mensajes de tal forma que tanto el servidor como el cliente puedan enviar y recibir mensajes que puedan ser interpretados por ambas partes. Algunos marcos para el envío de mensajes son el Protocolo Simple de Acceso a Objetos (SOAP) y Transferencia de Estado Representacional (REST).

Desarrollo de clientes ligeros

El desarrollo de los clientes ligeros se caracteriza por la poca implementación lógica en su código, está basado principalmente en el consumo de servicios para la obtención, actualización, eliminación y creación de información en el servidor. Este debe basarse en un lenguaje de marcado que pueda ser interpretado por un explorador y además, al tratarse de clientes que se utilizan en las diferentes plataformas móviles, se deben desarrollar clientes ligeros capaces de adaptarse a las diferentes especificaciones de pantallas disponibles en el mercado, si bien no se puede cubrir el 100 % de los dispositivos disponibles se debe hacer un análisis previo para identificar aquellos dispositivos móviles con más posibles usuarios del sistema que utilizaran el cliente ligero.

Proceso de desarrollo

El ciclo de desarrollo de un cliente ligero es similar al de cualquier otro tipo de cliente, en primera instancia se debe hacer una recopilación de las características (requerimientos) que debe tener el cliente, en segundo lugar, se debe llevar a cabo un diseño tanto gráfico como de la arquitectura que se va a seguir, posteriormente la fase de desarrollo y pruebas en la que se construye el sistema por etapas las cuales son probadas continuamente para asegurar la calidad continua. Finalmente se lleva a cabo la implementación del cliente ligero para ponerlo a disposición de los usuarios finales.

Cabe mencionar que generalmente los desarrollos de cualquier sistema no terminan tajantemente con la entrega del producto final, sino que se mantienen las actividades mantenimiento en las que se realizan diferentes tipos de pruebas para encontrar defectos

o fallos y de esta forma resolverlos y mantener el sistema funcionando correctamente.

A continuación, se muestra un flujo común de desarrollo de clientes ligeros.

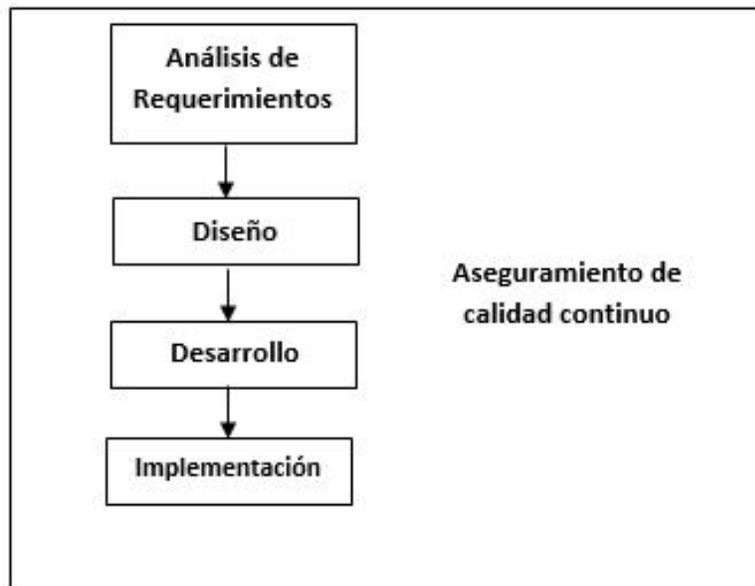


Figura 1.12: Ejemplo del desarrollo de un cliente ligero.

Modelos de aplicación

Hay dos principales modelos de aplicación para los clientes ligeros; la primera consiste en extender el acceso a un sitio web existente para el uso optimizado en dispositivos móviles, la segunda consta en el desarrollo de aplicaciones de clientes móviles ligeros para satisfacer necesidades de negocio específicas.

Extensión de aplicaciones Web existentes

Actualmente y ya desde hace unas décadas, existen innumerables sitios web en la red que se han mantenido funcionando a lo largo de los años, pero con el desarrollo de los nuevos dispositivos móviles ha sido importante incluir al sector de usuarios móviles en estas aplicaciones web tradicionales, una manera práctica para hacerlo es mediante la construcción de clientes inteligentes que únicamente consuman los servicios proporcionados por el sistema web existente, para ser provistos en los distintos dispositivos móviles. Lo anterior representa un beneficio muy grande para las empresas ya que no se debe desarrollar desde cero un sistema, sino que se trata de una extensión a las páginas web y por lo mismo el costo de implementar el sistema ligero es significativamente menor comparado con una implementación completa. Por otra parte, la implementación de un cliente ligero como parte de una extensión web es más práctica, rápida y compatible.

Clientes ligeros para nuevos nichos de negocio

Algunas veces no se puede simplemente extender una aplicación web existente para satisfacer todas las necesidades del negocio, sino que se deben crear clientes ligeros para satisfacer necesidades específicas. De esto trata el segundo modelo de aplicación de clientes ligeros, el cual parte desde cero para alcanzar nuevas metas específicas de negocio relacionadas con el uso de los dispositivos móviles, de esta forma se pueden desarrollar clientes a la medida y enfocados totalmente a la experiencia de usuario móvil.

Lenguajes y tecnologías para desarrollos en Internet inalámbrico

Lenguajes de marcado

Los clientes ligeros deben estar basados en lenguajes de marcado que puedan ser interpretados por los *microbrowsers*, estos lenguajes de marcado etiquetan el texto de manera tal que pueda ser procesado por los navegadores para mostrar de manera ordenada y estilizada la información en una página web. Los lenguajes de marcado más utilizados son HTML, XML y XAML.

HTML - *HyperText Markup Language*. Es el lenguaje utilizado por las páginas web para organizar la información contenida en ellas, ofrece funcionalidades para enlazar diferentes páginas web y hacer una navegación más interactiva.

XML - *eXtensible Markup Language* Es un lenguaje de marcado para estructurar información para que esta pueda ser almacenada, transmitida, procesada, visualizada e impresa, por muy diversos tipos de aplicaciones y dispositivos.

XAML - *eXtensible Application Markup Language* Es un lenguaje de marcado para la creación de interfaces gráficas de usuario para las plataformas *Windows Forms*, *Android*, *IOS* etcétera.

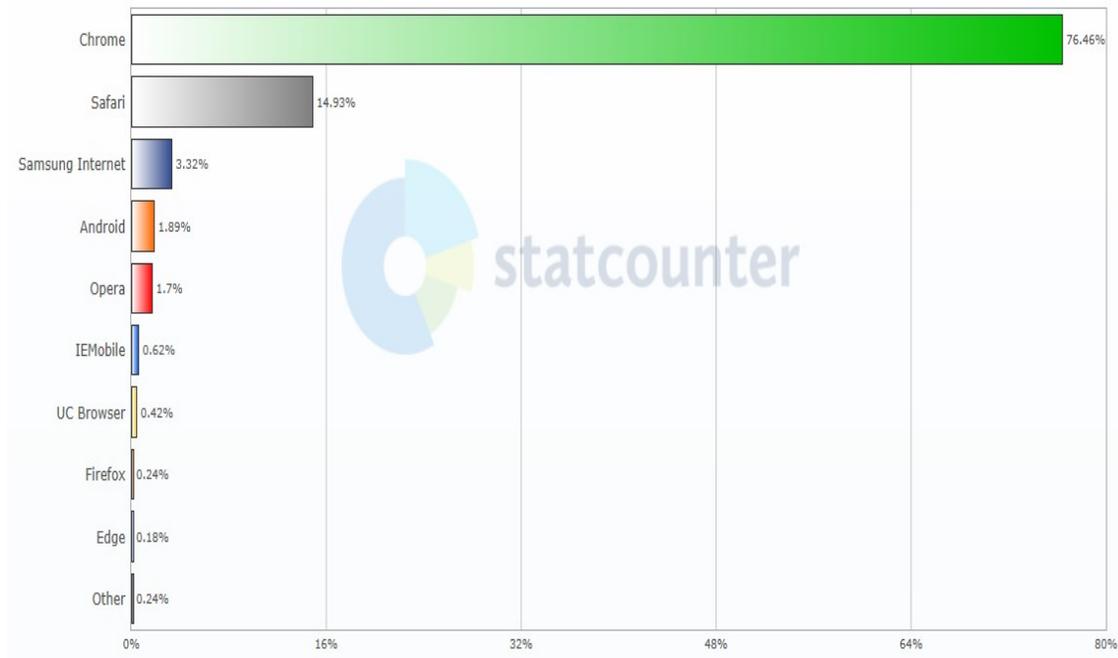


Figura 1.13: Uso de micronavegadores móviles en México (2018). Fuente: gs.statcounter.com

1.6. Administración de datos en sistemas móviles

1.6.1. Bases de datos móviles

Las bases de datos se han vuelto parte imprescindible para cualquier sistema informático y las aplicaciones móviles no son la excepción, pero a diferencia de las bases de datos tradicionales, las bases de datos móviles están optimizados para la administración de información en medios de almacenamiento limitados, el uso de poca memoria, bajo consumo de energía entre otras características propias de los dispositivos móviles.

Las bases de datos móviles más populares se muestran en la siguiente tabla.

Nombre	Desarrollador	Tipo de base de datos
SQLAnywhere	Sybase	Relacional
DB2 Everyplace	IBM	Relacional
SQL Server Compact	Microsoft	Relacional
SQLite	D'Richard Hipp	Librería en C
SQLBase	Grupta Technologies	Relacional

Tabla 1.3: Bases de datos móviles más populares. Fuente: CMAP Foundation Level Course.

1.6.2. Sincronización de bases de datos

El tipo de conexiones a bases de datos en las aplicaciones móviles se rigen por dos formas de sincronización de datos: almacenamiento-reenvío y sincronización continua, ambas pueden realizarse de manera síncrona o asíncrona.

Almacenamiento-reenvío

En este tipo de sincronización, el usuario móvil puede generar información en su dispositivo mientras no cuenta con conectividad al servidor remoto; de esta forma al establecerse la conexión, la información almacenada temporalmente de manera local se envía al servidor de base de datos.



Figura 1.14: Representación gráfica del almacenamiento-reenvío

Sincronización continua

La sincronización a bases de datos de manera continua requiere indispensablemente de una conexión estable con el servidor, dicha conexión puede ser síncrona o asíncrona pero no es posible almacenar información de manera local en el dispositivo móvil.



Figura 1.15: Representación gráfica de sincronización continua

Capítulo 2

Desarrollo

2.1. Fase de análisis de requerimientos

Los requerimientos de este proyecto se obtendrán a partir de historias de usuario y casos de uso, que describirán las actividades que puede realizar cada rol del Sistema de Apoyo para la Asignatura de Cómputo Móvil. Para la definición de las historias de usuario y el diseño de casos de uso se definen los siguientes actores: administrador, profesor y alumno.

2.1.1. Historias de usuario para rol Administrador

Rol: Administrador	ID: HU-ADMIN01
Como administrador	
Quiero una sección de “Administrar Profesores”	
Para ver la lista de todos los profesores del sistema	

Rol: Administrador	ID: HU-ADMIN02
Como administrador	
Quiero un botón	
Para crear nuevos profesores	

Rol: Administrador	ID: HU-ADMIN03
Como administrador	
Quiero un botón	
Para eliminar profesores	

Rol: Administrador	ID: HU-ADMIN04
Como administrador	
Quiero un botón	
Para editar profesores	

2.1.2. Historias de usuario para rol Profesor

Rol: Profesor	ID: HU-PROF01
Como profesor	
Quiero una sección de “Administrar Cursos”	
Para ver la lista de todos los cursos que tengo disponibles	

Rol: Profesor	ID: HU-PROF02
Como profesor	
Quiero un botón	
Para crear un nuevo curso	

Rol: Profesor	ID: HU-PROF03
Como profesor	
Quiero un botón	
Para eliminar un nuevo curso	

Rol: Profesor	ID: HU-PROF04
Como profesor	
Quiero un botón	
Para editar un curso	

Rol: Profesor	ID: HU-PROF05
Como profesor	
Quiero una sección de “Administrar Alumnos”	
Para ver la lista de todos los alumnos de un curso	

Rol: Profesor	ID: HU-PROF06
Como profesor	
Quiero una sección “Actividades”	
Para ver la lista de todas las actividades entregadas por los alumnos	

Rol: Profesor	ID: HU-PROF07
Como profesor	
Quiero un botón	
Para crear una actividad en un curso	

Rol: Profesor	ID: HU-PROF08
Como profesor	
Quiero un botón	
Para asignar calificación a cada tarea entregada por cada alumno	

Rol: Profesor	ID: HU-PROF09
Como profesor	
Quiero un botón	
Para editar calificación a cada tarea entregada por cada alumno	

Rol: Profesor	ID: HU-PROF10
Como profesor	
Quiero un botón	
Para eliminar una actividad de un curso	

Rol: Profesor	ID: HU-PROF11
Como profesor	
Quiero un botón	
Para editar una actividad de un curso	

Rol: Profesor	ID: HU-PROF12
Como profesor	
Quiero una sección	
Para ver la lista de todos los avisos creados por curso	

Rol: Profesor	ID: HU-PROF13
Como profesor	
Quiero un botón	
Para crear un aviso para un curso	

Rol: Profesor	ID: HU-PROF14
Como profesor	
Quiero un botón	
Para editar un aviso de un curso	

Rol: Profesor	ID: HU-PROF15
Como profesor	
Quiero un botón	
Para eliminar un aviso de un curso	

Rol: Profesor	ID: HU-PROF16
Como profesor	
Quiero una sección de “Calificaciones”	
Para ver todas las calificaciones de los alumnos	

Rol: Profesor	ID: HU-PROF17
Como profesor	
Quiero un botón	
Para asignar la calificación final a un alumno	

Rol: Profesor	ID: HU-PROF18
Como profesor	
Quiero un botón	
Para editar la calificación final a un alumno	

2.1.3. Historias de usuario para rol Alumno

Rol: Alumno	ID: HU-ALUM01
Como alumno	
Quiero una sección para consultar el temario de la materia	
Para tener acceso al material teórico del curso	

Rol: Alumno	ID: HU-ALUM02
Como alumno	
Quiero una sección de “Cursos”	
Para ver la lista los cursos a los que me podría inscribir	

Rol: Alumno	ID: HU-ALUM03
Como alumno	
Quiero un botón	
Para inscribirme a un curso	

Rol: Alumno	ID: HU-ALUM04
Como alumno	
Quiero una sección “Actividades”	
Para ver la lista de actividades creadas por el profesor en el curso en el que estoy inscrito	

Rol: Alumno	ID: HU-ALUM05
Como alumno	
Quiero una sección “Enviar Tarea”	
Para enviar una tarea relacionada a cada actividad	

Rol: Alumno	ID: HU-ALUM06
Como alumno	
Quiero una sección “Avisos”	
Para ver los avisos del curso al que estoy inscrito	

Rol: Alumno	ID: HU-ALUM07
Como alumno	
Quiero una sección “Calificaciones”	
Para ver mis calificaciones del curso en el que estoy inscrito	

Rol: Alumno	ID: HU-ALUM08
Como alumno	
Quiero una sección “Perfil”	
Para ver y modificar mis datos de perfil de usuario	

2.1.4. Diagramas de Casos de Uso

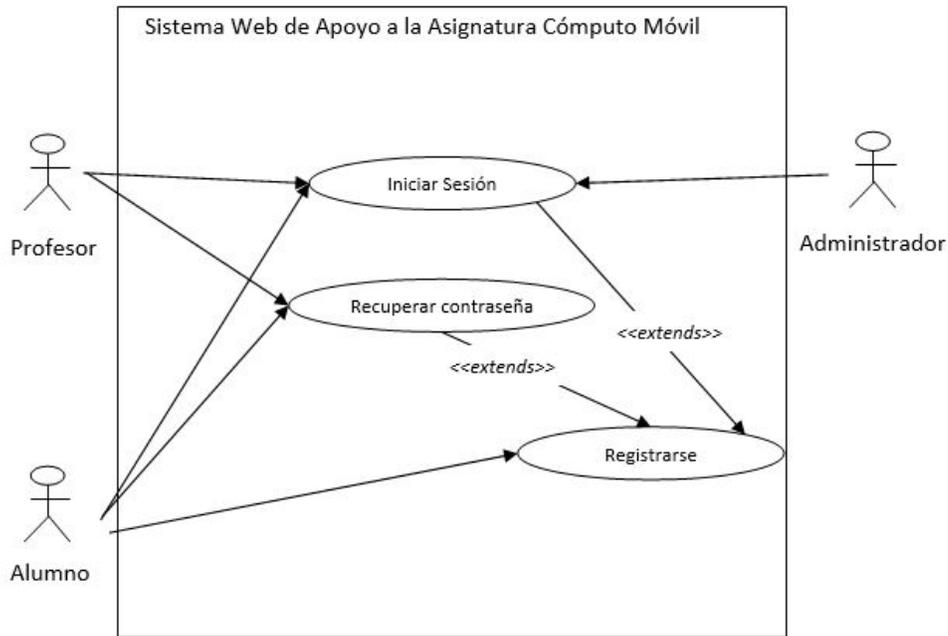


Figura 2.1: Diagrama de Caso de Uso de inicio de sesión y registro

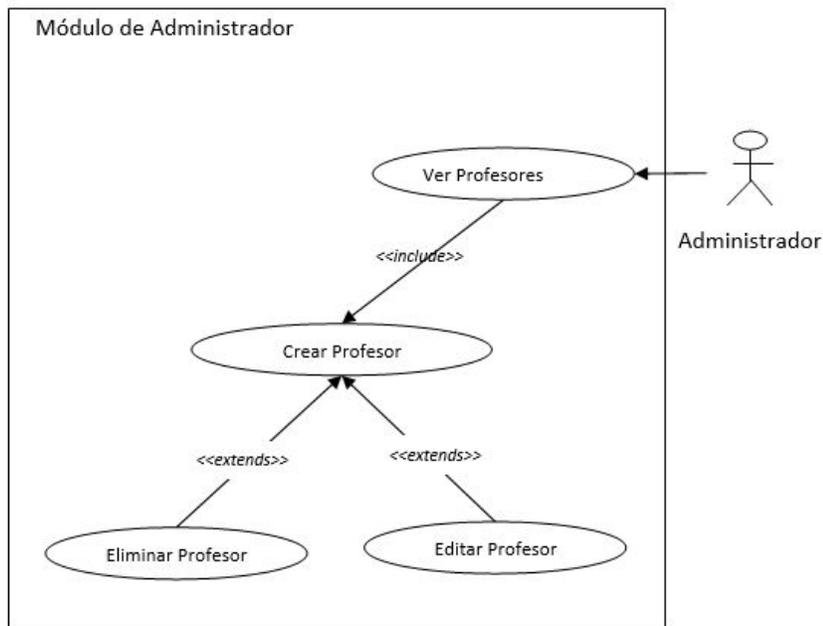


Figura 2.2: Diagrama de Caso de Uso del módulo de Administrador

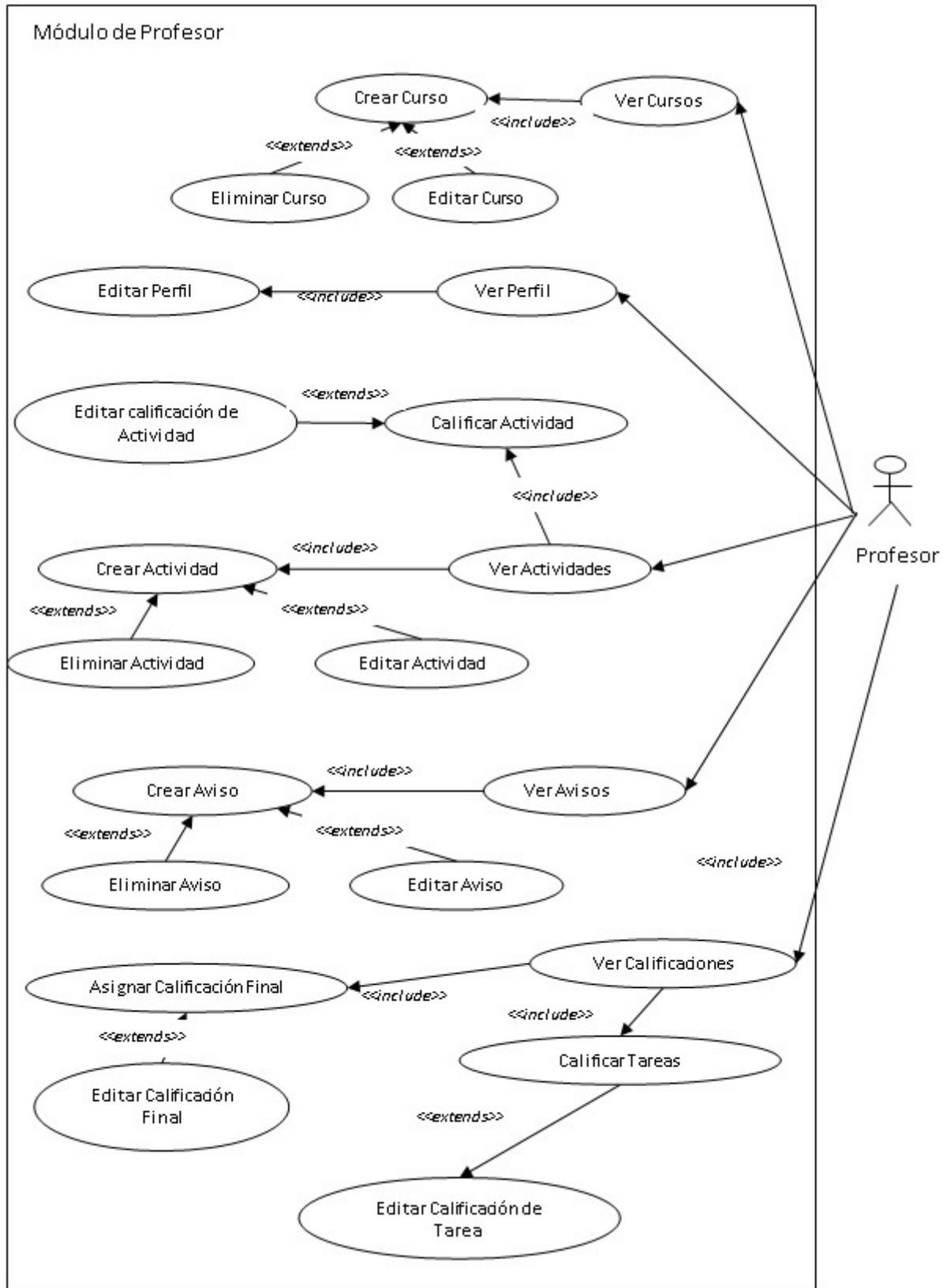


Figura 2.3: Diagrama de Caso de Uso del módulo de Profesor

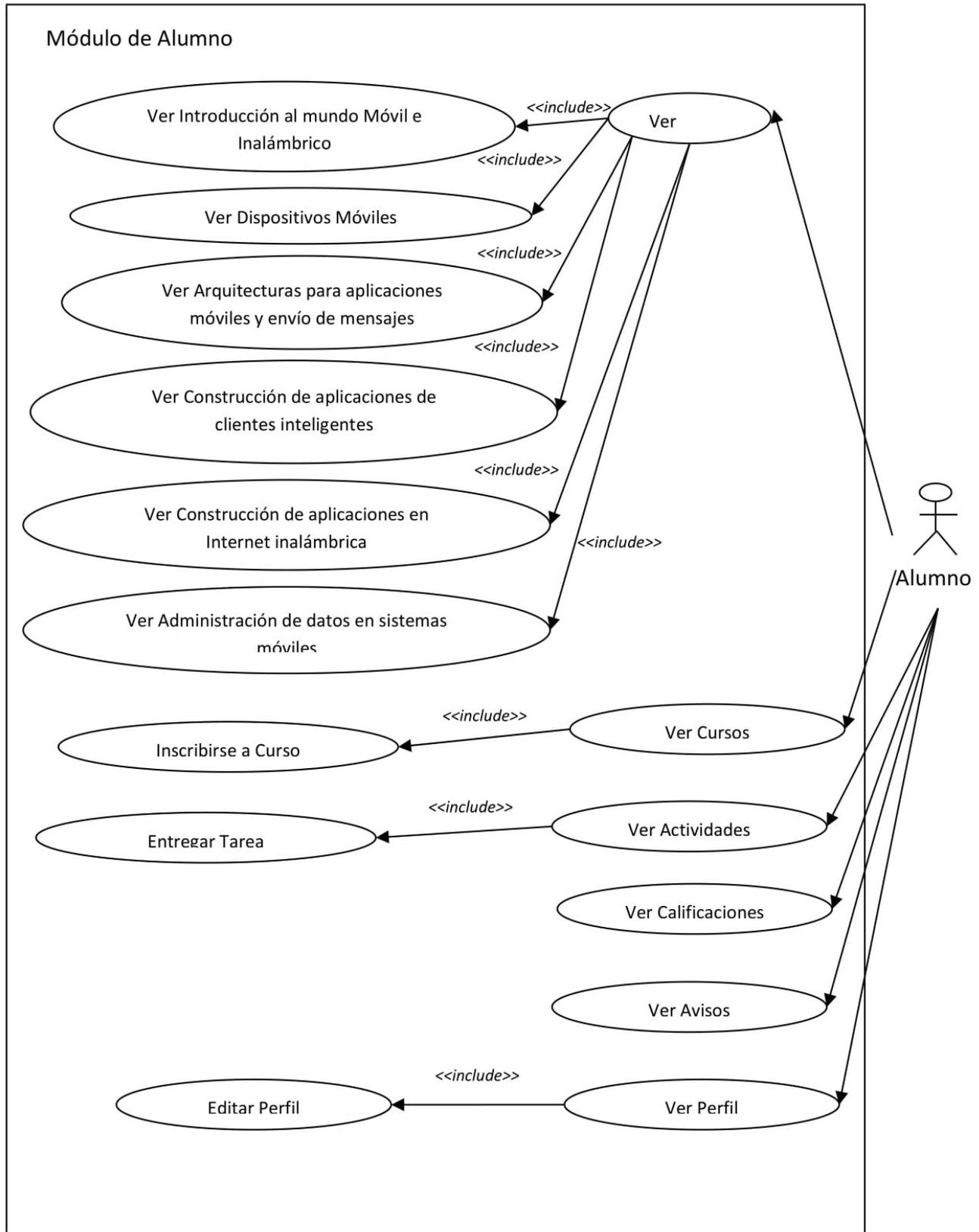


Figura 2.4: Diagrama de Caso de Uso del módulo de Alumno

2.2. Fase de diseño y construcción

La arquitectura del proyecto es de tipo multicapa o de tres niveles, cada una de las capas tiene una función específica dividiendo así la carga de trabajo y las responsabilidades de cada una. A continuación, se describe cada una de las capas que conforman al proyecto:

Capa de Acceso a Datos. En ella vive el almacén de datos, una instancia de base de datos sobre *SQL Server* diseñada para soportar la aplicación web. Esta capa se desplegará en la plataforma de *Microsoft Azure*.

Capa de *Business Logic*. La capa de negocio sirve de intermedio entre la Capa de Acceso a Datos y la Capa de Presentación, su función es la de recibir, ejecutar y responder a las peticiones de los usuarios respetando siempre las reglas establecidas por el negocio. Aunque sus principales objetivos son obtener, eliminar, actualizar y agregar información a la Capa de Acceso a Datos, las funciones que puede desempeñar podrían ser meramente lógicas sin tener que recurrir al almacén de datos. Al Igual que la Capa de Acceso a Datos, esta capa residirá sobre *Microsoft Azure* utilizando una *Web App* para su despliegue.

Para fines del proyecto, se desarrollará una *Web API REST* que trabajará con peticiones y respuestas de tipo JSON.

Capa de Presentación. Esta capa es la encargada de mostrar al usuario una interfaz fácil de utilizar para llevar a cabo las funciones especificadas durante la fase de análisis de requerimientos.

La Capa de Presentación nunca tiene acceso directo a la Capa de Acceso a Datos, siempre utiliza el puente de comunicación provista por la Capa de *Business Logic*.

El despliegue de este nivel será sobre un servidor web independiente al resto de capas de la arquitectura.

Cómo se puede observar, la arquitectura planteada anteriormente supone como ventajas una menor dependencia para operar entre cada capa, un mejor nivel de mantenibilidad y mayor seguridad. Sin embargo, la arquitectura también presenta desventajas y la más grande de ellas es su alto costo frente a arquitecturas de una o dos capas.

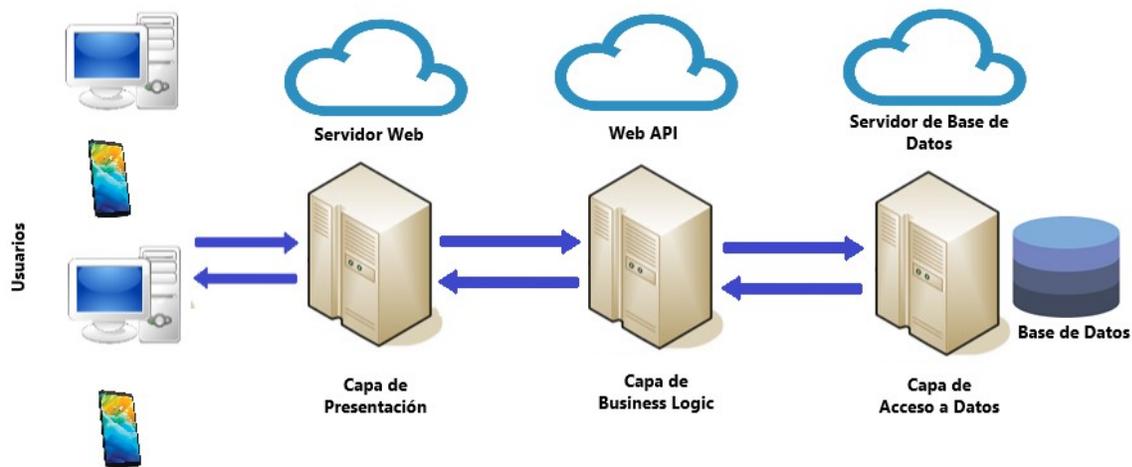


Figura 2.5: Representación gráfica la arquitectura del proyecto.

2.2.1. Diseño de la Base de Datos

La Base de Datos es el soporte fundamental que da vida a la aplicación web, en este apartado se muestra el diseño de la Base de Datos relacional a utilizar para el aplicativo. Se construirá utilizando *SQL Server* de *Microsoft*, el cuál es un servidor de bases de datos relacionales de fácil integración con la plataforma en la nube de la misma compañía, así como con el resto de tecnologías utilizadas para la construcción de la plataforma web. A continuación, se describe cada una de las tablas que componen a la Base de Datos.

Tablas

Tipo Usuarios: Catálogo para almacenar el tipo de usuarios que pueden acceder al sistema.

- 1 Administrador
- 2 Profesor
- 3 Alumno

El identificador de cada rol no puede ser modificado ya que la *Capa de Business Logic* utiliza esta tabla para identificar a cada usuario que accede al sistema.

Administradores: Contiene un catálogo de los administradores del sistema, por defecto existe un único administrador con el usuario *computomovil* y contraseña conocida.

Profesores: Contiene la definición de propiedades de los profesores, como llave primaria se utiliza el RFC del profesor en formato alfanumérico.

Alumnos: Contiene la definición de propiedades de cada alumno inscrito en el sistema web, como llave primaria se utiliza el número de cuenta escolar en formato numérico.

Actividades: Contiene la definición de actividades creadas por cada uno de los profesores, es decir las actividades que el profesor solicita a los alumnos inscritos en alguno de sus cursos.

Tareas: Contiene las tareas que los alumnos crean en respuesta a una actividad creada por el profesor del curso al que están inscritos.

Cursos: Contiene cada uno de los cursos dados de alta en el sistema por los profesores, cuenta con un campo de contraseña definido por el profesor que lo crea y que sirve para que los alumnos puedan inscribirse, cabe mencionar que los alumnos únicamente pueden estar inscritos a un curso a la vez ya que se trata de una plataforma dedicada exclusivamente a la materia de *Cómputo Móvil* por lo cual está pensada para que sirva como complemento de cada curso dentro de los horarios establecidos cada semestre por la Facultad.

Avisos: Esta tabla contiene los avisos que crean los profesores para cada uno de los cursos que han creado en la plataforma.

Modelo Relacional

En la siguiente figura se muestra el diagrama de la Base de Datos diseñada para la plataforma en el cuál se puede ver a detalle los tipos de dato utilizados en cada una de las tablas y su relación entre ellas.

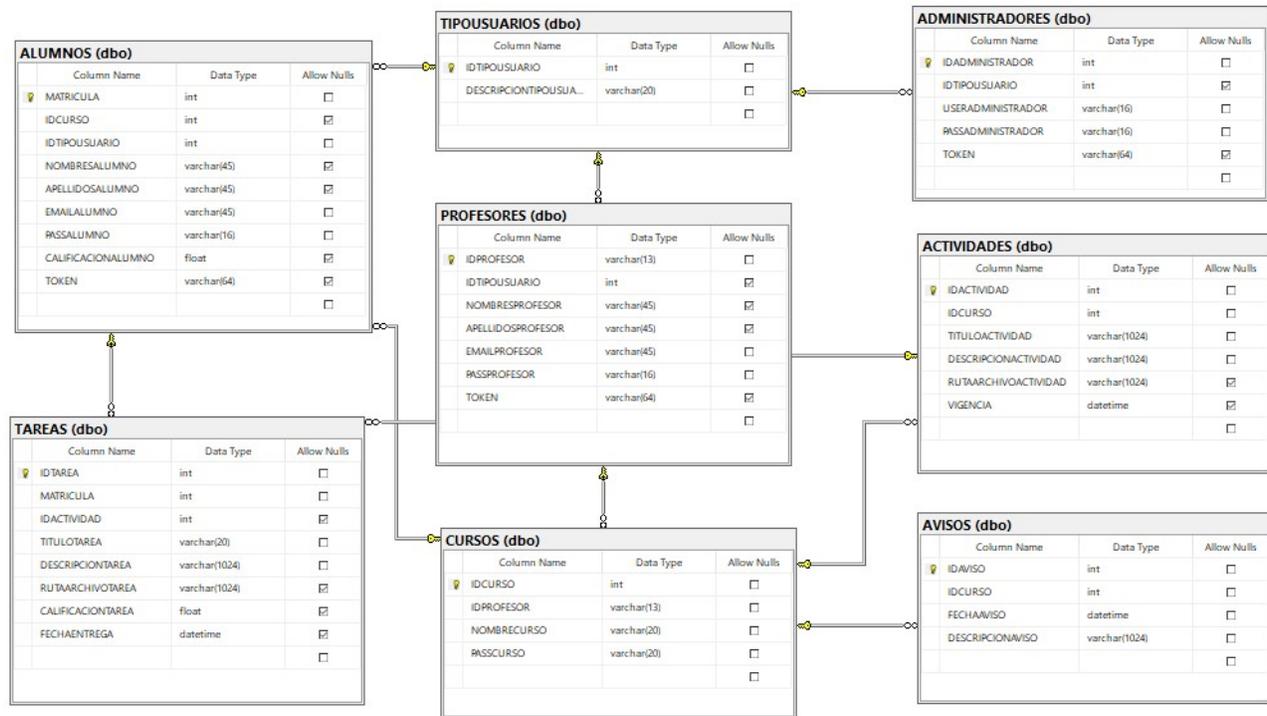


Figura 2.6: Modelo relacional de la Base de Datos.

2.2.2. Diseño del Web API REST

Un servicio de tipo API REST se basa en el protocolo HTTP para recibir peticiones y enviar datos, particularmente para la plataforma web se emplearán respuestas en formato JSON (*JavaScript Object Notation*) por su facilidad de implementación e interpretación tanto del lado del servidor API como del lado del Cliente.

La solución de este servicio está basada en el lenguaje de programación C#, así mismo se utiliza *Entity Framework* para realizar el mapeo de las entidades a las tablas en la Base de Datos *SQL Server*.

Cómo se explicó anteriormente, este servidor API sirve como intermediario entre el Acceso a Datos y la Capa de Presentación obedeciendo a las reglas de negocio definidas en la fase de análisis de requerimientos por lo que el servidor API se vuelve una parte esencial del desarrollo.

A fin de realizar cambios durante la construcción de la API se ha incluido la solución en un repositorio de control de versiones, de esta manera cada nueva sección de código o

corrección del mismo se subirá al repositorio evitando así diferentes versiones y teniendo como resultado una única fuente confiable para obtener la última versión del código. La información relacionada con el repositorio de control de versiones se puede consultar en el apéndice “Repositorio de código”.

Estructura de la solución en *Visual Studio*

Por el lado de la estructura de la solución del Web API, utilizando el entorno de desarrollo *Visual Studio* nos encontramos una estructura definida como se muestra en la siguiente figura.

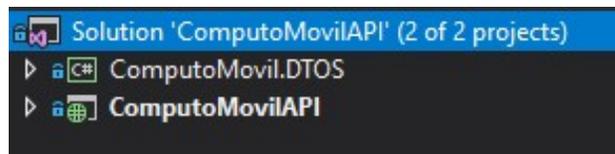


Figura 2.7: Estructura de la solución API.

Como se puede observar, la solución consta de dos proyectos. El primero está formado por las clases en C# que son un reflejo de los datos que se mostrarán en la interfaz del usuario, en otras palabras, `ComputoMovil.DTOS` contiene la definición del tipo de respuestas que entregará el servidor API. Este proyecto está referenciado dentro de `ComputoMovilAPI`, el cual es el proyecto principal y contiene toda la lógica de negocio, de él se hablará más adelante.

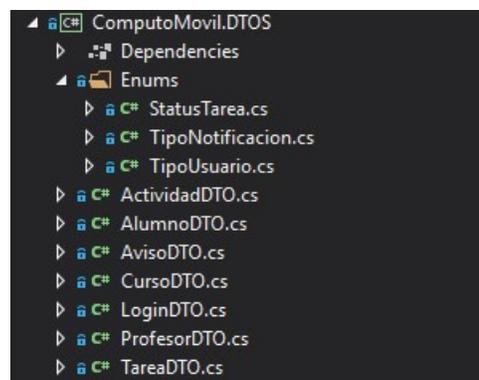


Figura 2.8: Estructura del proyecto `ComputoMovil.DTOS`.

La carpeta *Enums* contiene tres enumeradores:

- StatusTarea
- TipoNotificación
- Tipo usuario

La razón de ser de estos enumeradores es la de evitar el *hardcoding* o “código duro” dentro de la solución del web API.

```
1 using System.ComponentModel;
2
3 namespace ComputoMovil.DTOS.Enums
4 {
5     public enum TipoUsuario
6     {
7         [Description("ADMINISTRADOR")]
8         Administrador = 1,
9         [Description("PROFESOR")]
10        Profesor = 2,
11        [Description("ALUMNO")]
12        Alumno = 3
13    }
14 }
```

Figura 2.9: Ejemplo de Enum.

Por su parte, cada una de las clases con el sufijo *DTO*, son un reflejo de lo que el cliente desea recibir. Se muestra un ejemplo.

```
1 namespace ComputoMovil.DTOS
2 {
3     public class LoginDTO
4     {
5         public string user { get; set; }
6         public string password { get; set; }
7         public string token { get; set; }
8         public int tipoUsuario { get; set; }
9     }
10 }
```

Figura 2.10: Ejemplo de clase DTO.

En el ejemplo mostrado en la figura anterior se observa la estructura típica de una clase *DTO*, como se puede observar, esta clase contiene aquello que se desea enviar como respuesta al cliente a través de una conversión de un *DTO* a una respuesta JSON.

Por su parte, el proyecto principal, *ComputoMovilAPI* contiene toda la lógica de negocio para brindar respuestas a los clientes. Su estructura se muestra en la siguiente figura.

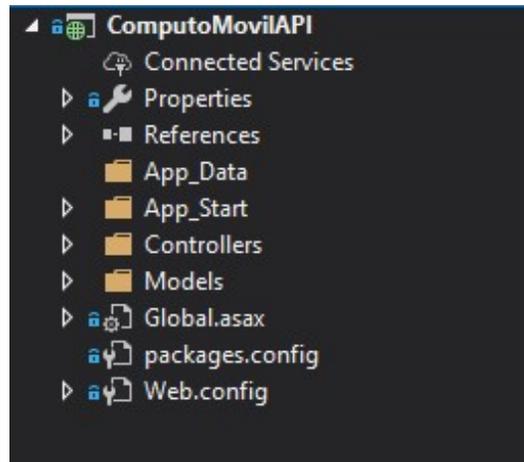


Figura 2.11: Estructura del proyecto *ComputoMovilAPI*.

La carpeta llamada *Controllers* contiene la definición de los servicios disponibles en el API, en estas clases de tipo *Controller* se utiliza una conexión al servidor de Base de Datos para obtener y enviar información para posteriormente enviar una respuesta al cliente.

Cada una de las clases *Controller* contiene la lógica de negocio especificadas en la fase de análisis de requerimientos, por ejemplo, la clase *TareasController* contiene un método llamado *PostTarea* el cuál es el servicio que llama la aplicación web para realizar el alta de una nueva tarea en la plataforma web.

PostTarea retorna *HttpResponseMessage*, dicho tipo de retorno especifica el estatus de la petición que se realizó y puede incluir también un mensaje en formato JSON para ser interpretado por la plataforma web y así poder mostrar información al usuario.

Cómo se puede observar en la siguiente figura, el tipo de retorno es a su vez *Task* esto se debe a que se incluye en el método *PostTarea* un procedimiento asíncrono para el envío de notificaciones.

```
[EnableCors(origins: "*", headers: "*", methods: "*")]
[Route("api/tareas/{idActividad}/{token}")]
public async Task<HttpResponseMessage> PostTarea(int idActividad, string token, TareaDTO tareaDTO)
{
    var usuario = Utilerias.ValidarUsuario(token);
    var idAlumno = Utilerias.GetUserByToken(token);
    int matricula;

    int.TryParse(idAlumno, out matricula);

    var actividad = db.ACTIVIDADES.Find(idActividad);

    if (usuario == (int)TipoUsuario.Alumno)
    {
        tareaDTO.IDACTIVIDAD = idActividad;
        var tarea = MapTareaDTotoEntity(tareaDTO);

        tarea.MATRICULA = matricula;
        tarea.IDACTIVIDAD = idActividad;
        db.TAREAS.Add(tarea);
        db.SaveChanges();

        Notificaciones
        return Request.CreateResponse(HttpStatusCode.OK);
    }
    return Request.CreateResponse(HttpStatusCode.Forbidden);
}
```

Figura 2.12: Ejemplo de servicio API.

La siguiente carpeta de interés dentro del proyecto *ComputoMovilAPI* es la llamada *Models*, dicha carpeta contiene el mapeo (entidades) de la Base de Datos a una clase común de C#, este mapeo es utilizado por *Entity Framework* para representar los datos de la Base de Datos utilizando programación en C#.

Seguridad del Web API

Para que el acceso a los servicios que provee el API sea seguro, se ha implementado un método de generación de *token*, este *token* es generado automáticamente en el momento que un usuario inicia sesión y se guarda en la Base de Datos, de esta forma cuando un usuario intenta acceder a un servicio al que sólo pueden acceder los profesores, se envía el *token* y se valida que pertenezca al rol de profesor, si no es así se envía un estatus *Forbidden* o una respuesta nula al cliente, evitando así el uso mal intencionado del web API.

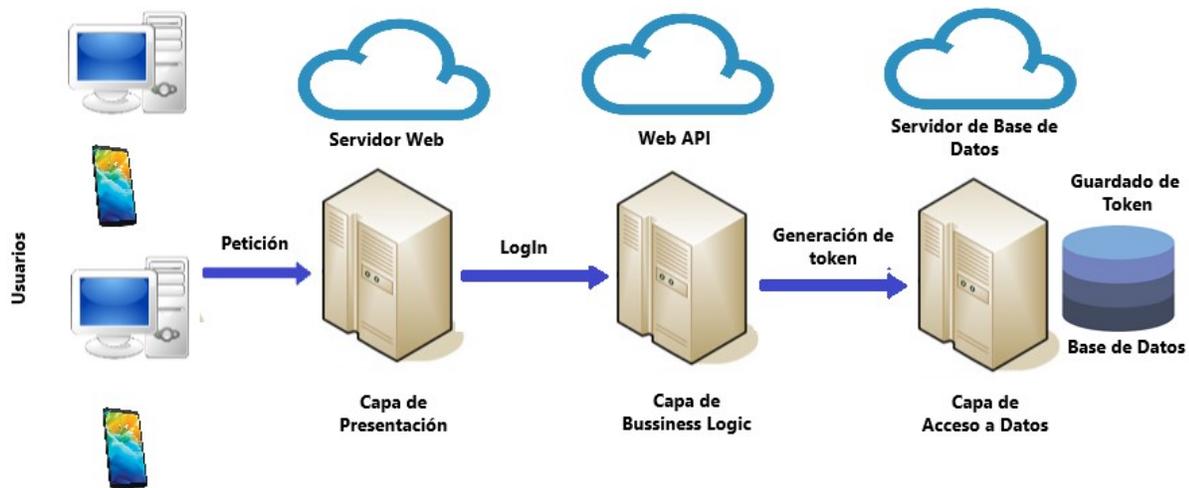


Figura 2.13: Proceso de generación de token al iniciar sesión.

2.2.3. Diseño de la Capa de Presentación

El proyecto correspondiente a la Capa de Presentación está basado en las tecnologías HTML, CSS, el *Framework Bootstrap*, JavaScript junto con la librería *JQuery* para lograr un ambiente amigable para el usuario.

El esqueleto de interfaz de usuario está construido en HTML y estilizado con CSS utilizando *Bootstrap* para lograr una plataforma web *responsive* (o adaptativa) para poder ser visualizado cómodamente en dispositivos móviles o de pequeño tamaño.

Gracias a las peticiones asíncronas realizadas con la librería *JQuery* que se realizan del lado del cliente al servidor Web API, es posible enviar una petición y posteriormente mostrar en pantalla la respuesta construyendo la vista dinámicamente.

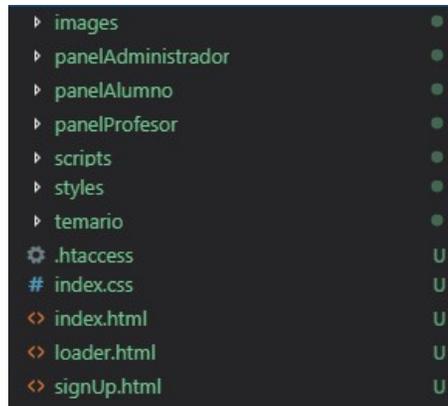


Figura 2.14: Estructura de la Capa de Presentación.

Cada una de las carpetas panelAdministrador, panelAlumno y panelProfesor, contienen los archivos HTML necesarios para mostrar la interfaz de cada usuario dependiendo su rol en el sistema.

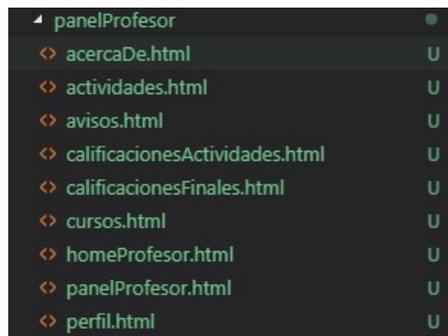


Figura 2.15: Ejemplo de interfaz de Profesor.

Cada panel de rol en este proyecto consta de una plantilla HTML base, en el ejemplo del panelProfesor, dicha plantilla es la denominada panelProfesor.html en la cual se van incrustando, según sea requerido, el resto de vistas HTML de esta forma y en conjunto con las peticiones asíncronas se consigue un efecto SPA (*Single Page Application*), ya que cada panel nunca es recargado, sino que espera una respuesta del servidor para poder dibujar en la plantilla la respuesta recibida.

2.3. Fase de implementación y pruebas

Para cada una de las capas se empleó un método de despliegue a producción distinto según las características particulares que presentan, sin embargo, cada una de ellas fue desarrollada de manera local en primera instancia y posteriormente llevada a producción como se describe para cada caso.

2.3.1. Preparación del Grupo de Recursos en *Microsoft Azure*

Una vez se dio por finalizada la fase de desarrollo se creó una suscripción *Azure* y un grupo de recursos en el cual se aloja tanto el servidor de Base de Datos como el *Web App* donde vive el Web API.

Cabe mencionar que, aunque ambos servicios forman parte del mismo grupo de recursos en *Azure* cada uno de ellos es independiente e incluso se podrían encontrar en diferentes áreas geográficas.

<input type="checkbox"/> NOMBRE ↑↓	SUSCRIPCIÓN ↑↓	UBICACIÓN ↑↓
<input type="checkbox"/>  Tesis	SuscripciónTesis	Centro de EE. UU.

4 elementos <input type="checkbox"/> Mostrar tipos ocultos			
<input type="checkbox"/> NOMBRE ↑↓	TIPO ↑↓	UBICACIÓN ↑↓	
<input type="checkbox"/>  computo-movil-dev	App Service	Centro-Sur de EE. UU.	***
<input type="checkbox"/>  computo-movil-server	SQL Server	Centro-Sur de EE. UU.	***
<input type="checkbox"/>  computomovilDB (computo-movil-s...	Base de datos SQL	Centro-Sur de EE. UU.	***
<input type="checkbox"/>  PlanGratisF1	Plan de App Service	Centro-Sur de EE. UU.	***

Figura 2.16: Grupo de recursos en Microsoft Azure.

Implementación de la Base de Datos

Se diseñó y construyó la Base de Datos utilizando *Transact SQL* y *SQL Server Management Studio*, durante la fase de desarrollo la Base de Datos residió de manera local. Tras haber realizado cambios y ajustado algunas propiedades de ella mientras se estuvo

en la fase de desarrollo, se procedió a migrar la Base de Datos a un servidor en la nube de *Microsoft Azure*.

Implementación del Web API

Al igual que la Base de Datos, el Web API se trabajó durante la fase de desarrollo sobre el entorno local. Al desplegar a producción el servicio web, se creó un *App Service* dentro del grupo de recursos creados para este proyecto en *Azure*.

Al utilizar *Visual Studio* como entorno de desarrollo para el Web API, el despliegue hacia un *App Service* se hace de manera muy sencilla vinculando la cuenta propietaria del grupo de recursos *Azure* en el IDE de *Visual Studio*, de esta manera el entorno de desarrollo obtiene los grupos de recursos vinculados a la cuenta y se puede proceder a desplegar el desarrollo al *App Service*.

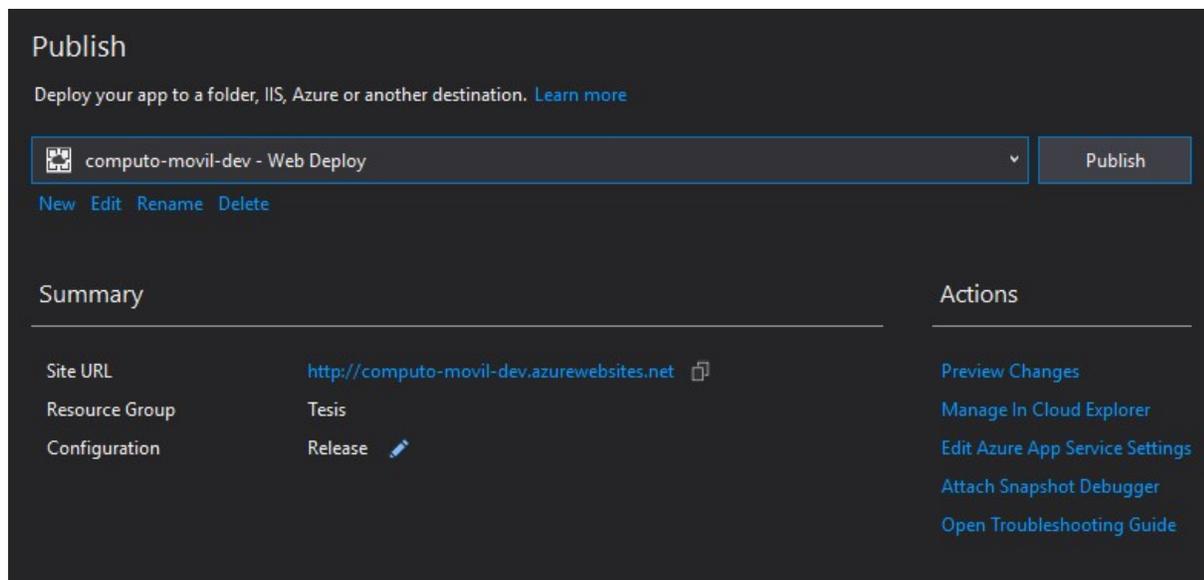


Figura 2.17: Perfil de publicación en Visual Studio para Azure App Service.

Implementación de Presentación

Para la Capa de presentación se trabajó en la fase de desarrollo sobre un servidor web local (servidor *WampServer*) y posteriormente se trasladaron todos los archivos a un servidor web en Internet, esta transferencia de archivos se llevó a cabo utilizando el

protocolo FTP (*File Transfer Protocol*) empleando el *software FileZilla* para transferir los archivos de la Capa de Presentación al servidor web en Internet.

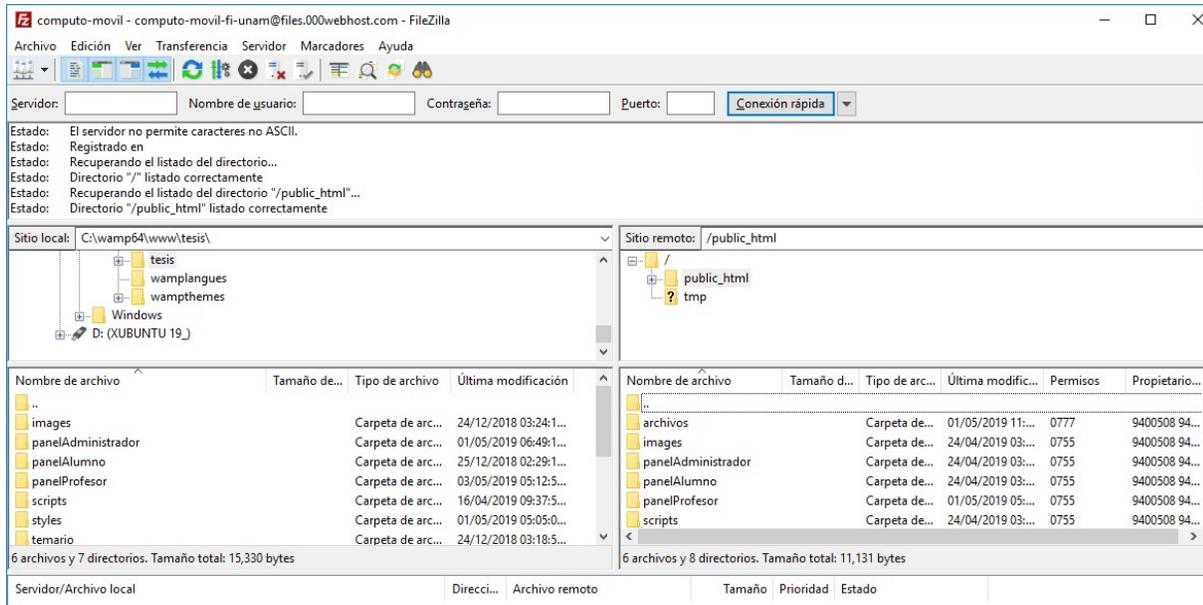


Figura 2.18: Transferencia de archivos de ambiente de desarrollo a ambiente de producción.

Capítulo 3

Pruebas de la plataforma

Para validar el sistema se realizó un mapeo con las historias de usuario y de esta manera asegurar que el sistema satisfaga cada una de las necesidades planteadas en ellas. Además, se agregaron algunos flujos alternos a los casos de prueba para validar el funcionamiento de la plataforma web. De esta forma se produjeron los entregables expuestos en el apéndice “Casos de Prueba” misma sección donde se podrá consultar la evidencia de su ejecución.

Rol: Administrador	ID: CP-ADMIN01
Objetivo:	Iniciar sesión con rol Administrador
Descripción:	Iniciar sesión con un usuario y contraseña válidos para el rol de Administrador
Resultado esperado:	Al iniciar sesión correctamente el sistema redireccionará al usuario al panel de Administrador
Evidencia:	 A screenshot of a mobile web browser displaying the administrator dashboard. The browser's address bar shows the URL: https://computo-movil-fi-unam.000webhostapp.com/panelAdministrador/panelAdministrador.html. The page has a decorative header with a pattern of gears and the text 'Cómputo Móvil'. Below the header, there is a red button labeled 'Menú' and a large white box with the text 'Bienvenido a la página de Administrador'.

Tabla 3.1: Ejemplo de Caso de Prueba.

La importancia de la ejecución de estos casos de prueba es vital para poder entregar un producto de software que cumpla con las expectativas del cliente y reducir el riesgo de problemas en producción ya que a medida que el ciclo de vida del software avanza, el impacto y costo de encontrar *bugs* en el sistema crece.

Cómo resultado obtenido después de hacer las pruebas y hacer correcciones durante

la fase de desarrollo, se obtuvo un sistema estable que cumple con los requerimientos del negocio. No obstante, es recomendable realizar una liberación *beta* a manera de pruebas UAT (*User Acceptance Tests*), de esta manera se puede obtener el visto bueno del cliente para proceder a realizar una liberación a producción de manera más confiable.

Capítulo 4

Resultados

Se obtuvo una aplicación web que satisface los problemas planteados en este escrito, por una parte la plataforma web contiene un apartado donde se pueden repasar los temas relacionados con la asignatura Cómputo Móvil sin embargo se detectó que una mejor funcionalidad para esta sección sería que los tópicos puedan ser editados libremente por cada uno de los profesores dotando a la plataforma de un dinamismo que permita ajustarse más a las necesidades de cada uno de los cursos impartidos en la Facultad para esta materia.

La siguiente sección que se logro fue la de la administración de los cursos del docente y no únicamente eso, sino que la aplicación se dividió en 3 módulos para dotar a la plataforma de mayor funcionalidad.

La administración por parte de los docentes es el corazón del sistema web, sin embargo, para los alumnos también es importante llevar un registro de su avance en el curso y estar pendiente de aquellos eventos que se produzcan en la plataforma y que sean de su interés. Para satisfacer esta característica se decidió incluir un servicio de notificaciones vía correo electrónico para mantener informados tanto a profesores como alumnos. De esta manera, el sistema notifica a los alumnos cuando un profesor crea una nueva actividad o un aviso y también notifica a los profesores cuando un alumno responde a alguna de sus actividades con la entrega de una tarea. En esta sección también se han identificado áreas de oportunidad que podrían incluirse en liberaciones posteriores. Ejemplo de esto sería bueno incluir una sección de comentarios para poder tener un canal de comunicación más directo y dinámico dentro de la plataforma entre docentes y alumnos.

Finalmente se puede comentar que se obtuvo como resultado una página web amigable, adaptativa y funcional para los fines que se plantearon en un inicio, así mismo la arquitectura de esta solución permitirá una mejor mantenibilidad y escalabilidad del proyecto en futuros cambios o correcciones que se lleguen a solicitar por parte de los usuarios.

Capítulo 5

Conclusiones

A pesar de que el sistema cubre todos los requerimientos expuestos en el análisis de requerimientos, a medida que se fue avanzando en el desarrollo se pudieron identificar áreas de oportunidad en la plataforma, mismas que el cliente podrá considerar incluir en una liberación posterior.

Como experiencia personal puedo afirmar que el desarrollo de este proyecto ha integrado muchos de los temas vistos durante la carrera, debo decir que elegí particularmente el desarrollar este trabajo con la finalidad de poner en práctica los conocimientos aprendidos durante mi estancia en la Facultad de Ingeniería, así como aprender nuevas tecnologías que me sean de utilidad dentro de mi vida profesional.

Al iniciar con el desarrollo de este proyecto no tenía la práctica suficiente en programación para poder concluirlo satisfactoriamente, sin embargo, con el paso de los meses pude mejorar mis *skills* técnicos lo que cada vez me facilitó más el trabajo de construcción pero también me hicieron conocer nuevas y mejores formas de llevar a cabo un desarrollo de este tipo.

Capítulo 6

Apéndices

6.1. Repositorio de Código

El código de la solución .NET se encuentra disponible a través de la plataforma *GitHub*. De esta forma, al realizar modificaciones al código del proyecto de Web API, es posible hacerlo de manera organizada y segura puesto que si por algún motivo el proyecto posee errores en versiones más recientes, basta con regresar a un estado funcional y depurar la aplicación.

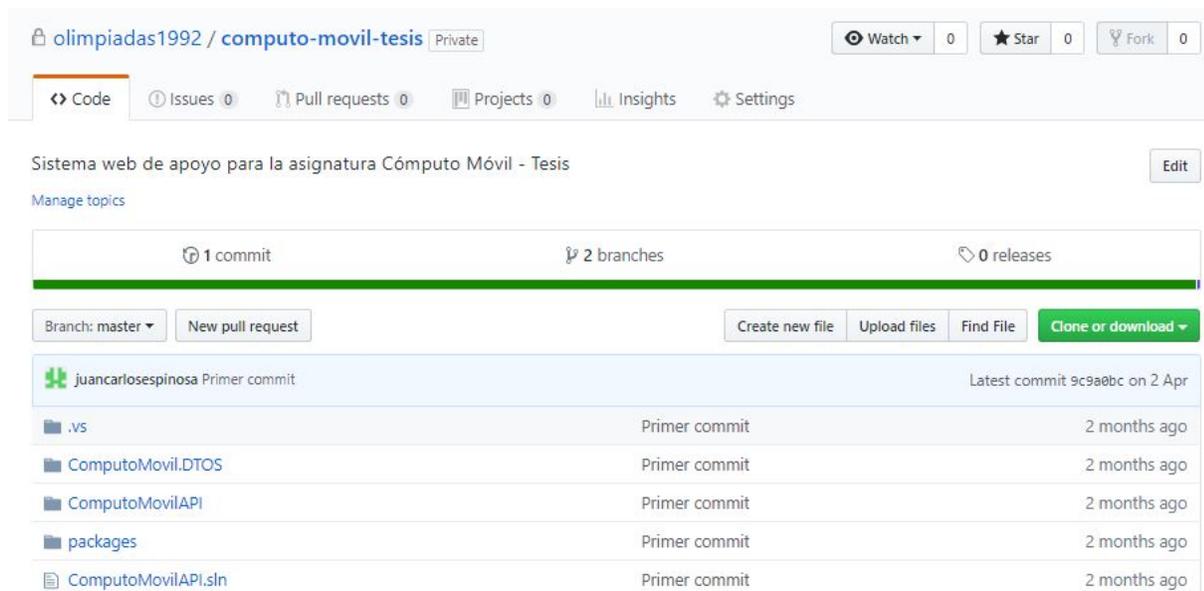


Figura 6.1: Estructura del repositorio de código.



Figura 6.2: Historial de cambios efectuados en el repositorio.

Para tener acceso al código se debe recibir una invitación para colaborar en el proyecto, de esta manera se puede descargar el código a través del *IDE* de *Visual Studio* para comenzar a manipular la solución de Web API.

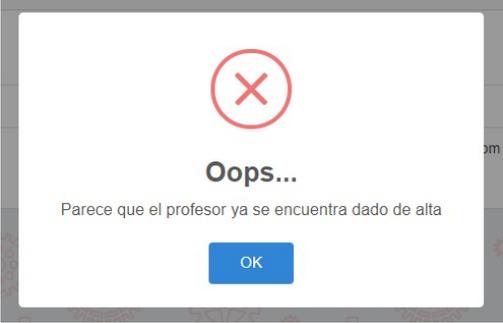
6.2. Casos de Prueba

6.2.1. Casos de prueba Módulo Administrador

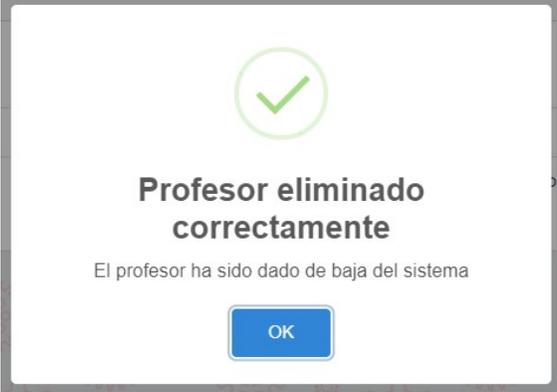
Rol: Administrador	ID: CP-ADMIN01
Objetivo:	Iniciar sesión con rol Administrador
Descripción:	Iniciar sesión con un usuario y contraseña válidos para el rol de Administrador
Resultado esperado:	Al iniciar sesión correctamente el sistema redireccionará al usuario al panel de Administrador
Evidencia:	

Rol: Administrador	ID: CP-ADMIN02
Objetivo:	Iniciar sesión con rol Administrador
Descripción:	Iniciar sesión con un usuario y contraseña no válidos para el rol de Administrador
Resultado esperado:	Al iniciar sesión con datos inválidos para el rol Administrador el sistema mostrará la leyenda “Contraseña incorrecta” y no iniciará sesión.
Evidencia:	

Rol: Administrador	ID: CP-ADMIN03
Objetivo:	Dar de alta a un nuevo Profesor en la plataforma
Descripción:	Dirigirse a la sección “Administrar Profesores” Dar <i>click</i> en el botón “Alta Profesor” Llenar los datos del formulario de alta con información válida. Dar <i>click</i> en el botón “Alta”
Resultado esperado:	Al ingresar correctamente los datos para dar de alta a un nuevo profesor, el sistema generará una contraseña y la enviará vía correo electrónico al profesor para que pueda iniciar sesión en el sistema, posteriormente la sección “Administrar Profesores” se actualizará y mostrará en el listado el profesor que se dio de alta.
Evidencia:	

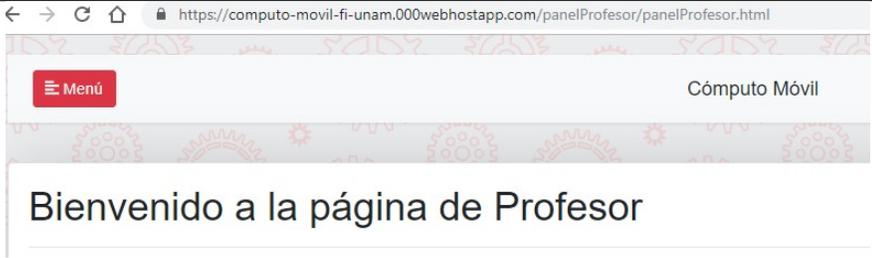
Rol: Administrador	ID: CP-ADMIN04
Objetivo:	Dar de alta a un Profesor que ya se encuentra registrado en la plataforma
Descripción:	Dirigirse a la sección “Administrar Profesores” Dar <i>click</i> en el botón “Alta Profesor” Llenar los datos del formulario de alta con información válida e ingresando el RFC de un profesor existente en el sistema. Dar <i>click</i> en el botón “Alta”
Resultado esperado:	Al ingresar correctamente los datos para dar de alta a un nuevo profesor considerando el RFC de un registro que ya existe en el sistema, el sistema mostrará un mensaje de error indicando que el profesor ya se encuentra dado de alta y no realizará el registro.
Evidencia:	

Rol: Administrador	ID: CP-ADMIN05
Objetivo:	Editar los datos de un profesor
Descripción:	Dirigirse a la sección “Administrar Profesores” Dar <i>click</i> en el botón “Editar” de alguno de los profesores existentes en el sistema Cambiar alguno de los datos del profesor Dar <i>click</i> en el botón “Actualizar”
Resultado esperado:	El sistema realizará los cambios solicitados y actualizará la lista de profesores dados de alta en el sistema.
Evidencia:	

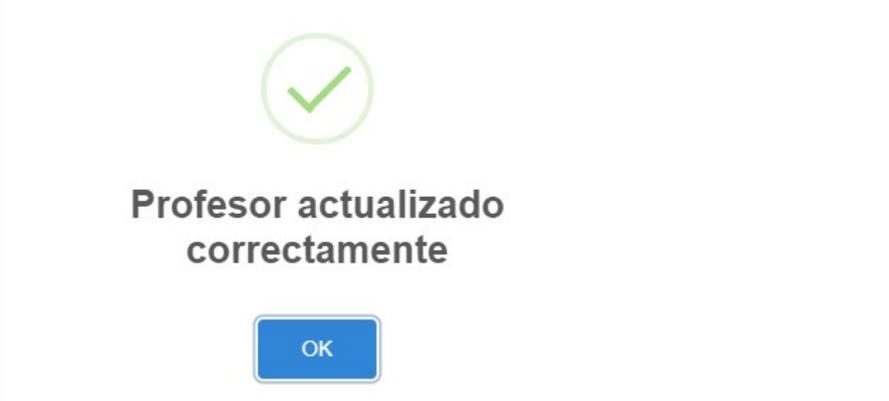
Rol: Administrador	ID: CP-ADMIN06
Objetivo:	Eliminar a profesor que aún no tiene cursos
Descripción:	Dirigirse a la sección “Administrar Profesores” Dar <i>click</i> en el botón “Eliminar” de alguno de los profesores existentes en el sistema que no cuente con ningún curso creado. Confirmar dando <i>click</i> en el botón “Sí, eliminar” del mensaje emergente.
Resultado esperado:	El sistema eliminará al profesor del sistema y actualizará la lista de profesores.
Evidencia:	

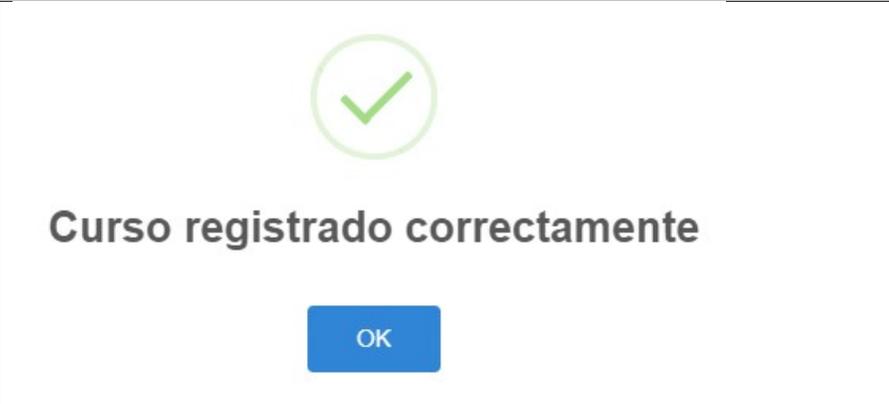
Rol: Administrador	ID: CP-ADMIN07
Objetivo:	Eliminar a profesor tiene cursos
Descripción:	Dirigirse a la sección “Administrar Profesores” Dar <i>click</i> en el botón “Eliminar” de alguno de los profesores existentes en el sistema que cuente con algún curso creado. Confirmar dando <i>click</i> en el botón “Sí, eliminar” del mensaje emergente.
Resultado esperado:	El sistema mostrará un mensaje de error y no realizará la baja del sistema del profesor.
Evidencia:	

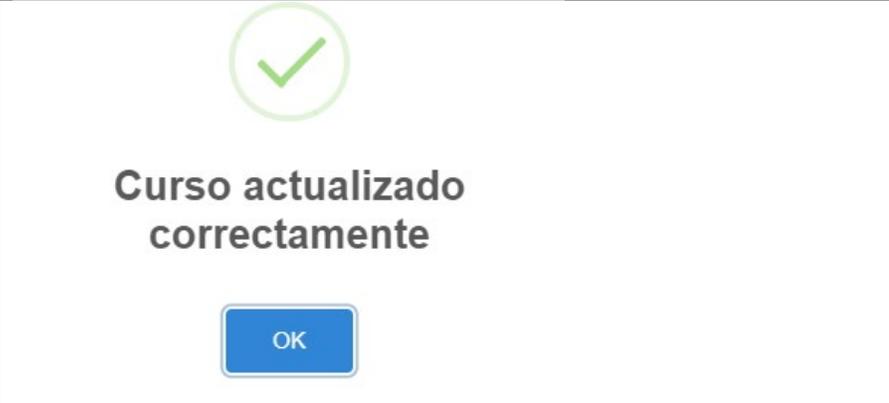
6.2.2. Casos de prueba Módulo Profesor

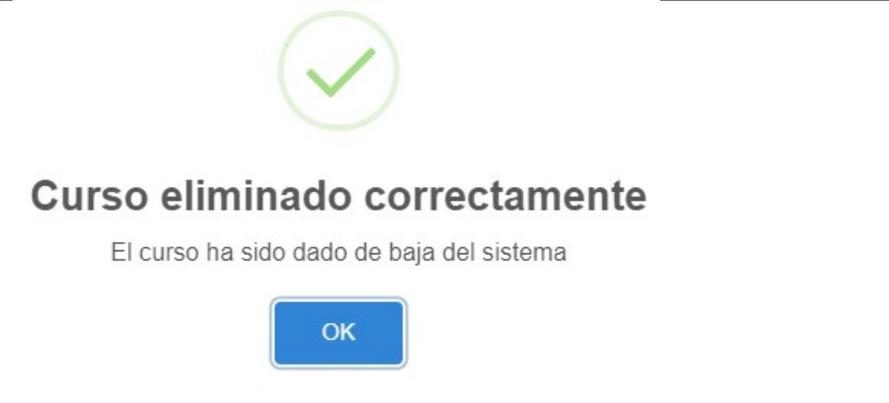
Rol: Profesor	ID: CP-PROF01
Objetivo:	Iniciar sesión con rol Profesor
Descripción:	Iniciar sesión con un usuario y contraseña válidos para el rol de Profesor
Resultado esperado:	Al iniciar sesión correctamente el sistema redireccionará al usuario al panel de Profesor.
Evidencia:	 <p>The screenshot shows a mobile web browser interface. At the top, there is a navigation bar with a red 'Menú' button on the left and 'Cómputo Móvil' on the right. Below the navigation bar, a large heading reads 'Bienvenido a la página de Profesor'. The background features a decorative pattern of gears and sun-like symbols.</p>

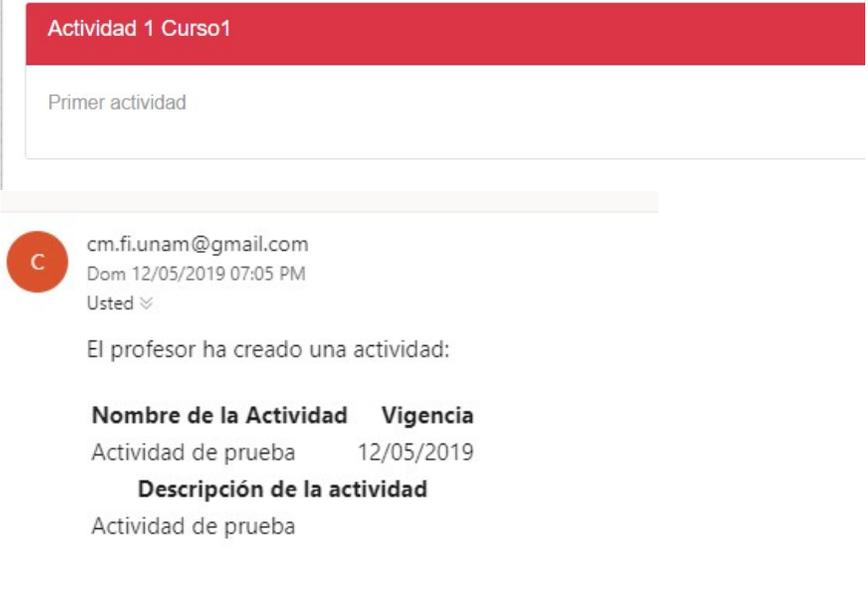
Rol: Profesor	ID: CP-PROF02
Objetivo:	Iniciar sesión con rol Profesor
Descripción:	Iniciar sesión con un usuario y contraseña no válidos para el rol de Profesor
Resultado esperado:	Al iniciar sesión con datos inválidos para el rol Profesor el sistema mostrará la leyenda “Contraseña incorrecta” y no iniciará sesión.
Evidencia:	 <p>The screenshot displays a login form titled 'Inicio de Sesión'. It contains two input fields: the first is for the username, containing 'AAAAAAAAAAAB', and the second is for the password, containing '....'. Below the fields is a red 'Iniciar Sesión' button. At the bottom of the form, the text 'Contraseña incorrecta' is displayed in red, indicating the login attempt failed.</p>

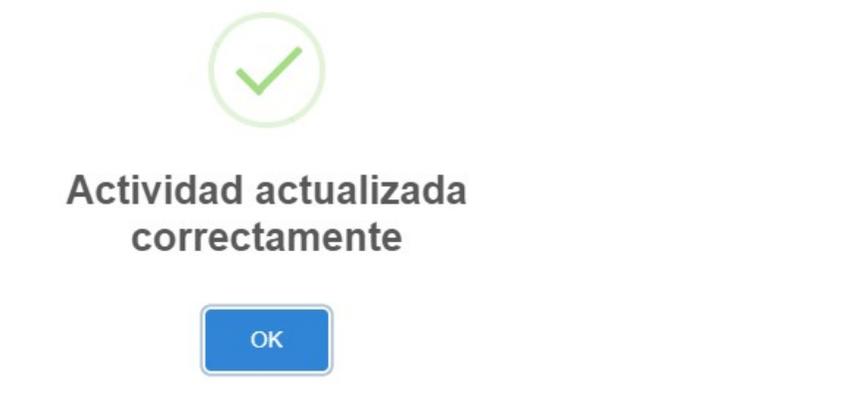
Rol: Profesor	ID: CP-PROF03
Objetivo:	Actualizar perfil Profesor
Descripción:	Dirigirse a la sección “Perfil” Actualizar el campo “Contraseña” Actualizar el campo “Repetir Contraseña” Dar <i>click</i> en “Actualizar”
Resultado esperado:	Al ingresar datos correctos y dar <i>click</i> en “Actualizar” el sistema realizará los cambios en la base de datos y mostrará un mensaje de éxito.
Evidencia:	

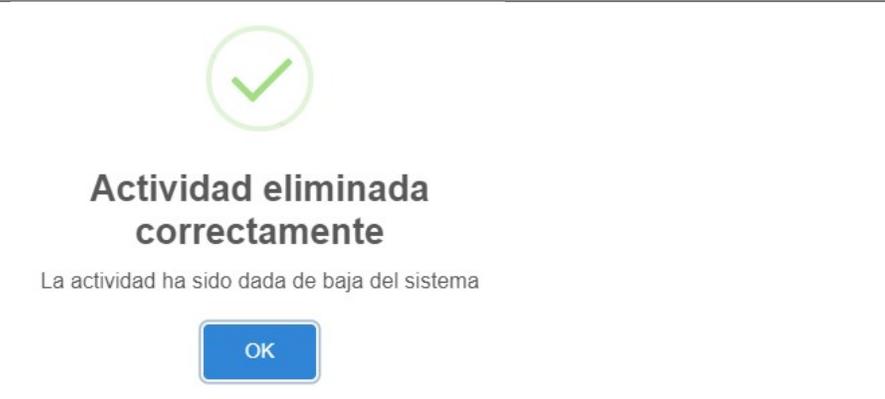
Rol: Profesor	ID: CP-PROF04
Objetivo:	Alta de Curso
Descripción:	Dirigirse a la sección “Cursos” Dar <i>click</i> en “Alta Curso” Llenar los campos correspondientes Dar <i>click</i> en “Alta”
Resultado esperado:	Al ingresar datos correctos y dar <i>click</i> en “Alta” el sistema creará el curso y actualizará la lista de cursos del profesor.
Evidencia:	

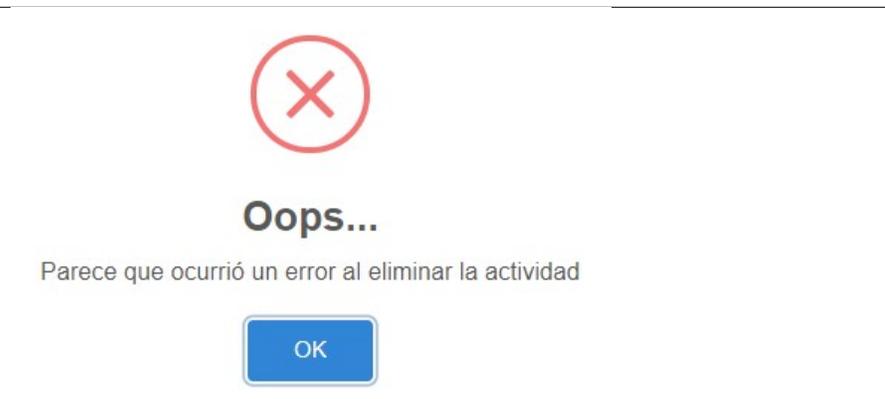
Rol: Profesor	ID: CP-PROF05
Objetivo:	Editar Curso
Descripción:	Dirigirse a la sección “Cursos”. Dar <i>click</i> en el botón “Editar” de alguno de los cursos dados de alta por el profesor. Modificar los campos correspondientes Dar <i>click</i> en “Actualizar Curso”
Resultado esperado:	Al ingresar datos correctos y dar <i>click</i> en “Actualizar Curso” el sistema hará los cambios en el curso y actualizará el listado.
Evidencia:	 <p style="text-align: center;">Curso actualizado correctamente</p> <p style="text-align: center;"><input type="button" value="OK"/></p>

Rol: Profesor	ID: CP-PROF06
Objetivo:	Eliminar Curso
Descripción:	Dirigirse a la sección “Cursos”. Dar <i>click</i> en el botón “Eliminar” de alguno de los cursos dados de alta por el profesor. Confirmar dando <i>click</i> en “Sí, eliminar”
Resultado esperado:	El sistema eliminará el curso seleccionado dando de baja a todos los alumnos inscritos en este y eliminando también actividades, tareas y avisos relacionados.
Evidencia:	 <p style="text-align: center;">Curso eliminado correctamente</p> <p style="text-align: center;">El curso ha sido dado de baja del sistema</p> <p style="text-align: center;"><input type="button" value="OK"/></p>

Rol: Profesor	ID: CP-PROF07				
Objetivo:	Alta de Actividad				
Descripción:	Dirigirse a la sección “Actividades”. Seleccionar algún curso dado de alta por el profesor Dar <i>click</i> en “Alta Actividad” Llenar todos los campos correspondientes Dar <i>click</i> en “Alta”				
Resultado esperado:	El sistema creará la actividad y actualizará la lista de actividades. Así mismo, se notificará vía correo electrónico a los alumnos inscritos en el curso seleccionado.				
Evidencia:	 <p>The screenshot shows a web interface for a course activity. At the top, there is a red header with the text "Actividad 1 Curso1". Below it, the text "Primer actividad" is visible. Below the interface, there is an email notification from "cm.fi.unam@gmail.com" dated "Dom 12/05/2019 07:05 PM". The email content states: "El profesor ha creado una actividad:" followed by a table:</p> <table border="1"> <thead> <tr> <th>Nombre de la Actividad</th> <th>Vigencia</th> </tr> </thead> <tbody> <tr> <td>Actividad de prueba</td> <td>12/05/2019</td> </tr> </tbody> </table> <p>Below the table, the text "Descripción de la actividad" is followed by "Actividad de prueba".</p>	Nombre de la Actividad	Vigencia	Actividad de prueba	12/05/2019
Nombre de la Actividad	Vigencia				
Actividad de prueba	12/05/2019				

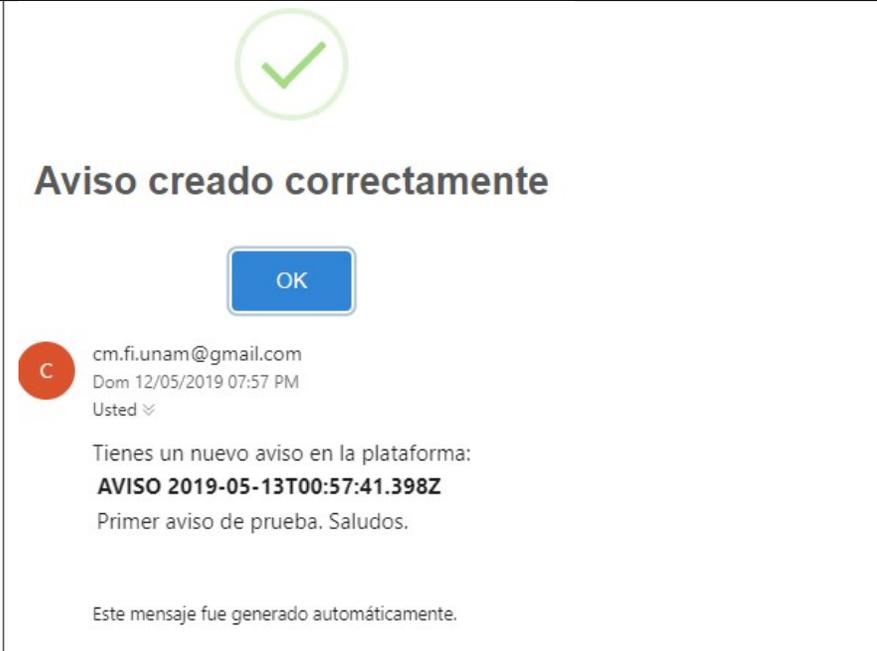
Rol: Profesor	ID: CP-PROF08
Objetivo:	Editar Actividad
Descripción:	Dirigirse a la sección “Actividades”. Seleccionar algún curso dado de alta por el profesor Dar <i>click</i> en “Editar” de alguna actividad dada de alta por el profesor Modificar alguno de los campos disponibles Dar <i>click</i> en “Guardar Actividad”
Resultado esperado:	El sistema actualizará la actividad y recargará la lista de actividades.
Evidencia:	 <p>The screenshot shows a confirmation message with a green checkmark icon in a circle. The text reads: "Actividad actualizada correctamente". Below the text is a blue button with the text "OK".</p>

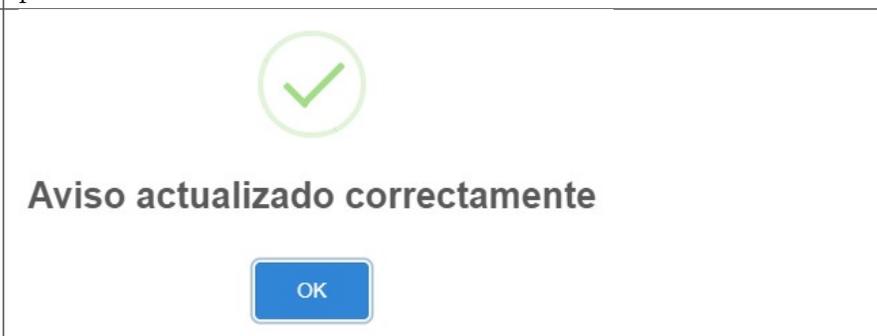
Rol: Profesor	ID: CP-PROF09
Objetivo:	Eliminar Actividad
Descripción:	Dirigirse a la sección “Actividades”. Seleccionar algún curso dado de alta por el profesor Dar <i>click</i> en “Eliminar” de alguna actividad dada de alta por el profesor que no cuente con tareas entregadas. Confirmar dando <i>click</i> en “Sí, eliminar”
Resultado esperado:	El sistema eliminará la actividad y recargará la lista de actividades.
Evidencia:	 <p style="text-align: center;">Actividad eliminada correctamente</p> <p style="text-align: center;">La actividad ha sido dada de baja del sistema</p> <p style="text-align: center;"><input type="button" value="OK"/></p>

Rol: Profesor	ID: CP-PROF10
Objetivo:	Eliminar Actividad
Descripción:	Dirigirse a la sección “Actividades”. Seleccionar algún curso dado de alta por el profesor Dar <i>click</i> en “Eliminar” de alguna actividad dada de alta por el profesor que cuente con tareas entregadas. Confirmar dando <i>click</i> en “Sí, eliminar”
Resultado esperado:	El sistema mostrará un mensaje de error evitando la baja de la actividad.
Evidencia:	 <p style="text-align: center;">Oops...</p> <p style="text-align: center;">Parece que ocurrió un error al eliminar la actividad</p> <p style="text-align: center;"><input type="button" value="OK"/></p>

Rol: Profesor	ID: CP-PROF11								
Objetivo:	Calificar Actividades								
Descripción:	Dirigirse a la sección “Calificaciones” =>“Calificaciones de Actividades”. Seleccionar algún curso dado de alta por el profesor. Seleccionar alguna actividad del curso seleccionado previamente. Dar <i>click</i> en el botón “Ver” de alguna tarea entregada. Asignar una calificación a la tarea seleccionada								
Resultado esperado:	El sistema recargará el listado de tareas entregadas mostrando la calificación puesta por el profesor.								
Evidencia:	<table border="1"> <thead> <tr> <th>Matricula</th> <th>Alumno</th> <th>Calificación</th> <th>Tarea</th> </tr> </thead> <tbody> <tr> <td>309044295</td> <td>Juan Carlos Espinosa Ortiz</td> <td>10</td> <td></td> </tr> </tbody> </table>	Matricula	Alumno	Calificación	Tarea	309044295	Juan Carlos Espinosa Ortiz	10	
Matricula	Alumno	Calificación	Tarea						
309044295	Juan Carlos Espinosa Ortiz	10							

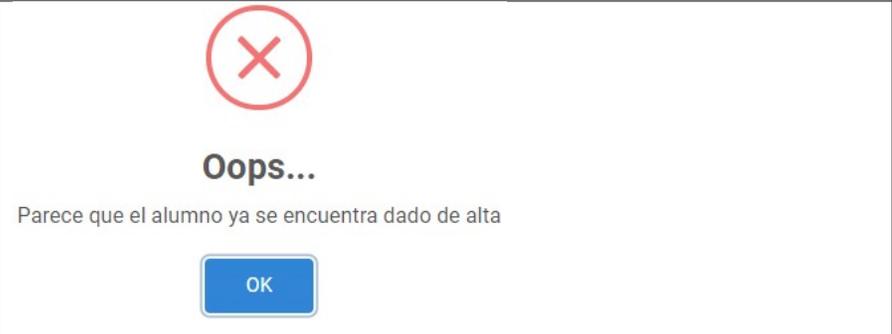
Rol: Profesor	ID: CP-PROF12								
Objetivo:	Calificaciones Finales								
Descripción:	Dirigirse a la sección “Calificaciones” =>“Calificaciones Finales”. Seleccionar algún curso dado de alta por el profesor. Asignar una calificación o dejar la calificación sugerida por el sistema. Dar <i>click</i> en “Calificar”								
Resultado esperado:	El sistema calificará a todos los alumnos del curso y cambiará la etiqueta de la calificación a “calificado”.								
Evidencia:	<table border="1"> <thead> <tr> <th>Matricula</th> <th>Nombre</th> <th>Actividad de prueba con tareas</th> <th>Calificación Final</th> </tr> </thead> <tbody> <tr> <td>309044295</td> <td>Espinosa Ortiz Juan Carlos</td> <td>10</td> <td>10 </td> </tr> </tbody> </table> <p style="text-align: right;"></p>	Matricula	Nombre	Actividad de prueba con tareas	Calificación Final	309044295	Espinosa Ortiz Juan Carlos	10	10 
Matricula	Nombre	Actividad de prueba con tareas	Calificación Final						
309044295	Espinosa Ortiz Juan Carlos	10	10 						

Rol: Profesor	ID: CP-PROF13
Objetivo:	Alta Aviso
Descripción:	Dirigirse a la sección “Avisos”. Seleccionar algún curso dado de alta por el profesor. Dar <i>click</i> en “Crear Aviso”. Llenar la descripción del aviso. Dar <i>click</i> en “Alta”.
Resultado esperado:	El sistema creará el aviso y recargará la lista de avisos creados para el curso seleccionado. Así mismo, se notificará a los alumnos del curso seleccionado vía correo electrónico.
Evidencia:	 <p>The screenshot shows a confirmation email with a green checkmark icon at the top. Below the icon is the text "Aviso creado correctamente". Underneath is a blue "OK" button. The email header includes a red circular profile picture with the letter 'C', the email address "cm.fi.unam@gmail.com", the date and time "Dom 12/05/2019 07:57 PM", and "Usted" with a dropdown arrow. The main body of the email states: "Tienes un nuevo aviso en la plataforma: AVISO 2019-05-13T00:57:41.398Z Primer aviso de prueba. Saludos." At the bottom, it says "Este mensaje fue generado automáticamente."</p>

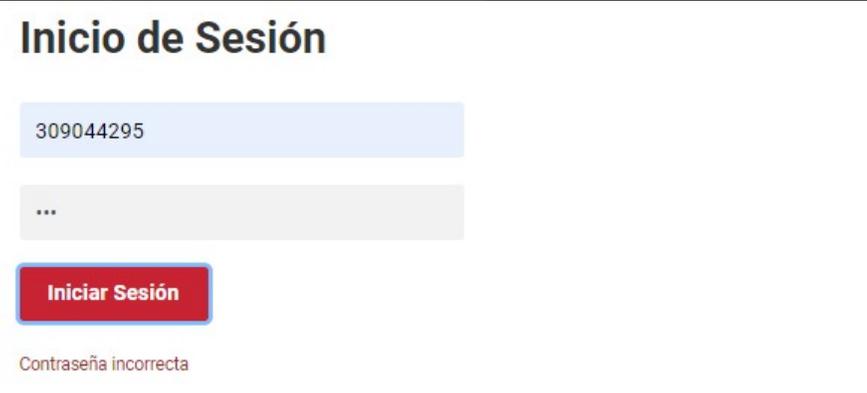
Rol: Profesor	ID: CP-PROF14
Objetivo:	Editar Aviso
Descripción:	Dirigirse a la sección “Avisos”. Seleccionar algún curso dado de alta por el profesor. Dar <i>click</i> en “Editar” de algún aviso creado previamente por el profesor. Modificar la descripción del aviso. Dar <i>click</i> en “Guardar Aviso”.
Resultado esperado:	El sistema actualizará el aviso y recargará la lista de avisos creados para el curso seleccionado.
Evidencia:	 <p>The screenshot shows a confirmation email with a green checkmark icon at the top. Below the icon is the text "Aviso actualizado correctamente". Underneath is a blue "OK" button.</p>

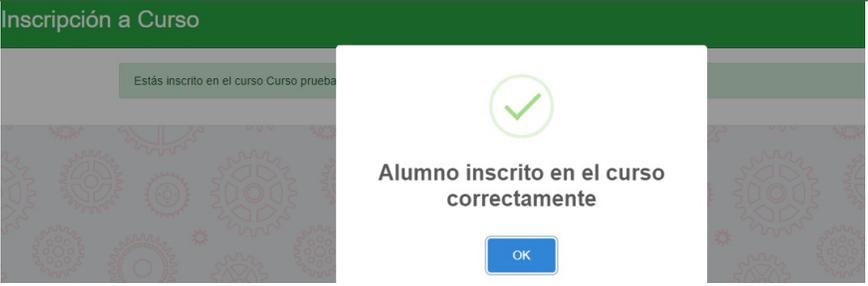
6.2.3. Casos de prueba Módulo Alumno

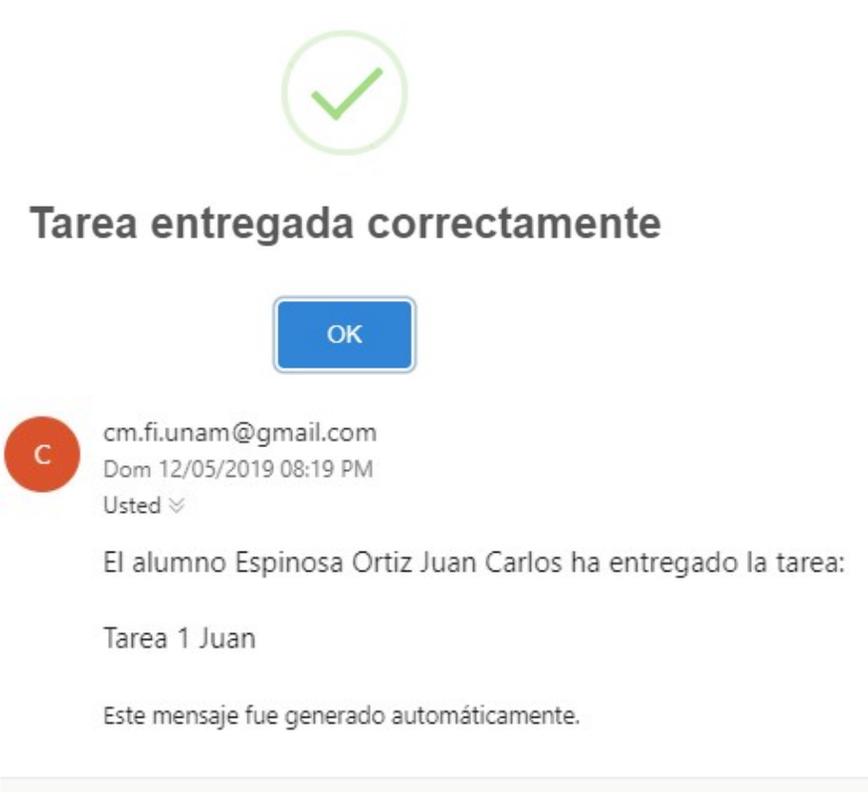
Rol: Alumno	ID: CP-ALUM01
Objetivo:	Registrar nuevo Alumno
Descripción:	Dirigirse a la sección de registro. Llenar cada uno de los campos solicitados. Dar <i>click</i> en “Registrar”
Resultado esperado:	El sistema dará de alta al alumno y se redireccionará a la sección de “Inicio de Sesión”.
Evidencia:	

Rol: Alumno	ID: CP-ALUM02
Objetivo:	Registrar Alumno dado de alta anteriormente
Descripción:	Dirigirse a la sección de registro. Llenar cada uno de los campos solicitados ingresando un número de cuenta de algún alumno que ya haya sido registrado anteriormente. Dar <i>click</i> en “Registrar”
Resultado esperado:	El sistema mostrará un mensaje de error y no realizará el registro del alumno.
Evidencia:	

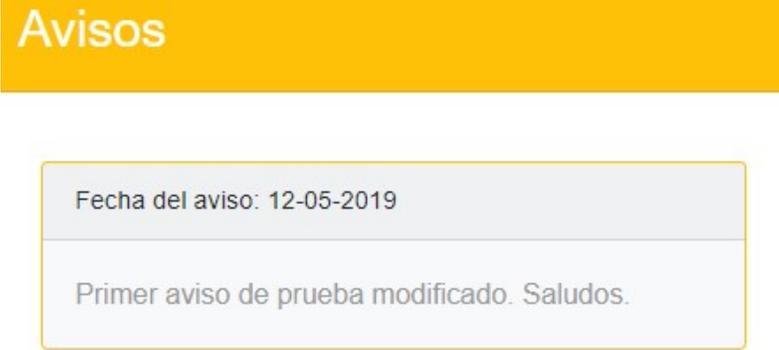
Rol: Alumno	ID: CP-ALUM03
Objetivo:	Iniciar sesión con rol Alumno
Descripción:	Iniciar sesión con un usuario y contraseña válidos para el rol de Alumno.
Resultado esperado:	Al iniciar sesión correctamente el sistema redireccionará al usuario al panel de Alumno.
Evidencia:	

Rol: Alumno	ID: CP-ALUM04
Objetivo:	Iniciar sesión con rol Alumno
Descripción:	Iniciar sesión con un usuario y contraseña no válidos para el rol de Alumno.
Resultado esperado:	Al iniciar sesión con datos inválidos para el rol Alumno el sistema mostrará la leyenda “Contraseña incorrecta” y no iniciará sesión.
Evidencia:	

Rol: Alumno	ID: CP-ALUM05
Objetivo:	Inscripción a Curso
Descripción:	Dirigirse a la sección “Cursos” Ingresar la contraseña de algún curso disponible en el listado. Dar <i>click</i> en el botón “Check”
Resultado esperado:	El sistema mostrará un mensaje confirmando la inscripción al curso y ocultará el listado de cursos del sistema.
Evidencia:	

Rol: Alumno	ID: CP-ALUM06
Objetivo:	Entregar Tarea
Descripción:	Dirigirse a la sección “Actividades” Dar <i>click</i> en “Enviar” de alguna de las actividades creadas en el curso al que está inscrito el alumno. Llenar los campos correspondientes. Dar <i>click</i> en “Alta”.
Resultado esperado:	El sistema mostrará un mensaje confirmando la entrega de la tarea y volverá al listado de actividades. Así mismo se notificará vía correo electrónico al profesor.
Evidencia:	

Rol: Alumno	ID: CP-ALUM07
Objetivo:	Calificaciones de Alumno
Descripción:	Dirigirse a la sección “Calificaciones”
Resultado esperado:	El sistema mostrará la lista de las calificaciones de sus tareas entregadas en el curso, así como la calificación final.
Evidencia:	

Rol: Alumno	ID: CP-ALUM08
Objetivo:	Ver Avisos
Descripción:	Dirigirse a la sección “Avisos”
Resultado esperado:	El sistema mostrará la lista avisos para el curso al que está inscrito el alumno.
Evidencia:	

6.3. Manual Técnico

El Sistema web de apoyo a la asignatura Cómputo Móvil está distribuido en 3 servidores distintos.

- Servidor de Base de Datos.
- Servidor de Web API.

- Servidor Web.

Tanto el Servidor Web API como el Servidor de Base de Datos se encuentran en un mismo grupo de recursos en *Azure*, no obstante, cada uno de los tres servidores presenta sus características y configuraciones particulares por lo que se redactan a continuación de manera separada.

Por razones de seguridad no se exponen en este escrito las credenciales de acceso a cada uno de los recursos que utiliza la plataforma.

6.3.1. Configuración de la Base de Datos

La base de datos es de tipo relacional, está alojada en un servidor *SQL Server* sobre *Microsoft Azure* y cuenta con un tipo de Intercalación *SQL_Latin1_General_CP1_CI_AS*. El tipo de autenticación que se utiliza para conectarse es *SQL Server Authentication*.

El código SQL de sus tablas y la inserción de los datos en los catálogos se muestra a continuación.

Script de creación de tablas

```

/***** Object: Table [dbo].[ACTIVIDADES]
CREATE TABLE [dbo].[ACTIVIDADES](
[IDACTIVIDAD] [int] IDENTITY(1,1) NOT NULL,
[IDCURSO] [int] NOT NULL,
[TITULOACTIVIDAD] [varchar](1024) NOT NULL,
[DESCRIPCIONACTIVIDAD] [varchar](1024) NOT NULL,
[RUTAARCHIVOACTIVIDAD] [varchar](1024) NULL,
[VIGENCIA] [datetime] NULL,
CONSTRAINT [PK_ACTIVIDADES] PRIMARY KEY NONCLUSTERED
(
[IDACTIVIDAD] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON)
ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[ADMINISTRADORES]
CREATE TABLE [dbo].[ADMINISTRADORES](
[IDADMINISTRADOR] [int] NOT NULL,

```

```

[IDTIPOUSUARIO] [int] NULL,
[USERADMINISTRADOR] [varchar](16) NOT NULL,
[PASSADMINISTRADOR] [varchar](16) NOT NULL,
[TOKEN] [varchar](64) NULL,
CONSTRAINT [PK_ADMINISTRADORES] PRIMARY KEY NONCLUSTERED
(
[IDADMINISTRADOR] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON)
ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[ALUMNOS]
CREATE TABLE [dbo].[ALUMNOS](
[MATRICULA] [int] NOT NULL,
[IDCURSO] [int] NULL,
[IDTIPOUSUARIO] [int] NOT NULL,
[NOMBRESALUMNO] [varchar](45) NULL,
[APELLIDOSALUMNO] [varchar](45) NULL,
[EMAILALUMNO] [varchar](45) NOT NULL,
[PASSALUMNO] [varchar](16) NOT NULL,
[CALIFICACIONALUMNO] [float] NULL,
[TOKEN] [varchar](64) NULL,
CONSTRAINT [PK_ALUMNOS] PRIMARY KEY NONCLUSTERED
(
[MATRICULA] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON)
ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[AVISOS]
CREATE TABLE [dbo].[AVISOS](
[IDAVISO] [int] IDENTITY(1,1) NOT NULL,
[IDCURSO] [int] NOT NULL,
[FECHAAVISO] [datetime] NOT NULL,
[DESCRIPCIONAVISO] [varchar](1024) NOT NULL,
CONSTRAINT [PK_AVISOS] PRIMARY KEY NONCLUSTERED
(
[IDAVISO] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON)
ON [PRIMARY]
) ON [PRIMARY]
GO

```

```
/****** Object: Table [dbo].[CURSOS]
CREATE TABLE [dbo].[CURSOS](
[IDCURSO] [int] IDENTITY(1,1) NOT NULL,
[IDPROFESOR] [varchar](13) NOT NULL,
[NOMBRECURSO] [varchar](20) NOT NULL,
[PASSCURSO] [varchar](20) NOT NULL,
CONSTRAINT [PK_CURSOS] PRIMARY KEY NONCLUSTERED
(
[IDCURSO] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON)
ON [PRIMARY]
) ON [PRIMARY]
GO

/****** Object: Table [dbo].[PROFESORES]
CREATE TABLE [dbo].[PROFESORES](
[IDPROFESOR] [varchar](13) NOT NULL,
[IDTIPOUSUARIO] [int] NULL,
[NOMBRESPROFESOR] [varchar](45) NULL,
[APELLIDOSPROFESOR] [varchar](45) NULL,
[EMAILPROFESOR] [varchar](45) NOT NULL,
[PASSPROFESOR] [varchar](16) NOT NULL,
[TOKEN] [varchar](64) NULL,
CONSTRAINT [PK_PROFESORES] PRIMARY KEY NONCLUSTERED
(
[IDPROFESOR] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON)
ON [PRIMARY]
) ON [PRIMARY]
GO

/****** Object: Table [dbo].[TAREAS]
CREATE TABLE [dbo].[TAREAS](
[IDTAREA] [int] IDENTITY(1,1) NOT NULL,
[MATRICULA] [int] NOT NULL,
[IDACTIVIDAD] [int] NULL,
[TITULOTAREA] [varchar](20) NOT NULL,
[DESCRIPCIONTAREA] [varchar](1024) NOT NULL,
[RUTAARCHIVOTAREA] [varchar](1024) NULL,
[CALIFICACIONTAREA] [float] NULL,
[FECHAENTREGA] [datetime] NULL,
CONSTRAINT [PK_TAREAS] PRIMARY KEY NONCLUSTERED
(
[IDTAREA] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
```

```

IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON)
ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[TIPOUSUARIOS]
CREATE TABLE [dbo].[TIPOUSUARIOS](
[IDTIPOUSUARIO] [int] NOT NULL,
[DESCRIPCIONTIPOUSUARIO] [varchar](20) NOT NULL,
CONSTRAINT [PK_TIPOUSUARIOS] PRIMARY KEY NONCLUSTERED
(
[IDTIPOUSUARIO] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON)
ON [PRIMARY]
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[ACTIVIDADES] WITH CHECK ADD CONSTRAINT
[FK_ACTIVIDA_CURSOS_AC_CURSOS] FOREIGN KEY([IDCURSO])
REFERENCES [dbo].[CURSOS] ([IDCURSO])
GO
ALTER TABLE [dbo].[ACTIVIDADES] CHECK CONSTRAINT
[FK_ACTIVIDA_CURSOS_AC_CURSOS]
GO
ALTER TABLE [dbo].[ADMINISTRADORES] WITH CHECK ADD CONSTRAINT
[FK_ADMINIST_ADMINISTR_TIPOUSUA] FOREIGN KEY([IDTIPOUSUARIO])
REFERENCES [dbo].[TIPOUSUARIOS] ([IDTIPOUSUARIO])
GO
ALTER TABLE [dbo].[ADMINISTRADORES] CHECK CONSTRAINT
[FK_ADMINIST_ADMINISTR_TIPOUSUA]
GO
ALTER TABLE [dbo].[ALUMNOS] WITH CHECK ADD CONSTRAINT
[FK_ALUMNOS_ALUMNOS_C_CURSOS] FOREIGN KEY([IDCURSO])
REFERENCES [dbo].[CURSOS] ([IDCURSO])
GO
ALTER TABLE [dbo].[ALUMNOS] CHECK CONSTRAINT [FK_ALUMNOS_ALUMNOS_C_CURSOS]
GO
ALTER TABLE [dbo].[ALUMNOS] WITH CHECK ADD CONSTRAINT
[FK_ALUMNOS_ALUMNOS_T_TIPOUSUA] FOREIGN KEY([IDTIPOUSUARIO])
REFERENCES [dbo].[TIPOUSUARIOS] ([IDTIPOUSUARIO])
GO
ALTER TABLE [dbo].[ALUMNOS] CHECK CONSTRAINT [FK_ALUMNOS_ALUMNOS_T_TIPOUSUA]
GO
ALTER TABLE [dbo].[AVISOS] WITH CHECK ADD CONSTRAINT
[FK_AVISOS_CURSOS_AV_CURSOS] FOREIGN KEY([IDCURSO])
REFERENCES [dbo].[CURSOS] ([IDCURSO])

```

```

GO
ALTER TABLE [dbo].[AVISOS] CHECK CONSTRAINT [FK_AVISOS_CURSOS_AV_CURSOS]
GO
ALTER TABLE [dbo].[CURSOS] WITH CHECK ADD CONSTRAINT
[FK_CURSOS_PROFESORE_PROFESOR] FOREIGN KEY([IDPROFESOR])
REFERENCES [dbo].[PROFESORES] ([IDPROFESOR])
GO
ALTER TABLE [dbo].[CURSOS] CHECK CONSTRAINT [FK_CURSOS_PROFESORE_PROFESOR]
GO
ALTER TABLE [dbo].[PROFESORES] WITH CHECK ADD CONSTRAINT
[FK_PROFESOR_PROFESORE_TIPOUSUA] FOREIGN KEY([IDTIPOUSUARIO])
REFERENCES [dbo].[TIPOUSUARIOS] ([IDTIPOUSUARIO])
GO
ALTER TABLE [dbo].[PROFESORES] CHECK CONSTRAINT
[FK_PROFESOR_PROFESORE_TIPOUSUA]
GO
ALTER TABLE [dbo].[TAREAS] WITH CHECK ADD CONSTRAINT
[FK_TAREAS_ACTIVIDAD_ACTIVIDA] FOREIGN KEY([IDACTIVIDAD])
REFERENCES [dbo].[ACTIVIDADES] ([IDACTIVIDAD])
GO
ALTER TABLE [dbo].[TAREAS] CHECK CONSTRAINT [FK_TAREAS_ACTIVIDAD_ACTIVIDA]
GO
ALTER TABLE [dbo].[TAREAS] WITH CHECK ADD CONSTRAINT
[FK_TAREAS_ALUMNOS_T_ALUMNOS] FOREIGN KEY([MATRICULA])
REFERENCES [dbo].[ALUMNOS] ([MATRICULA])
GO
ALTER TABLE [dbo].[TAREAS] CHECK CONSTRAINT [FK_TAREAS_ALUMNOS_T_ALUMNOS]
GO

```

Configuración de catálogo

La Base de Datos únicamente cuenta con un catálogo llamado “TIPOUSUARIOS” a continuación se muestra la inserción de su contenido, mismo que no puede ser alterado ya que tanto el Servidor de Web API como el Servidor Web utilizan esta información para operar correctamente.

```

INSERT INTO TIPOUSUARIOS VALUES(1, 'Administrador')
INSERT INTO TIPOUSUARIOS VALUES(2, 'Profesor')
INSERT INTO TIPOUSUARIOS VALUES(3, 'Alumno')

```

6.3.2. Configuración del Servidor Web API

El servidor Web API se encuentra alojado en el grupo de recursos creado en *Azure*, es un proyecto de tipo *ASP.NET Framework* y utiliza *Entity Framework* para facilitar la comunicación con el Servidor de Base de Datos.

Dentro del archivo *Web.config* de la solución se incluyen las credenciales de acceso al correo electrónico *cm.fi.unam@gmail.com*, dicho correo es utilizado para realizar el envío de las notificaciones y responder a la solicitud de reposición de contraseña por parte de los alumnos y profesores.

En el mismo archivo se incluye también la cadena de conexión a la base de datos *SQL Server*.

```
<appSettings>
  <add key="AdminEmail" value="cm.fi.unam@gmail.com" />
  <add key="AdminEmailPassword" value="*****" />
  <add key="SMTPName" value="smtp.gmail.com" />
  <add key="SMTPPort" value="587" />
</appSettings>
```

Figura 6.3: Configuración de credenciales de correo electrónico.

```
<connectionStrings>
  <add name="computomovilDBEntities" connectionString="metadata=res://*/Models.computomovilDBEntities.csdl|res://*/Models.computomovilDBEntities.ssdl|res://*/Models.computomovilDBEntities.msl;provider=System.Data.SqlClient;provider connection string="data source=computo-movil-server.database.windows.net;initial catalog=computomovilDB;user id=computomovil;password=*****;MultipleActiveResultSets=True;App=EntityFramework";providerName="System.Data.EntityClient" />
</connectionStrings>
```

Figura 6.4: Configuración de la cadena de conexión de la Base de Datos.

En la misma solución es necesario configurar las rutas de los servicios web y el tipo de respuesta que otorgarán, para este caso, respuestas en formato JSON. Esto es configurable en el archivo *WebApiConfig.cs*.

```
namespace ComputoMovilAPI
{
    public static class WebApiConfig
    {
        public static void Register(HttpConfiguration config)
        {
            // Web API configuration and services
            config.Formatters.XmlFormatter.SupportedMediaTypes.Add(new System.Net.Http.Headers.MediaTypeHeaderValue("application/json"));
            config.EnableCors();

            // Web API routes
            config.MapHttpAttributeRoutes();

            config.Routes.MapHttpRoute(
                name: "DefaultApi",
                routeTemplate: "api/{controller}/{id}/{idCurso}/{token}",
                defaults: new { id = RouteParameter.Optional,
                               idCurso = RouteParameter.Optional,
                               token = RouteParameter.Optional
                            }
            );
            var json = config.Formatters.JsonFormatter;
            json.SerializerSettings.PreserveReferencesHandling = Newtonsoft.Json.PreserveReferencesHandling.Objects;
            config.Formatters.Remove(config.Formatters.XmlFormatter);
        }
    }
}
```

Figura 6.5: Configuración del archivo WebApiConfig.cs.

Finalmente, para que pueda existir un canal de comunicación entre el cliente y el servidor API es importante que todos los servicios permitan el intercambio de recursos de origen cruzado o CORS (*Cross-Origin Resource Sharing*), esto se logra anteponiendo a cada método el encabezado [EnableCors(origins: "*", headers: "*", methods: "*")].

```
// PUT api/alumnos/309044295/token
[EnableCors(origins: "*", headers: "*", methods: "*")]
[Route("api/alumnos/inscribirCurso/{matricula}/{token}")]
public HttpResponseMessage PutInscribirAlumno(int matricula, string token, [FromBody]CursoDTO cursoDTO)
{
    // ...
}
```

Figura 6.6: Configuración CORS en servicios web.

De igual forma, es necesario incluir esta configuración del lado de *Azure* agregando el origen que estará permitido para realizar peticiones al servidor API.



Figura 6.7: Configuración CORS en Azure.

Finalmente, dentro del *App Service* en *Azure* que alberga el Web API se debe configurar la zona horaria que utilizará la aplicación, esto es debido a que algunos *Controllers* obtienen la fecha actual para llevar a cabo funcionalidades determinadas por el negocio, sin embargo como ya se ha mencionado, el Web App está ubicado geográficamente en una posición distinta a la de los usuarios finales, por lo que es necesario configurar la zona horaria del Web App con la zona horaria de la Ciudad de México, lugar de donde comúnmente será consultada la plataforma web.

Esta configuración se agrega en el portal de *Azure* dentro del *Web App* que contiene el Web API, a continuación, se agrega una nueva *application setting* dentro del apartado de configuración con los parámetros mostrados en la figura siguiente.

Add/Edit application setting

Name

WEBSITE_TIME_ZONE

Value

Central Standard Time (Mexico)

Figura 6.8: Configuración de la zona horaria del Web App en Azure.

6.3.3. Configuración del Servidor Web

La aplicación web utiliza tres pequeños archivos PHP: el primero para subir archivos, otro para eliminarlos y uno más para eliminar directorios. Debido a esto es importante que se instale el proyecto de *front-end* en un servidor capaz de ejecutar archivos PHP.

Por otra parte, del lado del código de la plataforma web, es importante importar los siguientes recursos para un correcto funcionamiento.

Bootstrap 4.1.0 Framework para diseño responsive web.

FontAwesome v5.3.1 Fuente de iconos.

SweetAlert2 Librería de alertas personalizadas.

JavaScript con JQuery 3.3.1 y JQuery 1.12.1 Indispensable para las peticiones asíncronas al servidor web API y uso del *DatePicker* personalizado.

CustomScrollBar 3.1.5 Barra scroll personalizada.

Por su parte, el envío de peticiones HTTP a través de *AJAX* deben contener el encabezado *Access-Control-Allow-Origin* para poder comunicarse efectivamente con el servidor web API.

Finalmente, para evitar la navegación entre directorios por medio de la URL, en el archivo del servidor *.htaccess* es necesario agregar la línea: *Options -Indexes*.

6.4. Manuales de Usuario

El sistema web consta de 3 módulos para cada uno de los roles: Módulo de Administrador, Módulo de Profesor y Módulo de Alumno. Cada uno de ellos cuenta con reglas de negocio específicas que describen su funcionamiento.

Antes de ingresar a cualquiera de los 3 módulos es necesario contar con las credenciales necesarias para hacerlo.

Dependiendo de qué tipo de rol se tenga, la obtención de las credenciales se lleva a cabo de una manera u otra. Para administradores, las credenciales se otorgan directamente en la Base de Datos, para profesores se debe solicitar a un administrador que se les dé de

alta en el sistema y para los alumnos, las credenciales se obtienen al realizar el registro a través del sistema web.

Una vez que se tienen las credenciales correspondientes, la navegación dentro de cada uno de los módulos contiene funcionalidades particulares para cada uno.

6.4.1. Módulo Administrador

Este módulo únicamente cuenta con una sección de bienvenida y una más de administración de profesores.

Sección de Bienvenida

En esta sección se detallan las funcionalidades del módulo de Administrador, es decir, se explica cómo operar el apartado de Administrar Profesores.

Sección de Administrar Profesores

En este apartado se muestra un listado de todos los profesores dados de alta en el sistema, de igual forma se muestra un botón para crear a un nuevo profesor y para cada uno de los profesores en el sistema se muestran las opciones de editar su información y eliminarlos de la plataforma.

Alta de profesores en el sistema.

Para dar de alta a un nuevo profesor se debe dar *click* en el botón “Alta Profesor”, dicha acción mostrará una ventana emergente con un formulario de registro, se describen sus campos a continuación.

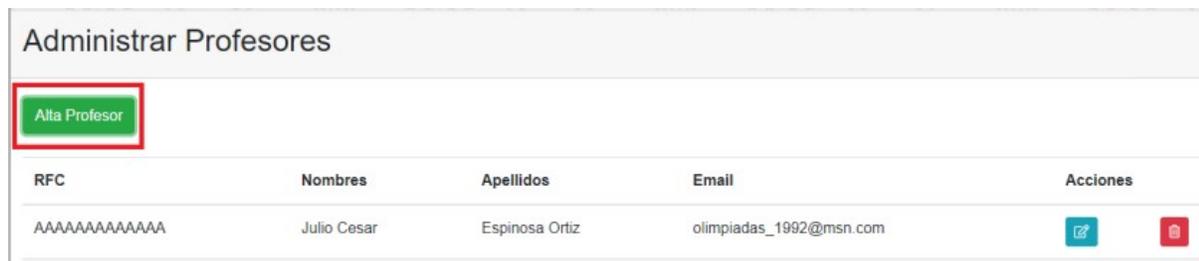


Figura 6.9: Ubicación del botón “Alta Profesor” en el panel de Administrador.

RFC. Es la clave RFC del nuevo profesor, funge como identificador único para los profesores dentro del sistema por lo que no puede repetirse en la plataforma.

Nombres. Nombres del nuevo profesor.

Apellidos. Apellidos del nuevo profesor

Email. Dirección de correo electrónico del profesor. Es importante ingresar correctamente este campo ya que el sistema genera una contraseña y la envía al profesor utilizando la información de este campo para que pueda ingresar a la plataforma.

Una vez completados los campos mencionados, dar *click* en “Alta”, esta acción genera una contraseña aleatoria y la envía al profesor vía correo electrónico.

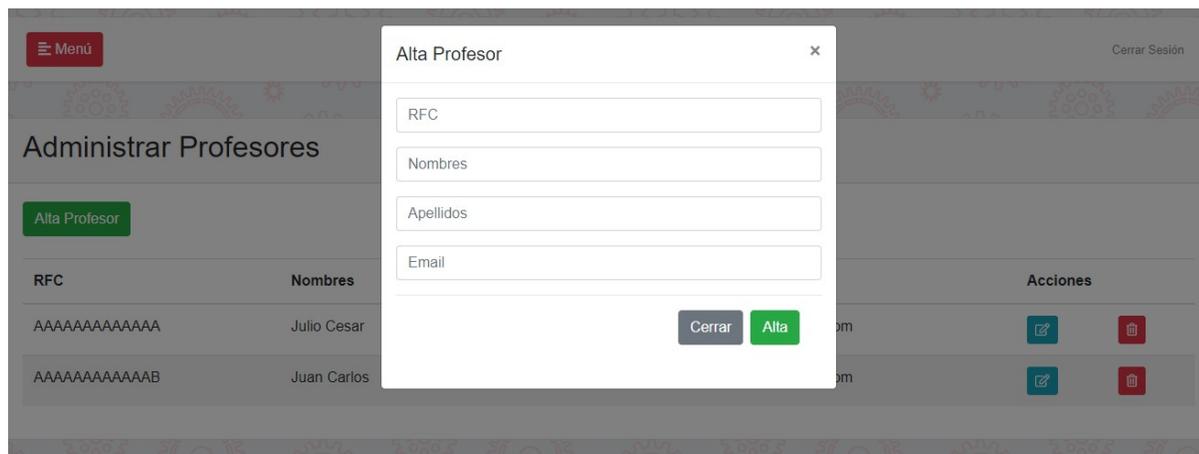
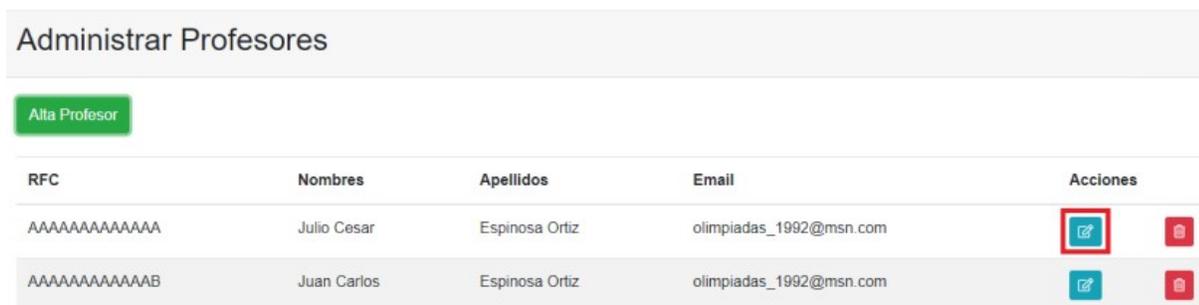


Figura 6.10: Ventana emergente “Alta Profesor” para registrar a un nuevo profesor.

Editar Profesor

Para editar la información de un profesor, es necesario dar *click* en el botón azul con el icono de una hoja y un lápiz.

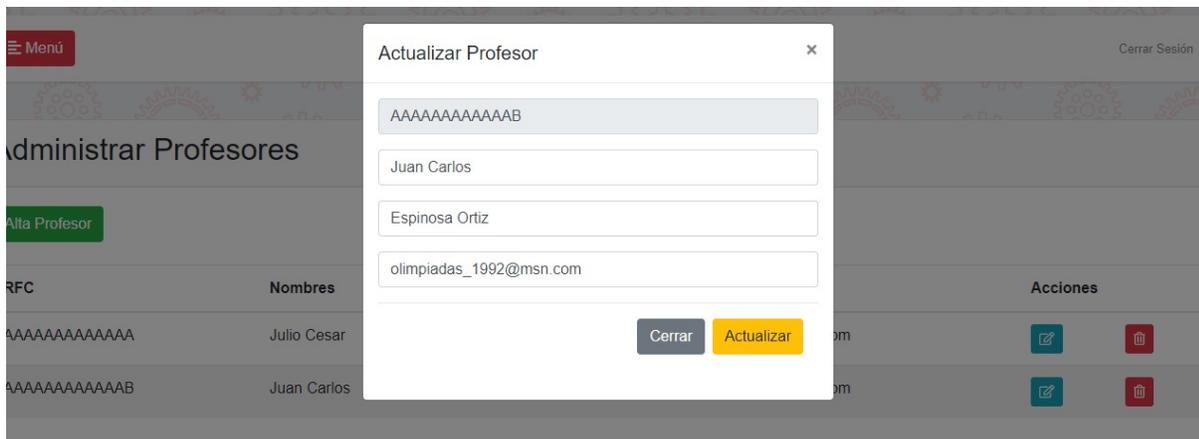


The screenshot shows a web interface titled "Administrar Profesores". At the top left, there is a green button labeled "Alta Profesor". Below it is a table with the following columns: RFC, Nombres, Apellidos, Email, and Acciones. The table contains two rows of data. In the "Acciones" column of the first row, a blue icon of a notepad and pencil is highlighted with a red square.

RFC	Nombres	Apellidos	Email	Acciones
AAAAAAAAAAAAA	Julio Cesar	Espinosa Ortiz	olimpiadas_1992@msn.com	 
AAAAAAAAAAAAAB	Juan Carlos	Espinosa Ortiz	olimpiadas_1992@msn.com	 

Figura 6.11: Ubicación del botón “Editar Profesor”.

Al dar *click* en el botón “Editar” se mostrará una ventana emergente con los mismos campos que la venta “Alta Profesor” pero esta vez se mostraran los datos precargados para que puedan ser modificados (excepto el campo RFC, el cual no es editable).



The screenshot shows a modal window titled "Actualizar Profesor" overlaid on the "Administrar Profesores" interface. The modal contains four input fields with the following values: RFC (AAAAAAAAAAAAAB), Nombres (Juan Carlos), Apellidos (Espinosa Ortiz), and Email (olimpiadas_1992@msn.com). At the bottom of the modal, there are two buttons: "Cerrar" (grey) and "Actualizar" (yellow).

Figura 6.12: Ventana emergente “Actualizar Profesor”.

Eliminar Profesor

Finalmente en la lista de profesores, para cada uno de ellos se muestra un botón con un icono de basurero, al dar *click* en él se mostrará una ventana emergente pidiendo

una confirmación para proceder con el eliminado del profesor. Para poder eliminar a un profesor, es necesario que este no tenga ningún curso activo en el sistema.

Administrar Profesores				
Alta Profesor				
RFC	Nombres	Apellidos	Email	Acciones
AAAAAAAAAAAAA	Julio Cesar	Espinosa Ortiz	olimpiadas_1992@msn.com	 
AAAAAAAAAAAAAB	Juan Carlos	Espinosa Ortiz	olimpiadas_1992@msn.com	 

Figura 6.13: Ubicación del botón “Eliminar Profesor”.

6.4.2. Módulo Profesor

Módulo de Profesor

El módulo de Profesor cuenta con gran parte de las funcionalidades del sistema web, en él se pueden administrar los datos de perfil del profesor, cursos, actividades, calificaciones y avisos, se describe a continuación cada una de las secciones mencionadas.

Sección de Bienvenida

En esta sección se detalla cada una de las funcionalidades del módulo de Profesor, es decir los apartados de Cursos, Actividades, Calificaciones y Avisos.

Sección de Perfil de Profesor

Esta sección permite ver y actualizar los datos del perfil del profesor.

Mi Perfil
RFC AAAAAAAAAAAB

Nombre(s):
Juan Carlos

Apellido(s):
Espinosa Ortiz

Email:
profesor_juan@correo.com

Contraseña:
...

Repetir contraseña:

Actualizar

Figura 6.14: Pantalla de la sección de “Perfil de Profesor”.

Sección de Cursos

En esta sección es posible ver el listado de cursos que tiene actualmente dados de alta en el sistema el profesor, de igual forma es posible crear, editar y eliminar los cursos.

Alta de Cursos

Para dar de alta un nuevo curso en la plataforma web se debe dar *click* en el botón “Alta Curso”, a continuación de mostrará en pantalla una ventana emergente con los siguientes campos:

Nombre del curso. Es el nombre con el que el profesor y los alumnos identificarán el curso en la plataforma.

Contraseña del curso. Con esta clave los alumnos podrán inscribirse al curso.



Figura 6.15: Ubicación del botón “Alta Curso”.

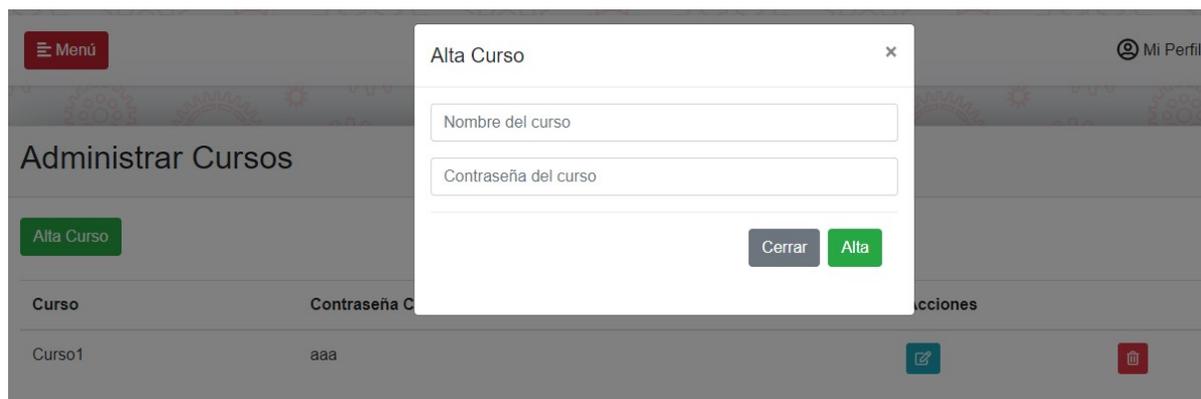


Figura 6.16: Ventana emergente “Alta Curso” para registrar un nuevo curso.

Editar Curso

Para editar un curso es necesario dar *click* en el en el botón azul con el icono de una hoja y un lápiz.



Figura 6.17: Ubicación del botón “Editar Curso”.

Al dar *click* en el botón para editar el curso se desplegará una ventana emergente con

la información precargada para que pueda ser modificada. Para aplicar los cambios se debe dar *click* en el botón “Actualizar” de la pantalla emergente.



Figura 6.18: Ventana emergente “Editar Curso”.

Eliminar Curso

Para eliminar un curso es necesario dar *click* en el icono de basurero del curso que se desea eliminar de la plataforma. Al hacerlo se mostrará un mensaje de confirmación ya que esta acción desinscribe a todos los alumnos del curso, elimina sus tareas, elimina las actividades y los avisos relacionados al curso.



Figura 6.19: Ubicación del botón “Eliminar Curso”.

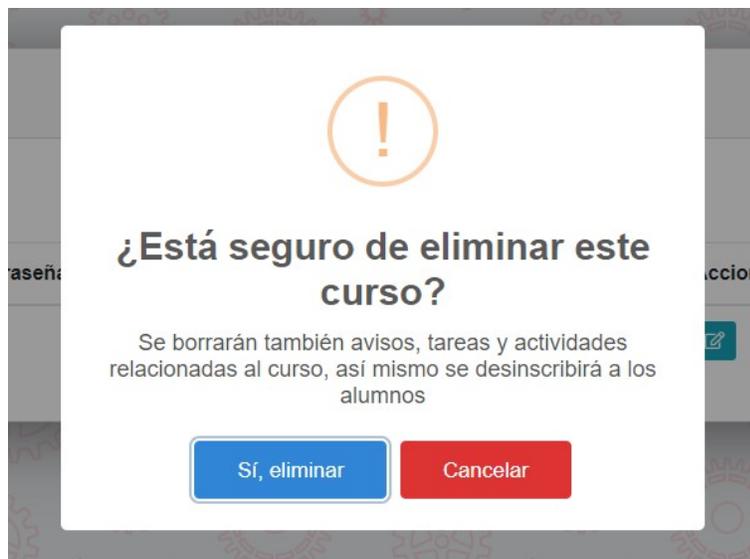


Figura 6.20: Mensaje de confirmación de eliminación de curso.

Sección de Actividades

En esta sección es posible administrar las actividades para cada uno de los cursos que ha dado de alta el profesor. Para poder hacerlo es necesario seleccionar primero el curso del que se desea administrar las actividades.



Figura 6.21: Botón para seleccionar curso en la sección de administrar actividades.

Alta de Actividades

Para dar de alta a una actividad es necesario tener cursos dados de alta en el sistema y haber seleccionado uno dentro de la sección de actividades, cuando se ha realizado lo anterior se mostrará el botón "Crear Actividad" y al dar *click* en él se desplegará una ventana emergente con los siguientes campos.

Título de Actividad. Nombre de la actividad creada por el profesor.

Descripción de Actividad. En este campo se puede incluir una breve descripción de lo que el profesor espera que los alumnos entreguen como tarea.

Seleccionar archivo. Este campo es opcional y sirve para adjuntar algún archivo a la actividad creada.

Vigencia de la actividad. Este campo de tipo fecha sirve para indicar la fecha de vigencia de la actividad, es decir la fecha límite de entrega de tareas relacionada con la actividad en cuestión.

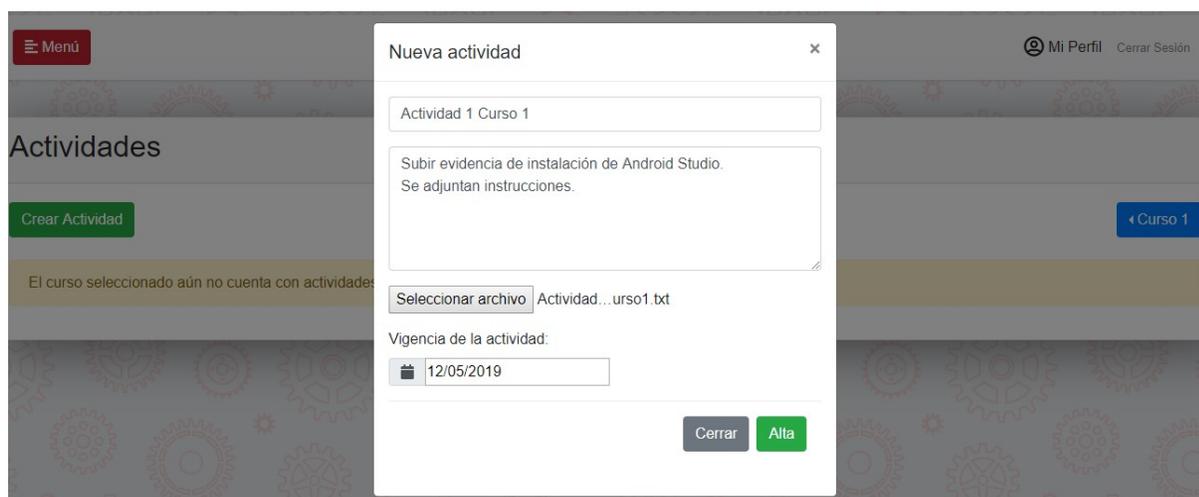


Figura 6.22: Ventana emergente “Nueva Actividad”.

Una vez creada la actividad, el sistema lo notificará a todos los alumnos inscritos en el curso vía correo electrónico.

Editar Actividad

Para editar alguna de las actividades es necesario seleccionar el curso del que se desean administrar las actividades, posteriormente se debe dar *click* en el botón rojo con el icono de una hoja con un lápiz. A continuación, se desplegará una ventana emergente con la información precargada de la actividad para que pueda ser modificada por el profesor. Para guardar los cambios realizados es necesario dar *click* en el botón “Guardar Actividad”.

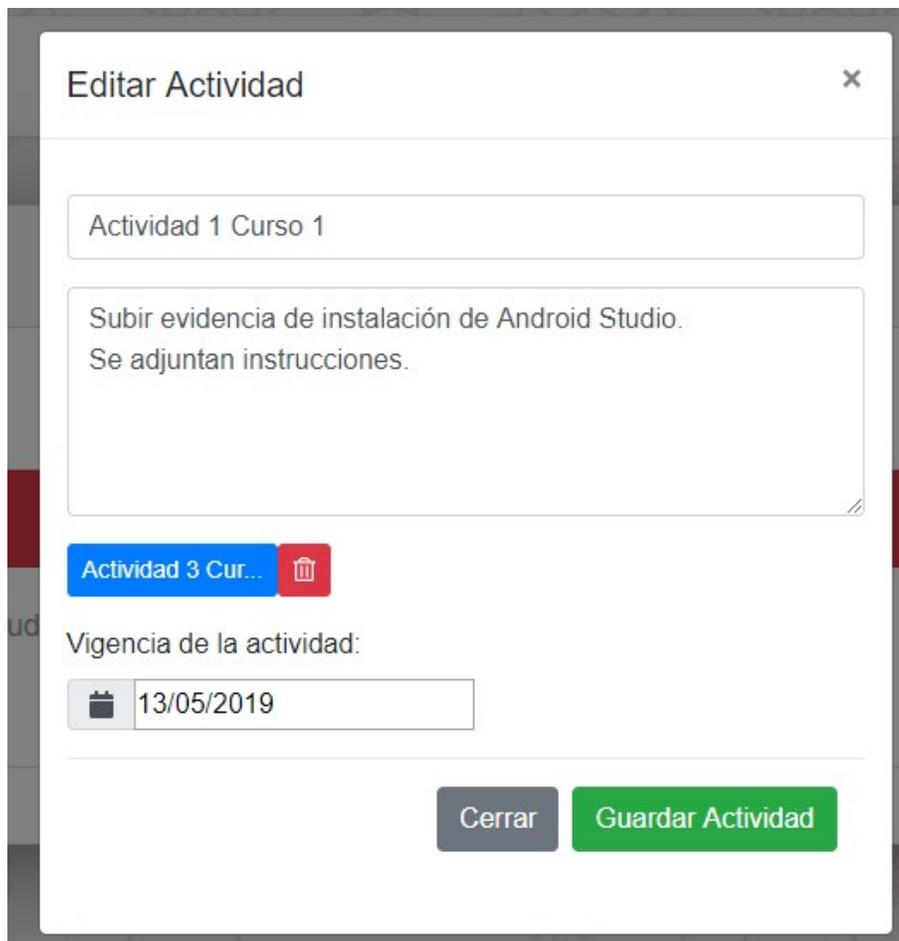


Figura 6.23: Ventana emergente “Editar Actividad”.

Eliminar Actividad

Para eliminar una actividad se debe seleccionar el curso del que se desean administrar las actividades, posteriormente se procede a dar *click* en el botón rojo con el icono de basurero de la actividad que se desea eliminar de la plataforma, a continuación, se mostrará un mensaje para confirmar la acción. Sólo será posible eliminar la actividad si ésta no cuenta con tareas entregadas por parte de los alumnos, si fuera el caso el sistema simplemente mostrará un mensaje de error evitando su eliminación.

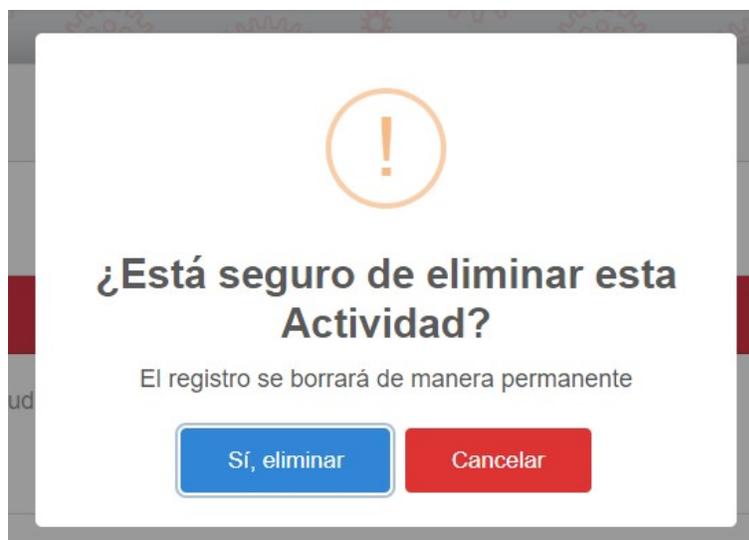


Figura 6.24: Mensaje de confirmación “Eliminar Actividad”.

Sección de Calificaciones

La sección de calificaciones está dividida en dos apartados: Calificaciones de Actividades y Calificaciones Finales. En la primera se pueden evaluar las tareas entregadas por los alumnos por cada una de las actividades dejadas por el profesor, en cambio Calificaciones Finales sirve para asignar una calificación final a cada alumno del curso.

Calificaciones de Actividades

Para calificar las tareas relacionadas a una actividad es necesario seleccionar el curso y la actividad en donde se encuentran las tareas que se desean calificar, una vez realizado esto se desplegará una tarjeta con la información de la actividad a calificar y una lista con las tareas entregadas hasta el momento por los alumnos. Cuando una tarea aún no ha sido calificada, el listado muestra la fila del alumno en color amarillo, una vez evaluada la tarea la fila aparece en color blanco.

Para calificar la tarea se debe dar *click* en el botón verde con el icono de un ojo para ver los detalles de la entrega del alumno a través de una ventana emergente, en dicha ventana se debe asignar la calificación de la tarea y dar *click* en “Calificar”.

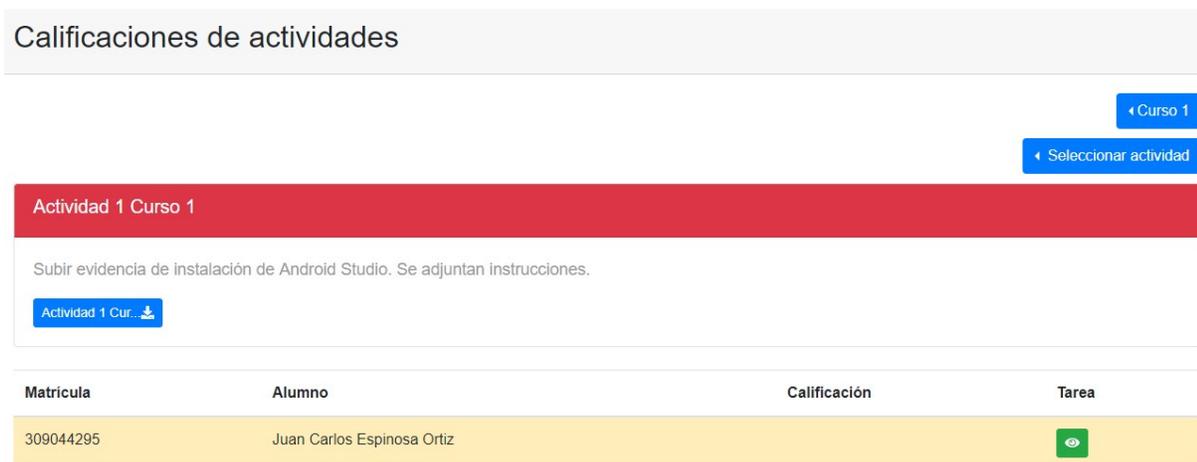


Figura 6.25: Apartado “Calificaciones de Actividades”.



Figura 6.26: Ventana emergente de detalles de Tarea.

Calificaciones Finales

En este apartado se asignan las calificaciones finales del curso, para esto el sistema sugiere una calificación final la cual es el promedio de las tareas entregadas por el alumno, sin embargo, es posible asignar una calificación distinta a la sugerida.

En la pantalla “Calificaciones Finales” se debe seleccionar el curso en el cual se desea asignar calificaciones finales, enseguida se muestra un listado con todos los alumnos ins-

critos en dicho curso con una calificación sugerida. Para calificar basta con asignar una calificación a cada alumno o dejar la calificación sugerida por el sistema y dar *click* en el botón que se encuentra al final del listado de alumnos llamado “Calificar”.

Calificaciones finales

← Curso 1

Matrícula	Nombre	Actividad 1 Curso 1	Calificación Final
309044295	Espinosa Ortiz Juan Carlos	10	<input type="text" value="10.0"/> sugerido

Calificar

Figura 6.27: Pantalla de “Calificaciones Finales” antes de asignar la calificación final.

Una vez realizada la evaluación final, el campo de Calificación Final mostrará la etiqueta “calificado”.

Calificaciones finales

← Curso 1

Matrícula	Nombre	Actividad 1 Curso 1	Calificación Final
309044295	Espinosa Ortiz Juan Carlos	10	<input type="text" value="10"/> calificado

Calificar

Figura 6.28: Pantalla de “Calificaciones Finales” después de asignar la calificación final.

Sección de Avisos

En la sección de Avisos es posible crear, editar y eliminar avisos. Dichos avisos se mostrarán en el panel del alumno y sirven como canal de comunicación unidireccional del profesor hacia sus alumnos.

Crear Aviso

Para crear un aviso es necesario seleccionar el curso al que se desea enviar el aviso, posteriormente se debe dar *click* en el botón “Crear Aviso”, dicha acción desplegará una

ventana emergente con un único campo en el cual se debe ingresar la descripción del aviso. A continuación se debe dar *click* en el botón “Alta” esto creará el aviso y lo notificará a los alumnos inscritos en el curso seleccionado.

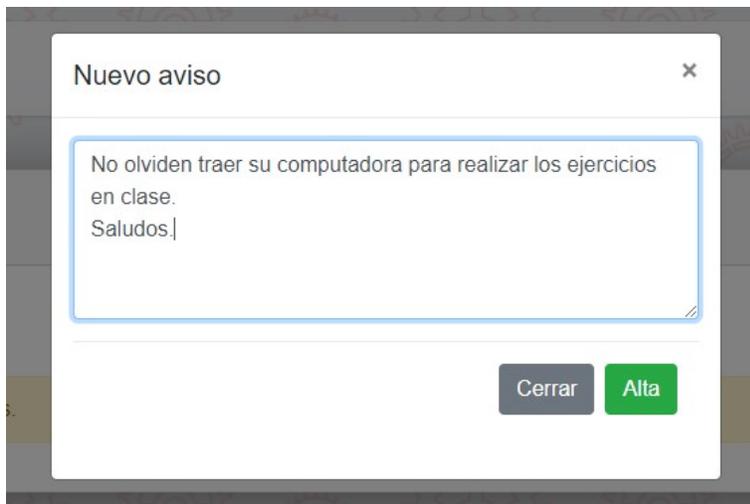


Figura 6.29: Ventana emergente “Nuevo Aviso”.

Editar Aviso

Para editar un aviso basta con dar *click* en el botón azul con el icono de una hoja y un lápiz del aviso que se desea editar, esta acción desplegará una ventana emergente con el aviso precargado y habilitado para ser modificado. Para guardar los cambios se procede a dar *click* en el botón “Guardar Aviso”.

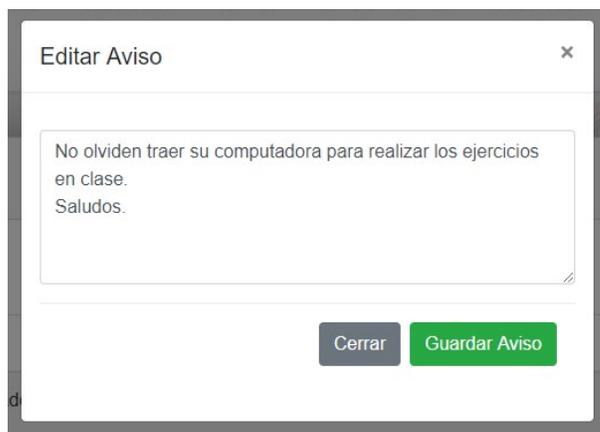


Figura 6.30: Ventana emergente “Editar Aviso”.

Eliminar Aviso

Si se requiere eliminar un aviso de manera permanente, se debe dar *click* en el botón rojo con el icono de un basurero del aviso y confirmar la eliminación en el mensaje de confirmación.

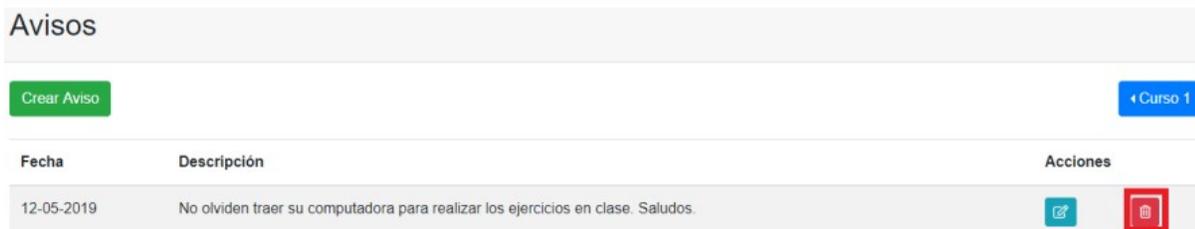


Figura 6.31: Ubicación del botón “Eliminar Aviso”.

6.4.3. Módulo Alumno

A través este módulo, los alumnos se pueden inscribir a un curso y tener acceso al material teórico de la materia, ver los avisos del curso al que están inscritos, así como consultar las actividades y responder a ellas con el envío de una tarea.

Sección de Bienvenida

En esta sección se muestra un tutorial de las funcionalidades del módulo de alumno.

Sección de Perfil de Alumno

En esta sección se cargan los datos personales del alumno para su consulta y actualización.

Mi Perfil
Cta. 309044295

Nombre(s):

Apellido(s):

Email:

Contraseña:

Repetir contraseña:

Figura 6.32: Pantalla de la sección de “Perfil de Alumno”.

Sección de Cursos

En esta sección se muestra un listado de todos los cursos dados de alta en el sistema y con ellos se muestra un campo para ingresar la contraseña del curso que le permitirá al alumno inscribirse en él.

La contraseña del curso debe ser provista por el profesor de manera directa. Para inscribirse al curso basta con ingresar la contraseña del curso y dar *click* en el botón verde con el icono de palomita. Una vez inscrito a un curso no será posible inscribirse a otro por lo que el listado de cursos ya no se mostrará.

Inscripción a Curso		
Curso	Profesor	Inscripción
Curso1	Juan Carlos Espinosa Ortiz	<input type="text"/> <input checked="" type="checkbox"/>

Figura 6.33: Listado de cursos del sistema antes de la inscripción.

Inscripción a Curso

Estás inscrito en el curso Curso1

Figura 6.34: Listado de cursos del sistema después de la inscripción.

Sección de Actividades

En esta sección se muestra una lista de las actividades envidadas por el profesor, aquí es posible enviar una tarea relacionada con dicha actividad. Para responder a la actividad con una tarea es necesario dar *click* en el botón “Enviar” que aparece en cada una de las actividades. Esta acción desplegará una ventana emergente con los siguientes campos.

Título Tarea. Título de la tarea que se desea enviar.

Descripción. Campo para redactar la descripción de la tarea.

Seleccionar archivo. Este campo es opcional y sirve para adjuntar algún archivo a la tarea creada.

Una vez completados los campos se debe dar *click* en el botón “Entregar Tarea”, dicha acción enviará la tarea al profesor y lo notificará vía correo electrónico.



Entregar Tarea

Tarea 1 Juan

Hola profesor, adjunto la evidencia de la instalación de Android Studio.

Saludos.
Juan Espinosa.

Seleccionar archivo Actividad...so 1.png

Cerrar Entregar Tarea

Figura 6.35: Ventana emergente “Entregar Tarea”.

El envío de tareas únicamente está disponible cuando la fecha de entrega de la actividad aún no pasa, de lo contrario el sistema no permitirá entregar la tarea.

Editar Tarea

Para editar una tarea se debe dar *click* en el mismo botón llamado “Enviar” de una tarea que ya se haya entregado anteriormente. Esto desplegará una ventana emergente con la información precargada de la tarea habilitada para ser modificada. Para guardar los cambios es necesario dar *click* en el botón “Actualizar”.

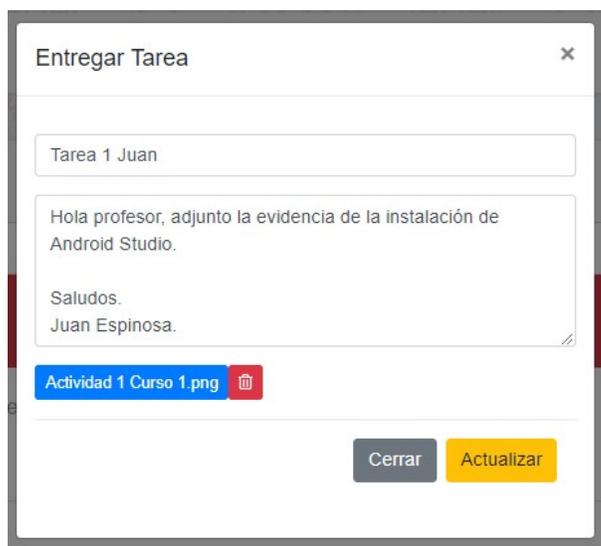


Figura 6.36: Ventana emergente “Entregar Tarea” con datos precargados.

La edición de tareas es posible sólo cuando no ha pasado la fecha de entrega de la actividad. Si la fecha ya pasó únicamente se mostrará la información para su consulta.

Sección de Calificaciones

En esta sección se muestra el estatus y las calificaciones de las actividades, así como la calificación final del curso.

Mis calificaciones	
Actividad	Calificación
Actividad 1 Curso 1	Entregada
Calificación Final	
No calificado	

Figura 6.37: Pantalla de calificaciones antes de recibir evaluaciones.

Mis calificaciones	
Actividad	Calificación
Actividad 1 Curso 1	10 Calificada
Calificación Final	
10	

Figura 6.38: Pantalla de calificaciones después de recibir evaluaciones.

Sección de Avisos

En esta sección se consultan los avisos enviados por el profesor al curso en el cual el alumno está inscrito.

Avisos

Fecha del aviso: 12-05-2019

Se recorre la fecha del examen para el viernes 17 de mayo

Fecha del aviso: 12-05-2019

No olviden traer su computadora para realizar los ejercicios en clase. Saludos

Figura 6.39: Pantalla de la sección de avisos.

Capítulo 7

Referencias

Infosec. Seguridad en Cómputo Móvil. Noviembre 11, 2018, de UNAM Sitio web:

<http://infosec.aragon.unam.mx/tematicas/view/77>

Museo Digital del PC. Historia del Computador Personal. Noviembre 11, 2018, de Museo Digital del PC Sitio web:

http://museopc.ujaen.es/mediawiki/index.php/Historia_del_Computador_Personal

Computación Aplicada al Desarrollo. Historia del Internet. Noviembre 11, 2018, de Computación Aplicada al Desarrollo Sitio web: http://www.cad.com.mx/historia_del_Internet.htm

CASIO. Historia del Negocio de las Calculadoras Electrónicas de CASIO. Noviembre 11, 2018, de CASIO Sitio web: <http://www.casio-intl.com/latin/es/calc/history/>

Tecnología Hecha Palabra. (2006). El Primer Teléfono Celular De La Historia. Noviembre 11, 2018, de Tecnología Hecha Palabra. Sitio web:

<http://www.tecnologiahechapalabra.com/ciencia/miscelanea/articulo.asp?i=443>

García, E. (1998). Windows CE, el sistema operativo "pequeño con grandes" posibilidades. Agosto 25, 2018, de Dealer World Sitio web: <http://www.dealerworld.es/archive/windows-ce-el-sistema-operativo-pequeno-con-grandesposibilidades>

Martí, A. (2016). ¿Qué fue de Palm? Agosto 25, 2018, de Xataka Móvil Sitio web: <https://www.xatakamovil.com/xatakamovil/que-fue-de-las-palm>

Oracle. ¿Qué es J2ME o Java ME? Agosto 25, 2018, de Oracle Sitio web: https://www.java.com/es/download/faq/whatis_j2me.xml

Yuan, M. (2012). Construya mensajería de texto móvil en sus aplicaciones web. Agosto 30, 2018, de IBM Sitio web: <https://www.ibm.com/developerworks/ssa/library/mo-sms-web/index.html>

Microsoft Azure. ¿Qué es middleware? Septiembre 30, 2018, de Microsoft Sitio web: <https://azure.microsoft.com/es-mx/overview/what-is-middleware/>

Barzanallana, R. (2012). Informática. Lenguajes. Octubre 7, 2018, de Universidad de Murcia Sitio web: <https://www.um.es/docencia/barzana/DIVULGACION/INFORMATICA/Que-son-lenguajes-marcado.html>

Exes. XML ¿qué es? Octubre 7, 2018, de Exes Sitio web: <http://www.mundolinux.info/que-es-xml.htm>

Notas del curso Certified Mobile App Professional Testing – Foundation Level (2018)

Notas del curso Cómputo Móvil, profesor Ing. Marco Martínez Quintana (2018)

Glosario

API (Application Programming Interface) conjunto de recursos de software disponibles para ser consumidas por otro software. 17, 35, 38–43, 45, 46, 68, 69, 73–77, 101

ARPANET (Advanced Research Projects Agency Network) red de computadoras precursora de lo que hoy se conoce como Internet. 3, 101

Bluetooth estándar de comunicación para la transmisión de voz y datos entre dispositivos a través de un enlace por radiofrecuencia. 3, 101

bug error de software. 48, 101

FTP (File Transfer Protocol) protocolo de transferencia de archivos entre computadoras conectadas en una red. 3, 47, 101

GPS (Global Positioning System) sistema de posicionamiento global para la ubicación de puntos geográficos por medio de una red satelital. 101

GSM (Groupe Spécial Mobile) estándar de comunicación para la telefonía móvil. 4, 101

hardware componente material que forma parte de la estructura física de una computadora. 2, 5, 6, 8–10, 19, 101

HD (High Definition) sistema de imagen y video de alta calidad. 4, 101

HTTP (Hypertext Transfer Protocol) protocolo de transmisión de información en Internet. 38, 77, 101

IDE (Integrated Development Environment) aplicación utilizada para el desarrollo de software. 46, 54, 101

JSON (JavaScript Object Notation) formato de texto para la representación de información de fácil entendimiento por los humanos. 35, 38, 41, 74, 101

LCD (Liquid Cristal Display) componente electrónico capaz de representar imágenes o formas. 2, 101

microbrowser navegador web optimizado para el uso en pantallas pequeñas, son capaces de enviar y recibir información por redes de telefonía móvil. 12, 23, 101

P2M (peer-to-mail) software que permite compartir archivos a través de cuentas de correo electrónico. 3, 101

P2P (peer-to-peer) red de computadoras en las que no existe un servidor fijo pues cada nodo hace las veces de cliente y servidor según sea requerido. 3, 101

protocolo conjunto de reglas que permiten la comunicación entre sistemas informáticos. 4, 9, 17, 38, 47, 101

relevador dispositivo electromagnético que hace la función de un interruptor activado mediante un circuito eléctrico. 2, 101

script conjunto de instrucciones lógicas usualmente contenidas en un archivo de texto. 19, 101

SMTP (Simple Mail Transfer Protocol) protocolo para el envío de correos electrónicos. 3, 101

software producto informático intangible construido a partir de lógica enfocada a dar solución o soporte a un fin específico. 1–3, 5, 6, 8, 18, 19, 47, 48, 101

SPA (Single-Page Applications) aplicación web que utiliza una única página para mostrar contenido. 44, 101

- SSH** (Secure SHell) protocolo para la comunicación remota con servidores. 3, 101
- stand-by** estado inactivo de un aparato electrónico. 2, 101
- streaming** consumo de contenido online sobre demanda. 5, 101
- Telnet** (Telecommunication Network) protocolo de comunicación para controlar computadoras de manera remota. 3, 101
- token** conjunto de caracteres que sirven para identificar a un individuo dentro de una aplicación informática. 42, 43, 101
- URL** (Uniform Resource Locator) secuencia de caracteres utilizados para localizar un recurso en Internet. 12, 77, 101
- WAP** protocolo basado en los estándares de Internet que permiten a los teléfonos móviles navegar por Internet. 4, 101
- wireless** que hace referencia a un objeto inalámbrico o que no utiliza cables para operar. 9, 101