UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

Plataforma educativa en línea para la asignatura de Criptografía

MATERIAL DIDÁCTICO

Que para obtener el título de

Ingeniero en Computación

P R E S E N T A

Ricardo Sáenz Barragán

ASESOR DE MATERIAL DIDÁCTICO

Ing. Jorge Alberto Solano Gálvez

Ciudad Universitaria, Cd. Mx., 2021

Índice

1. Objetivo de las actividades y/o proyectos.

El presente material de apoyo a la docencia está enfocado en la asignatura de Criptografía, la cual es parte del plan de estudios de la carrera de Ingeniería en computación del plan de estudios 2016 en la Facultad de Ingeniería. Las actividades a realizar son:

- Instalar una plataforma educativa en línea. Crear y agregar contenido de la asignatura.
    - Crear un espacio de apoyo para los docentes y alumnos que cursan la asignatura de Criptografía (alumnos de 9no semestre en el plan de estudios de la carrera de Ingeniería en computación).
    - Crear manuales de instalación y de uso para la plataforma educativa.
- Proporcionar material didáctico para la asignatura.
    - Brindar información de los temas y las herramientas útiles para desarrollar el temario de la asignatura tanto para profesores como para alumnos.
- Realizar actividades y contenido didáctico-pedagógicas para la asignatura.
    - Reforzar el conocimiento adquirido y recalcar el conocimiento adquirido de los usuarios.
- Recabar y facilitar material de consulta de uso público que apoyará el conocimiento de alumnos de la universidad.

1.1. Objetivos

Instalar y configurar una plataforma en línea que sea accesible a profesores que imparten la asignatura de Criptografía y a alumnos que cursan la misma; con la cual será posible evaluar a los alumnos inscritos y gestionar actividades como tareas y prácticas.
Generar material didáctico-pedagógico que facilite la adquisición de conocimiento de la asignatura de Criptografía.
Crear material de consulta de uso público para personas que estén interesadas en temas criptográficos.

1.2. Alcance

El proyecto tiene como alcance instalar y configurar una plataforma educativa en línea y crear parte del material didáctico-pedagógico que contendrá dicha plataforma, sean imágenes, diagramas, animaciones y actividades de las unidades de la asignatura; el material generado podrá ser utilizado por miembros de la asignatura de Criptografía (alumnos, profesores-académicos, etc.) de igual manera el material podrá ser implementado de forma autónoma a la plataforma educativa; también se creará material de consulta de uso público. La plataforma contará con las herramientas para la evaluación y control de calificaciones de los alumnos inscritos en la asignatura de Criptografía.

Este es un proyecto en conjunto con otro tesista, en el cual desarrollara los temas a profundidad de la asignatura de Criptografía. El enfoque del proyecto actual es la plataforma que se usará para almacenar la información y los materiales de apoyo, sean ilustraciones y actividades. De igual

manera el material del actual proyecto se puede usar de forma autónoma para explicar temas de la asignatura de Criptografía.

Orientación del proyecto: Las herramientas brindadas por la plataforma permitirán el acceso a la información de la asignatura, sirviendo de apoyo para los alumnos. De está manera el material estará al alcance de los alumnos inscritos apoyando al estudio de la asignatura ante cualquier situación adversa que se les presente. Por ejemplo, facilitará la preparación para un examen y evitará el retraso académico. Dicha plataforma, además de ser benéfica en la actual pandemia posteriormente podrá servir para el aprendizaje autodidacta, ya sea como material de apoyo en clases presenciales o para alumnos sin derecho a reinscripción (ASDRI) que deseen prepararse para el examen extraordinario.

1.2.1. Temas a abordar

En el plan de estudios 2016 de la licenciatura en Ingeniería en Computación se puede ver en 9° y 10° semestre la sección correspondiente a "optativas de campo de profundización" (Fig. 1) en donde se encuentra la asignatura de Criptografía, correspondiente al campo de profundización de "Ingeniería de Software".

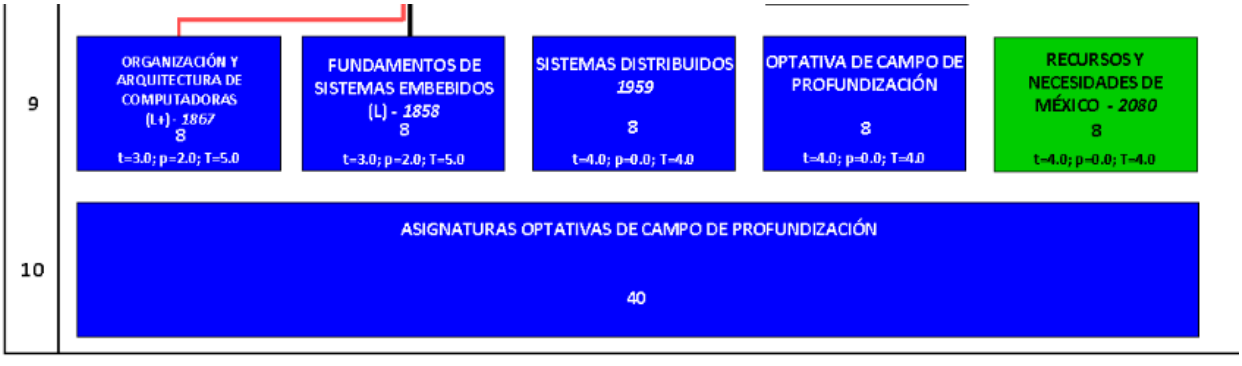


Fig. 1 Sección final de plan de estudios de la licenciatura en Ingeniería en Computación.

En el programa de estudios de la asignatura de Criptografía, dentro del plan de estudios de la licenciatura, se seleccionaron temas específicos que fueran capaces de ser abstraídos a formas gráficas, que tuvieran la capacidad de ludificación o fueran de interés para un público general ajeno a la facultad. De estos temas se decidió desarrollar material didáctico que fuera útil para explicarlos. De igual manera, en conjunto con la Dra. Rocío Alejandra Aldeco Pérez, docente que imparte la asignatura, se agregaron temas de utilidad, como es el caso de Blockchain y sus derivados.

A continuación se presentan el contenido seleccionado del temario[1], más los temas que fueron añadidos, estos ordenados por unidades:

[1] S.A (2015) PROYECTO DE MODIFICACIÓN DEL PLAN DE ESTUDIOS DE LA LICENCIATURA DE INGENIERÍA EN COMPUTACIÓN. Facultad de Ingeniería UNAM, CD.MX., México, Recuperado de https://www.ingenieria.unam.mx/programas_academicos/licenciatura/Computacion/2016/asignaturas_computacion_2016.pdf

Unidad 1: Panorama general

- Conceptos básicos de criptografía
- Servicios y mecanismos de seguridad.

Unidad 2: Técnicas clásicas de cifrado

- Introducción y clasificación de los sistemas de cifrado
    - Número de claves: algoritmos simétricos y asimétricos
    - Formas de procesar datos: algoritmos en flujo y en bloque.
    - Operaciones utilizadas: sustitución y transposición
- Algoritmos de sustitución.
    - Monoalfabética: Polybios, César, Afín y Playfair
    - Polilfabética: Vigenéré.
- Algoritmos de transposición.

Unidad 3: Gestión de claves

- Generadores y distribución de claves
    - Generadores pseudoaleatorios
- Protocolos de Distribución de Llaves
    - Diffie-Hellman
    - TLS / SSL

Unidad 4: Criptografía simétrica o de clave secreta

- Introducción a la criptografía simétrica
    - Características de los algoritmos simétricos
- DES y 3DES (Data Encryption Standard)
- AES (Advanced Encryption Standard)
    - Orígenes
    - Proceso de cifrado y descifrado (bloques de 128, 192 y 256 bits)
    - Modos de funcionamiento
    - Aplicaciones
    - Análisis de seguridad

Unidad 5: Criptografía asimétrica o de clave pública

- Introducción a la criptografía asimétrica
- Funciones Hash
    - Orígenes
    - Funciones sólo de ida y sus propiedades
    - Funcionamiento de MD5 y sus ataques
    - Funcionamiento SHA1 y sus ataques
    - Funcionamiento SHA2 y SHA3
    - Aplicaciones de los algoritmos
- Introducción a Criptografía Cuántica
    - Introducción y entrelazamiento cuántico
    - Propiedades y protocolos

Unidad 6: Aplicaciones criptográficas

- Firmas Digitales
- Certificados Digitales
- Aplicaciones en Redes
- Aplicaciones Descentralizadas
    - Hash chains
    - Blockchain

2. Definición del problema.

2.1. Antecedentes

A partir de la situación de pandemia mundial que vivimos actualmente, debido al COVID-19, los procesos de enseñanza-aprendizaje han tenido que ser modificados para satisfacer las necesidades estudiantiles de educación a distancia y así poder continuar con el desarrollo de las actividades académicas y la adquisición del conocimiento de forma eficiente.

Ante ésta problemática, la carrera de Ingeniería en Computación no ha sido la excepción viéndose afectada por dicho contexto. Dentro del plan de estudios de esta carrera se decidió tomar a la asignatura de **Criptografía** para desarrollar el proyecto de apoyo a la docencia, gracias a la posibilidad que tiene de desenvolverse a través de actividades didácticas y al potencial de abstracción que existe en el contenido de los temas para ser representados en un formato gráfico. Debido a que la asignatura contiende temas complejos y muchos conceptos que pueden resultar difíciles de entender, la implementación y uso de diversas técnicas, así como de recursos didácticos servirán de apoyo a diferentes tipos de aprendizaje, como:

- Asociativo: en el que el sujeto asocia un concepto con apoyo visual.
- Significativo: en el cual el sujeto recolecta información y la relaciona con conocimientos previos.
- Por descubrimiento: aprendizaje activo que se sostiene con las actividades a través de las cuales el sujeto aprende de forma directa.
- Receptivo: en el cual el sujeto recibe conocimientos y los ejecuta sin necesidad de descubrirlos por su cuenta.

Estos tipos de aprendizaje no siempre son atacados al momento de tomar la asignatura de forma presencial.

Con este proyecto es posible brindar material didáctico que de otra forma no es sugerido dentro del plan de estudios, esto siendo; el uso de plataformas educativas, búsquedas especializadas en internet y redes sociales con fines académicos.

Por lo que se plantea la creación de una plataforma, que apoye el desarrollo de la materia; Criptografía es una asignatura optativa que pertenece al campo de profundización de Ingeniería de Software, dicha clase consta de 64 horas teóricas ofertadas a estudiantes del 9no semestre en adelante. La plataforma contendrá material didáctico y visual, a través del cual se explicarán los conceptos y actividades preestablecidas en el temario, con el afán de llevar de manera remota la asignatura. Si es posible en un futuro el aprendizaje de la misma podrá ser autodidacta. Además de esto, el docente podrá valerse del recurso mencionado en clases presenciales y en línea, siendo un recurso extra.

En la creación de este material se utilizaron métodos teórico-prácticos, con el objetivo de transmitir y evaluar conocimientos. Esto ayudará a la integración de esto en el ámbito educativo.

Como resultado final se planea que el uso de esta plataforma apoye a la optimización del uso de espacio en el tiempo de materias presenciales. Además de que cuente con material adicional, de dominio público, que la convierta en un apoyo para otras asignaturas dónde la criptografía pueda incluirse como un apoyo transversal.

El eficiente desarrollo del sistema en línea en materias como Criptografía, Seguridad Informática y asignaturas afines del área de las Tecnologías de la Información se centra en el apoyo a profesores y alumnos, además de que facilita el acercamiento de dichos temas a personas interesadas en el área.

Para sustentar el éxito esperado en el funcionamiento de la propuesta, de puede analizar el creciente uso que las tecnologías de la información y la comunicación (TIC) han tenido dentro del ámbito educativo en los últimos años. Dichas tecnologías han ayudado al conocimiento de forma complementaria, facilitando el acceso a la información a estudiantes de todos los niveles educativos, así como permitido a los docentes ampliar el panorama con el que contaban, cuestión que facilita la innovación y mejora las estrategias para el proceso de enseñanza–aprendizaje.

Esto se describe más a detalle en el término TAC, que se refiere a las Tecnologías del Aprendizaje y el Conocimiento. El principal objetivo de estas se basa en orientar a las TICs en los diferentes procesos de enseñanza-aprendizaje, creando metodologías centradas en dichos materiales y poniéndolas al alcance de los participantes en la tarea educativa. Así surgen nuevas modalidades, en las que sus principales recursos son; el uso de internet, sistemas de gestión de aprendizaje y diversos recursos digitales.

Algunas de las modalidades que surgen de estos recursos son la semi-presencial (b- learning) y la modalidad a distancia (e-learning). Estos dentro de sistemas cuentan con varias ventajas, como la adaptación de los procesos educativos a cada sujeto, es decir; el lugar y tiempo de la tarea educativa se adaptará a las actividades de los estudiantes sin importar cuestiones externas. Además, la responsabilidad generada en el estudiante suele ser mayor, volviendo al profesor un medio que facilite el aprendizaje, sin olvidar que en el proceso se potencia el uso de herramientas virtuales impulsando las habilidades que puede desarrollar.

2.2. Ventajas y desventajas del uso de Moodle

Hoy en día existen varias herramientas digitales para el aprendizaje, mejor conocidas como *Sistemas de Gestión de Aprendizaje* (SGA; en inglés, *learning management system* o LMS). Pero una que ha resaltado sobre las demás es Moodle. Al ser una plataforma gratis con una gran facilidad de acceso, comunidad activa, gran capacidad de ser modificada en diseño y funcionalidad a las necesidades de cada usuario tanto para pequeñas y grandes empresas, ha destacado para la educación a distancia.

Moodle permite adaptarse a diferentes métodos de enseñanza, ya sea totalmente a distancia o combinado con clases presenciales. El almacenamiento de la plataforma es en la nube, por lo tanto los usuarios pueden acceder al material en cualquier momento mientras tengan un dispositivo y una conexión a internet.

La plataforma proporciona un espacio de colaboración para profesores y estudiantes. Las características clave incluyen creación masiva de contenido con respaldo, administración de archivos integrada y soporte multilingüe. También proporciona varias opciones para monitorear el progreso de los estudiantes y la creación segura de cuentas de usuarios con autenticación.

Un SGA proporciona la facilidad de integrarse con una amplia variedad de software, como herramientas de comunicación, colaboración, gestión de documentos y otras aplicaciones de productividad. Es compatible con cualquier dispositivo con un navegador web y cuenta con una aplicación nativa para dispositivos Android. Debido a que es de código abierto es posible usar el sistema de forma gratuita, de igual manera la organización Moodle pone a disposición paquetes de paga si no tienes los recursos técnicos, estos pueden llegar a costar a partir de $80 dólares por año.

Dentro de las principales ventajas de Moodle es la creación masiva de contenido con respaldo, la administración de archivos incorporada y el soporte multilingüe.

La interfaz de la plataforma es intuitiva y fácil de aprender usando herramientas de arrastrar y soltar. Cuenta con un tutorial inicial para las acciones básicas de cada usuario y es posible crear tutoriales específicos siendo el administrador de la plataforma.

Moodle siempre se está actualizando para mejorar las experiencias del usuario, protección contra errores y hackeos. Su enorme comunidad y cantidad de usuarios comparten constantemente consejos y las mejores practicas de uso. A pesar de que Moodle se mantiene actualizado constantemente, es necesario que el administrador actualice de forma manual y constante la plataforma instalada al igual que las dependencias que usa Moodle, por ejemplo PHP y MySQL.

Más allá de personalizar el diseño, Moodle se puede configurar para admitir el aprendizaje totalmente a distancia o combinado con clases presenciales. Eso significa que los procesos y los roles de los usuarios se pueden personalizar para adaptarse a cualquier metodología.

De las mayores desventajas de Moodle es el requisito técnico que se necesita para instalar y mantener la plataforma; si el uso es mediante los paquetes que ofrece la organización o un tercero, donde uno no se encargan de administrar el lado técnico de Moodle puede ser de muy fácil su uso, pero de no ser así es necesario tener un conocimiento bastante avanzado para la instalación, igual que para el mantenimiento y actualización de la plataforma, lo cual puede causar que muchos interesados en la SGA sean ahuyentados de usarla.

3. Metodología.

Para lograr el objetivo del presente trabajo se implemento una plataforma que apoye el desarrollo de la materia para la asignatura de criptografía. A través de material didáctico y visual se explicarán los conceptos y actividades con base en el secciones del temario, para llevar de manera remota la asignatura y si es posible que en un futuro el aprendizaje de la misma sea autodidacta. Además de esto, el docente podrá valerse del recurso mencionado en clases presenciales y en clases en línea. A continuación, se describen los puntos a cubrir para lograr el objetivo planteado.

- *Instalación de la plataforma*

  Realizar un análisis con las ventajas y desventajas de la plataforma Moodle. Describir el proceso mediante el cual se obtiene la herramienta, así como los complementos necesarios para el funcionamiento e implementación de estándares de seguridad para hacer una plataforma práctica y segura.

  - *Instalación*

    Se muestra los pasos a seguir para una correcta instalación de la plataforma, además de las medidas de seguridad necesarias para la óptima operación de la misma.

  - *Configuración*

    Se adaptó la plataforma a las necesidades particulares que presente el curso, además de incluir la instalación de complementos y diseño específico; por ejemplo, el uso de colores y logotipos de la Facultad de Ingeniería.

  - *Manual de administrador*

    La creación un instructivo de apoyo que facilite el mantenimiento, seguridad y administración (creación de usuarios, modificación y gestión de contenido, modificación de diseño, etc.) de la plataforma.

  - *Manual de usuario*

    La creación un instructivo de apoyo que facilite a docentes la visualización de contenido, calificación e inscripción de alumnos, etc.; a alumnos el uso del contenido dentro de la plataforma, entrega de documentos, etc.

- *Generación de contenido*

  Se realizó una investigación y documentación de temas específicos de la asignatura de criptografía para el desarrollo de diagramas y animaciones afines de las unidades, al igual se desarrollará material de apoyo correspondiente al plan de estudios 2016. Con base en dicha investigación se buscó la explicación de dos temas seleccionados que se darán de forma sencilla y útil, así abarcando las necesidades básicas del temario.

  - *Animaciones*

    Generación de material visual y audiovisual, que apoye y complemente el desarrollo de los temas referentes a la asignatura de Criptografía. Desde animaciones simples que complementen conceptos, hasta videos explicativos en donde se trate un tema específico, enfocados en los sub-temas de las últimas unidades de la asignatura de Criptografía plan de estudios 2016.

  - *Actividades*

    Creación e implementación de recursos didácticos simples, con el objetivo de reforzar lo aprendido de forma teórica durante el curso.

  - *Material de consulta*

    Recopilación de información con respecto a sub temas de la asignatura de Criptografía que serán proporcionados de manera pública, para poder ser consultados por alumnos que cursen la materia y personas interesadas.

4. Descripción del material didáctico.

En esta sección se describirá la forma de uso y administración de la plataforma, al igual se hablará del material didáctico realizado de este proyecto.

4.1. Manual de administración

El administrador es aquel que puede modificar el contenido del curso, la apariencia de la plataforma, otorgar acceso a la plataforma, entre otras cosas. La siguiente sección contiene lo necesario para administrar la plataforma.

Los pasos para la apariencia y funcionalidad de la plataforma deberán ser seguidos cada que se instale una nueva instancia de Moodle, después de eso no habrá necesidad de modificar esos procesos, en este manual se muestran dichos pasos. Por otro lado, la administración de usuarios y la creación y edición de contenido son cambios contantes que se deben realizar por el administrador, de igual manera los pasos a seguir se mostrarán a continuación en el manual.

Para acceder y administrar la aplicación de Moodle se debe ingresar usando la cuenta de administrador que fue creada durante el proceso de instalación de Moodle.

Moodle permite crear varios tipos de usuarios, pero para cuestión del documento solo existen dos, los usuarios y los administradores, que son aquellos que tienen el control de toda la plataforma, creación/modificación/eliminación de usuarios, creación/modificación/eliminación de contenido entre otras cosas. Los usuarios son aquellos como los profesores y alumnos, cada uno con sus restricciones.

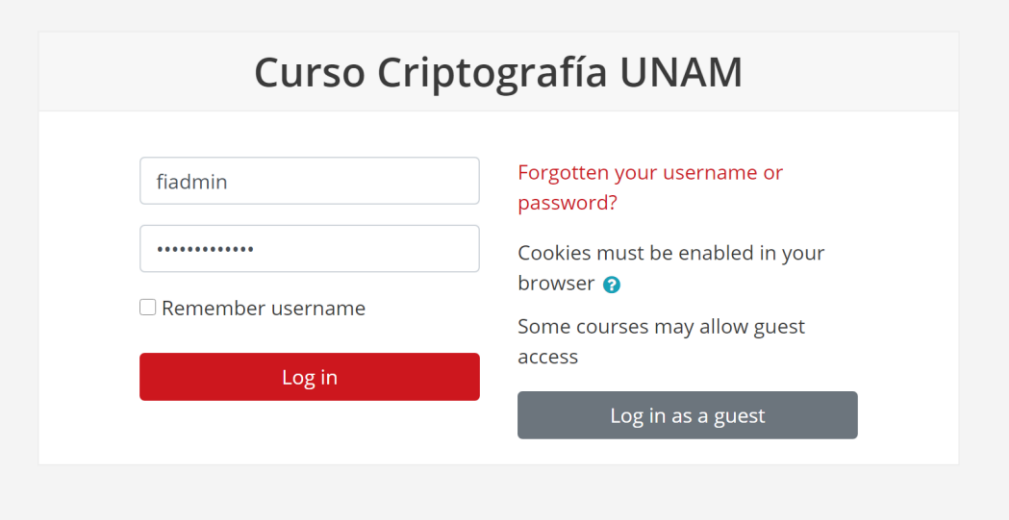


Fig. 2 Pantalla de autenticación de ingreso a plataforma

4.1.1. Modificación de estilo

Antes de comenzar a agregar contenido es necesario darle un estilo propio a la plataforma de Moodle, para esto se ingresa a la pestaña de *Administración de Sitio* en el lado izquierdo de la página:

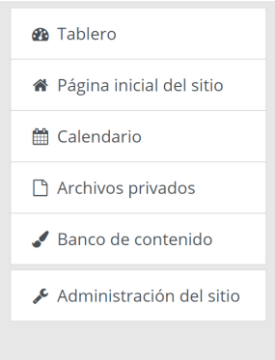


Fig. 3 Barra lateral de perfil de administrador

Bajo la pestaña de *Apariencia* realizaremos cambios a tres subpestañas:

- HTML adicional
- Moodle Docs
- Temas → Boost (Impulso)

Fig. 4 Menú de apariencia de Moodle

Primero ingresamos a *Moodle Docs* y borramos la liga de por defecto en "Raíz de Moodle Docs", guardamos los cambios y regresamos a la pestaña de apariencia.

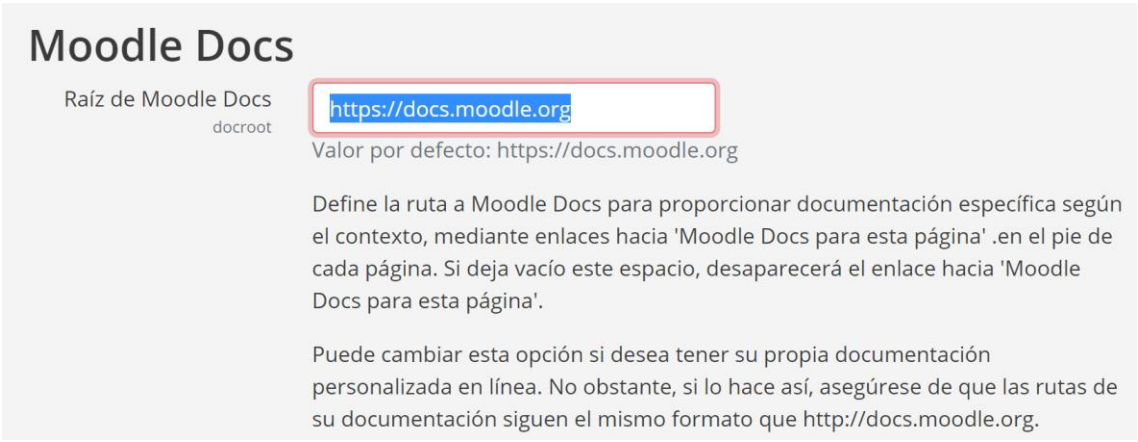


Fig. 5 Pestaña de Moodle Docs

Seguimos a *HTML Adicional* donde agregaremos el siguiente código HTML a la sección de "Antes de cerrar BODY" este código es el pie de página del sitio web, donde se pondrá las direcciones de la página web de la Facultad de Ingeniería y de la UNAM:

```
<div>
    <div style="font-size: 14px;line-height: 1.8;color: #fff;">
        Todos los derechos reservados © 1999 - 2020 /
        <a href="http://www.ingenieria.unam.mx">Facultad de Ingeniería</a>/<a href="ht
tp://www.unam.mx">UNAM</a>/
    </div>
    <div style="font-size: 11px; line-height: 1.2; color: #fff;">
        <p>Esta página electrónica es parte de la Facultad de Ingeniería de la UNAM. P
uede ser reproducida con fines no lucrativos, siempre y cuando no se mutile, se cite l
a fuente completa y su dirección electrónica.
        </p>
    </div>
</div>
```

Guardamos los cambios y regresamos a la pestaña de *Apariencia.*

Por último, modificaremos la apariencia del sitio (color, texto, header, etc.), para esto es necesario crear un archivo en cualquier editor de texto en la computadora, el archivo debe ser guardado con la extensión **.scss** con el código encontrado en el Anexo del documento **Código de estilo para la plataforma p. 132**

Una vez generado el archivo .scss podemos subirlo a la plataforma, para esto debemos ingresar a la pestaña de *Temas → Boost (Impulso),* arrastrando el archivo o seleccionándolo desde el explorador de archivos hacia la sección de "Archivos de preconfiguración adicional del tema" una vez se suba el archivo se deben guardar los cambios y posteriormente se debe seleccionar el nombre del archivo en la sección de "Preconfiguración del tema"

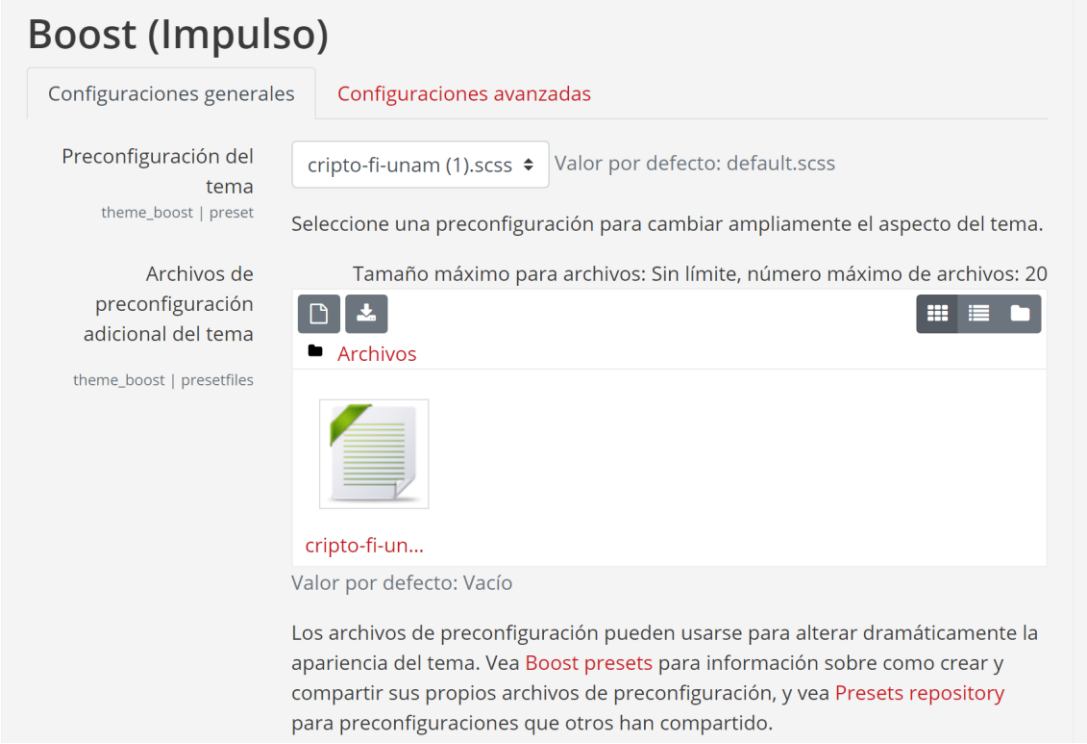


Fig. 6 Sección de Tema para la plataforma de moodle

Por último, modificamos en la misma pestaña en la sección de "Color de marca" el color por defecto por el siguiente "#cd171e" una vez hecho eso, se deben guardar los cambios.

Fig. 7 Cambio de color en el tema de la plataforma

Al volver a cargar el sitio web se deberán ver los cambios que se realizaron y la apariencia debería ser similar a la siguiente:

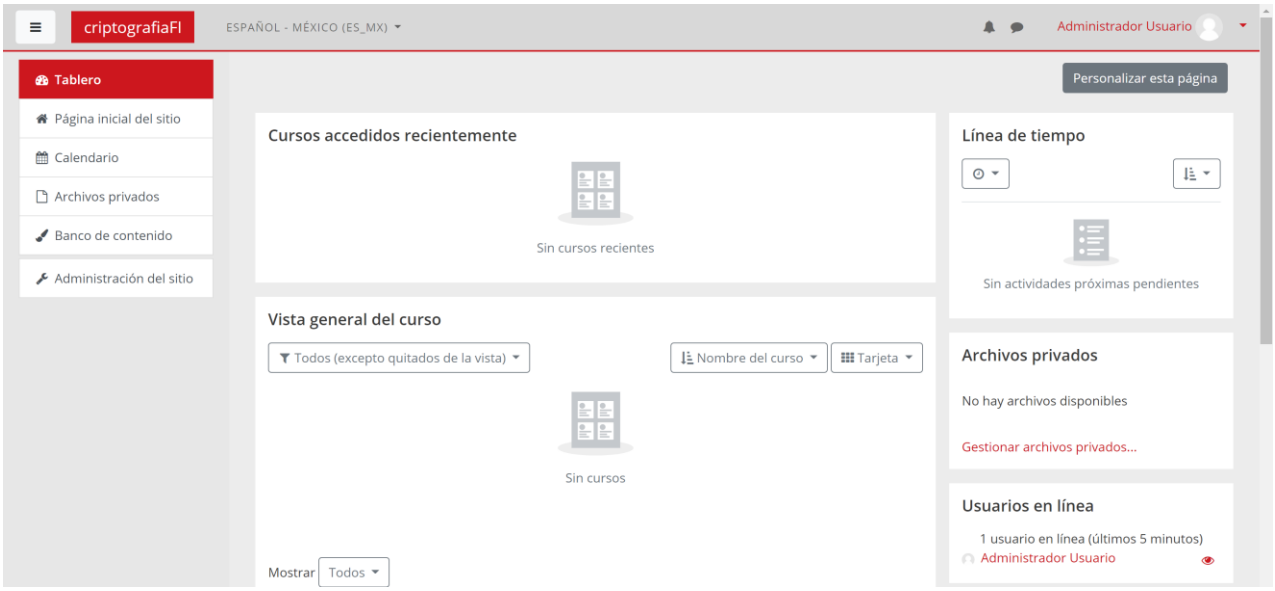


Fig. 8 Vista de la plataforma con la apariencia modificada

4.1.2. Instalar Plug-in

Los plug-ins son componentes de software que agregan una característica a un programa de computación existente. Creados para extender las funcionalidades con las que cuenta un programa base.

Moodle nos permite el uso de extensiones gratuitas que expanden la funcionalidad de la plataforma y ayuda a adaptarlas a las necesidades de cada curso. En el caso del curso de Criptografía se añadieron las extensiones de *H5P* y *Game module*, las cuales son caracterizadas por añadir a la plataforma actividades didácticas-interactivas que son avaladas por expertos y serán utilizadas en este proyecto para la creación de actividades más adelante. Estos se descargan de las siguientes ligas:
https://docs.moodle.org/all/es/Game_module
https://moodle.org/plugins/mod_hvp

Para instalar los plugins debemos modificar el servidor (revisar manual de instalación), ingresando por medio de SSH al servidor, una vez dentro debemos ejecutar los siguientes comandos para permitir la modificación de Moodle:

```
sudo chmod -R 777 /var/www/html/moodle
```

Una vez terminada la configuración por medio de la interfaz **SE DEBE REGRESAR a los permisos anteriores**:

```
sudo chmod -R 0755 /var/www/html/moodle
```

Con los permisos en el directorio del servidor podemos ir a la pestaña de *Administración de Sitio* y luego en la subpestaña de plugins seleccionamos “instalar plugin”

Fig. 9 Pestaña de plugins

Con los archivos de los plugins que descargamos en la sección anterior podemos subir los archivos .zip, una vez se hayan cargado los archivos seleccionamos la “instalación de plugin desde archivo ZIP”. Esperamos que se realice la carga y los plugins se deben de instalar sin mayor problema.

Fig. 10 Pantalla de instalación de plugins

Para comprobar la instalación podemos ir a la pestaña de vista general de plugins y verificar que el plugin se encuentre activo:

Fig. 11 Pantalla de plugins instalados

Una vez que termines la instalación recuerda **REGRESAR a los permisos anteriores** en el servidor:

```
sudo chmod -R 0755 /var/www/html/moodle
```

4.1.3. Control de usuarios

El uso del sitio esta restringido a todo aquel que tenga una cuenta. La creación de usuarios tiene varias formas; Auto-registro, registro individual manual o creación de cuentas de forma masiva mediante archivos CSV, entre otras opciones. Para nuestro caso nos enfocaremos en creación de cuentas de forma individual y de forma masiva.

Fig. 12 Pestaña de Usuarios en administración de sitio

Cuentas individuales

Para la creación de cuentas de forma individual nos dirigimos a la pestaña de *Administración del sitio*, en la subpestaña de *Usuarios* veremos la categoría de *Agregar usuario.*

Una vez dentro de Agregar Usuarios, podemos ingresar los datos de un nuevo usuario. Esto se recomienda para la creación de usuarios administradores o de tipo profesor. Los cuales son pocos y se puede tener el control de ellos.

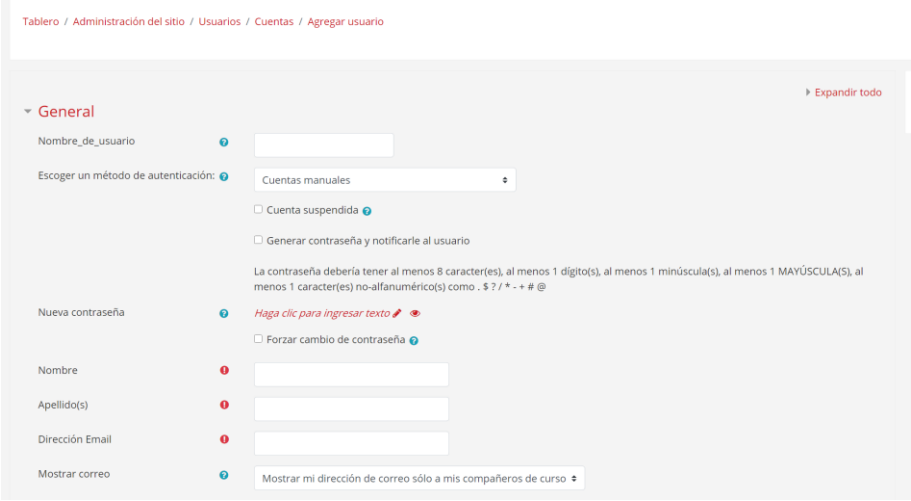


Fig. 13 Página de creación de usuario

Generación masiva de cuentas

La generación masiva de usuarios se recomienda cuando se agregan a los estudiantes o es necesario modificar varios usuarios al mismo tiempo. En esta sección es posible generar grandes cantidades de usuarios, inscribirlos a cursos, modificarlos, entre otras cosas.

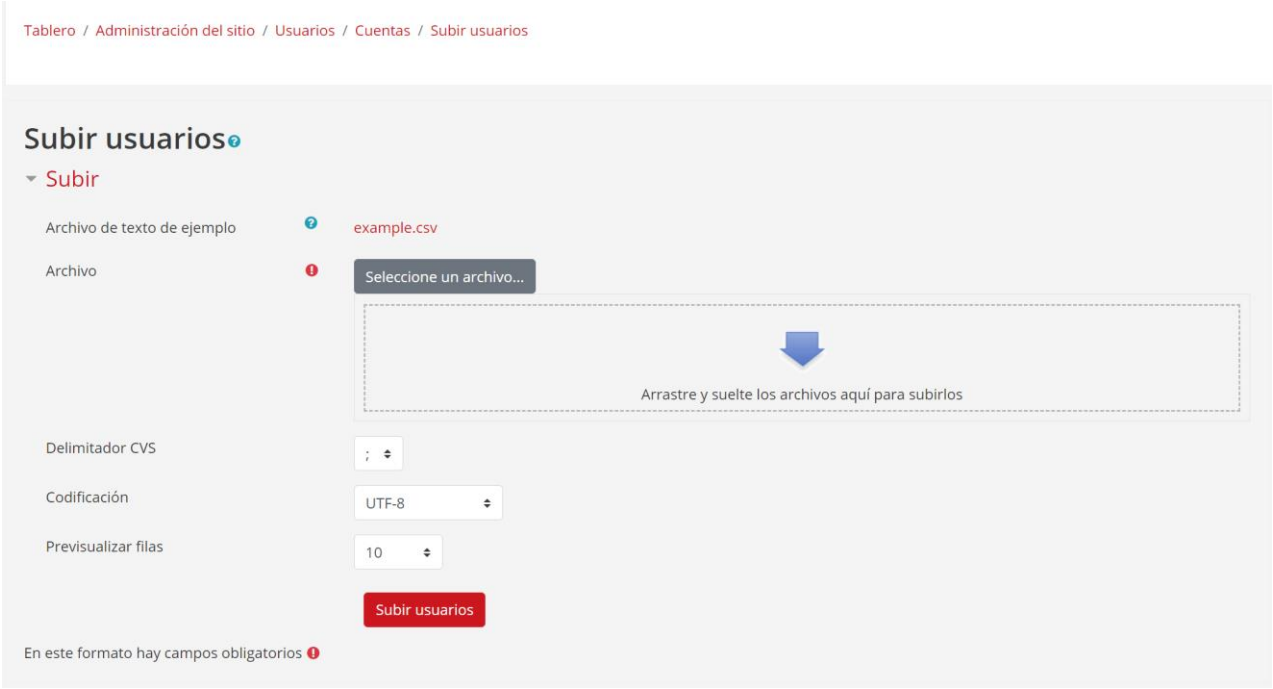


Fig. 14 Página de gestión masiva de usuarios

Para subir usuarios es necesario generar un archivo de tipo **.csv** Moodle nos ofrece un ejemplo del archivo en la misma sección de *Subir usuarios.* Para mayor información de las opciones para subir usuarios visitar la página de documentación de Moodle:
(https://docs.moodle.org/all/es/38/Subir_usuarios)

Roles de Usuarios

Moodle nos permite tener un control en los roles de usuario, entre ellos el rol de administración. Para dar permisos a un usuario de administrador nos dirigimos a la pestaña de *Administración del sitio*, en la subpestaña de *Usuarios* veremos la categoría de *Permisos y Administradores del sitio:*

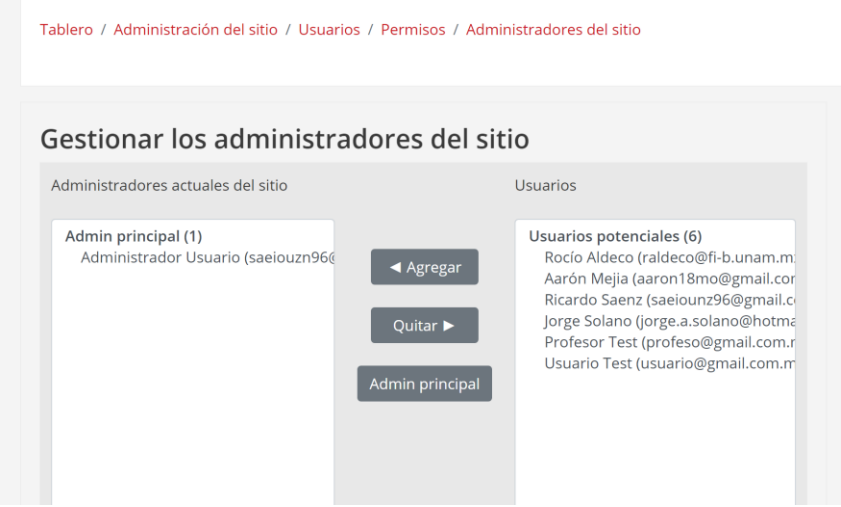


Fig. 15 Definir administradores del sitio

Cualquier usuario dentro de los administradores de sitio tendrá las mismas capacidades que el administrador original. Por lo que hay que ser cuidadoso al dar este rol a usuarios.

Para definir los roles de profesor/generador de contenido/estudiante de los usuarios dentro de un curso es necesario haber creado el curso. Una vez generado el curso podemos ingresar a las preferencias y seleccionamos *Más…*:

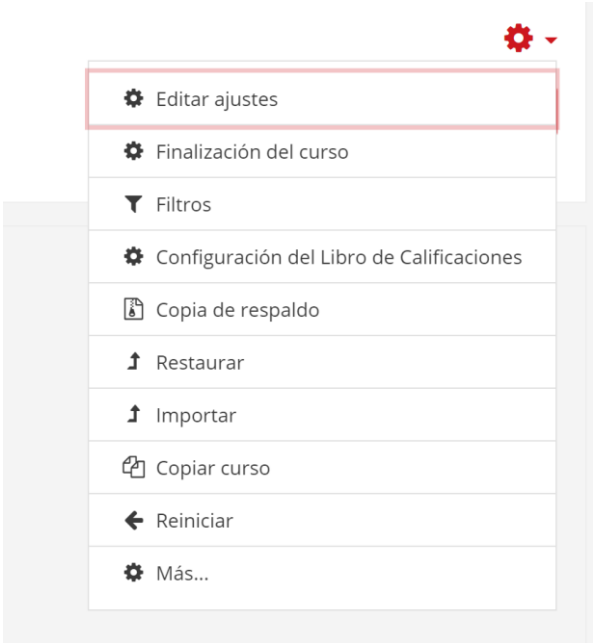


Fig. 16 Preferencias del curso

En dicha sección podemos agregar los roles a los usuarios que tendrán acceso al curso y únicamente a las funciones del curso, para esto seleccionamos la pestaña de *usuarios inscritos:*

Fig. 17 Página de administrador de curso

Dentro de Usuarios inscritos podemos seleccionar *Inscribir usuarios* donde de forma manual podemos seleccionar usuarios y asignarles roles:

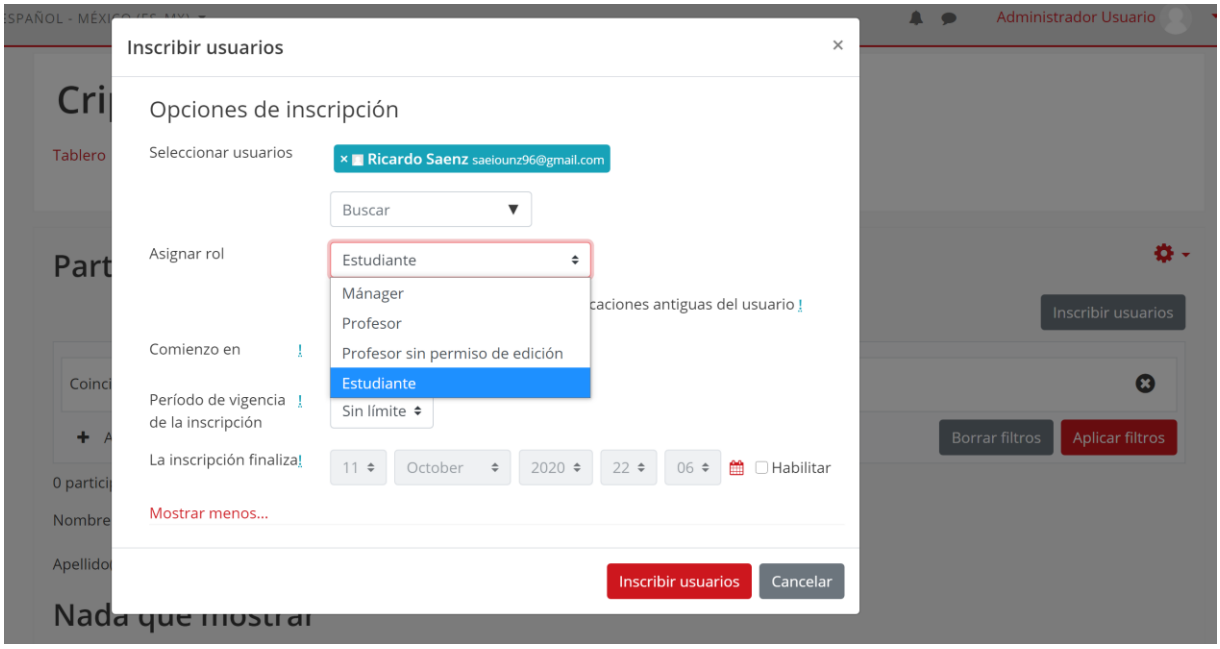


Fig. 18 Página de inscripción de usuarios y asignar roles al curso

4.1.4. Curso

Para poder empezar a subir contenido es necesario crear un curso, en Moodle permite crear varios cursos dentro de la plataforma, en este caso solo vamos a hacer uno “Criptografía”. Para esto nos

dirigimos a la pestaña de *Administración del sitio*, en la subpestaña de *Curso* veremos la categoría de *Gestionar cursos y categorías:*

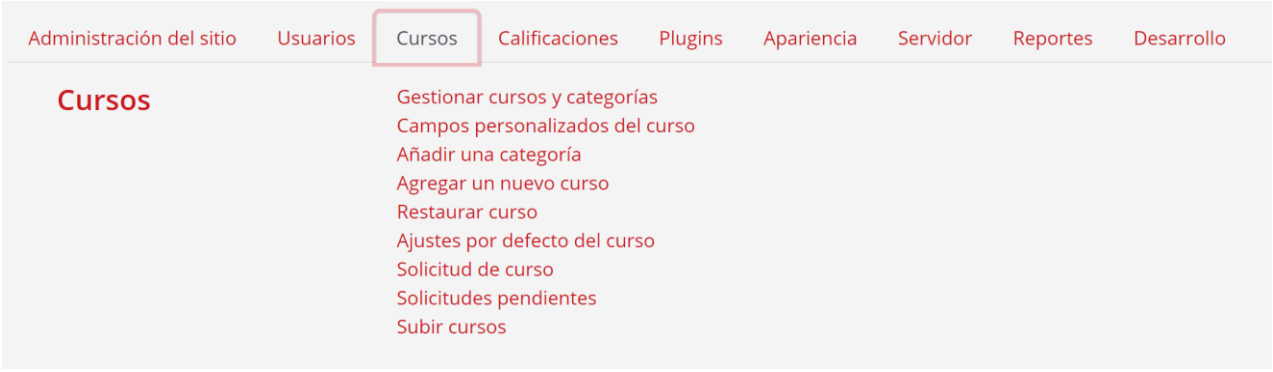


Fig. 19 Pestaña de curos

Normalmente Moodle permite tener varios cursos y dichos cursos se pueden gestionar en diferentes categorías, debido a que nosotros solo tendremos un curso usaremos la categoría por defecto de Moodle llamada **Misceláneos** para guardar nuestro curso.

Dentro de la categoría de Misceláneos podemos crear un nuevo curso:

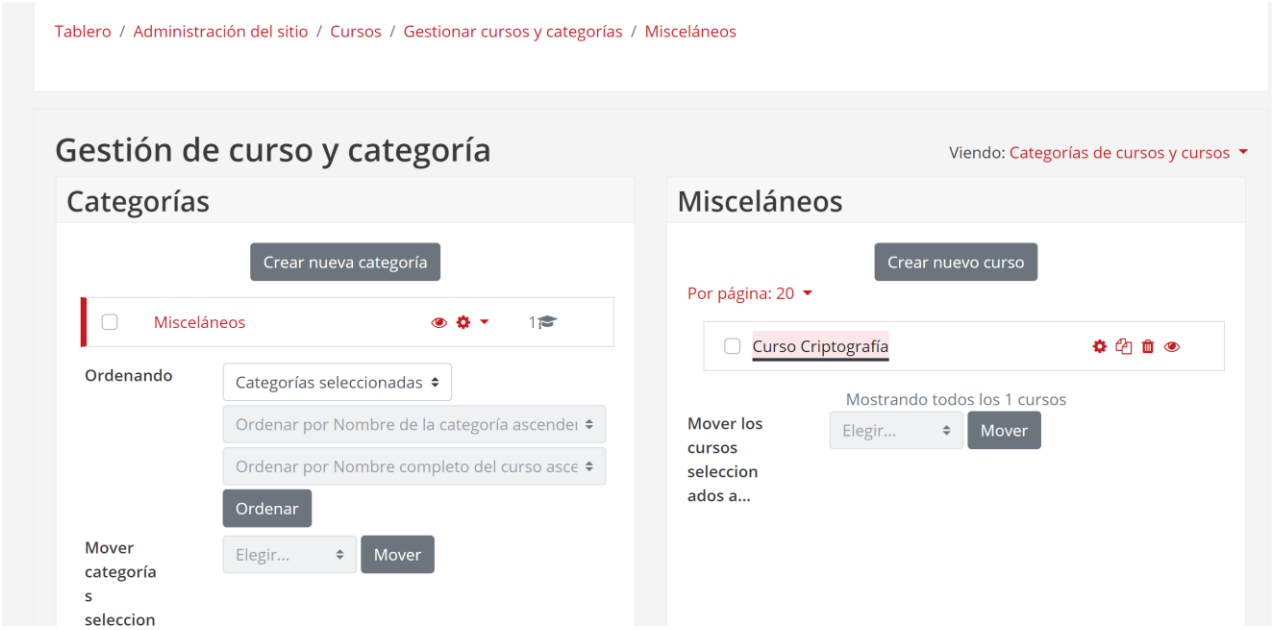


Fig. 20 Creación y gestión de cursos

Al agregar un nuevo curso nos pedirá los datos básicos para el curso; Nombre completo del curso, nombre corto del curso y una descripción.

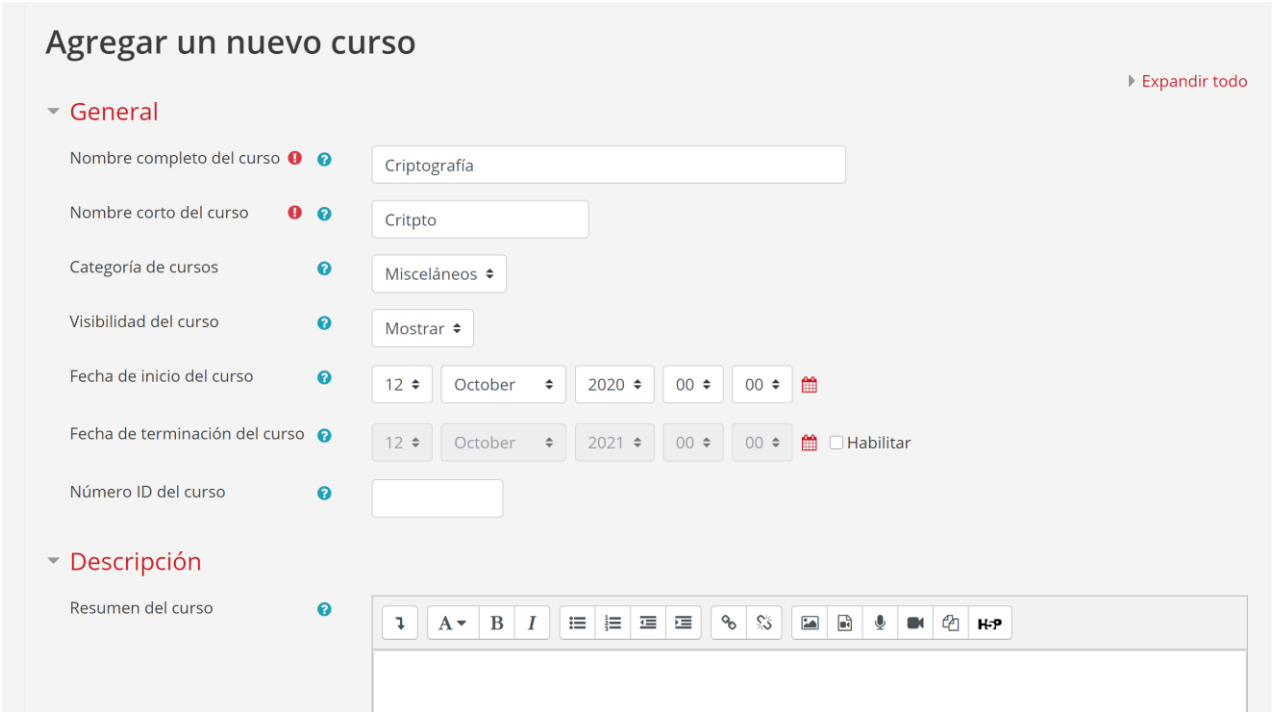


Fig. 21 Creación de nuevo curso

Al terminar de llenar los datos del curso presionamos “Guardar cambios y mostrar” en la parte inferior de la ventana, lo cual nos llevara a la sección para añadir participantes y ahí podemos movernos al nuevo curso que creamos y podremos comenzar a añadir elementos.

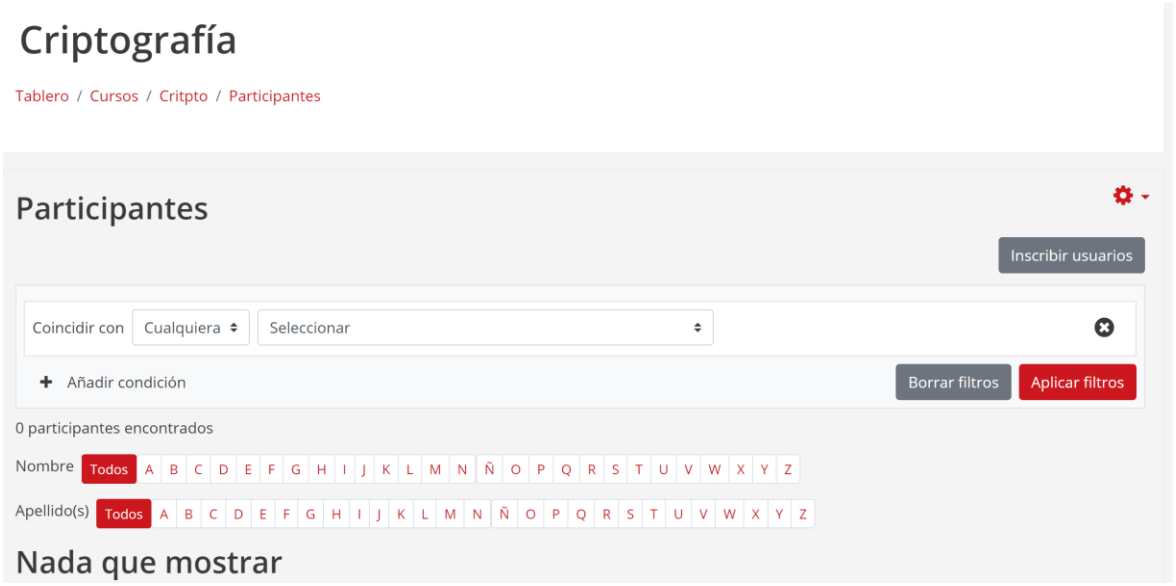


Fig. 22 Participantes del curso

Un nuevo curso se debe de ver como la siguiente pantalla, una vez esto se completo el proceso de creación de un curso:

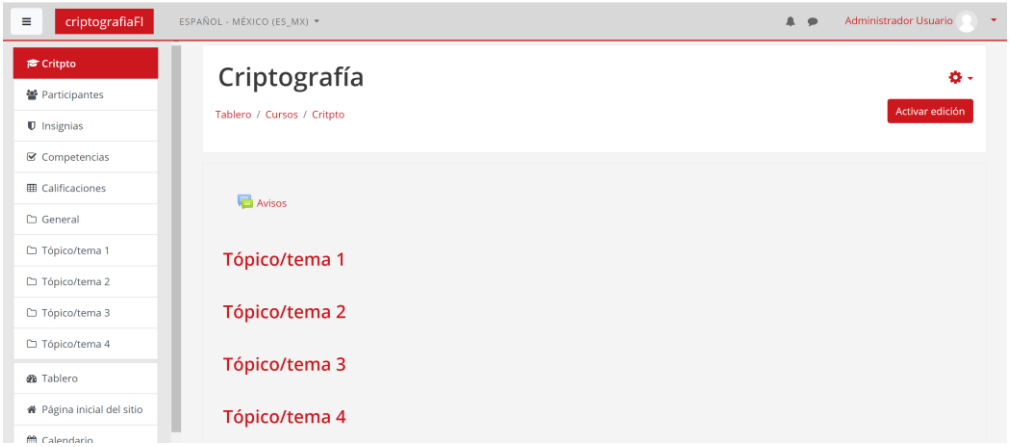


Fig. 23 ventana de un curso recién creado

4.1.5. Temas

Para la creación de temas en un curso podemos seleccionar la *Activación de edición* dentro del curso:

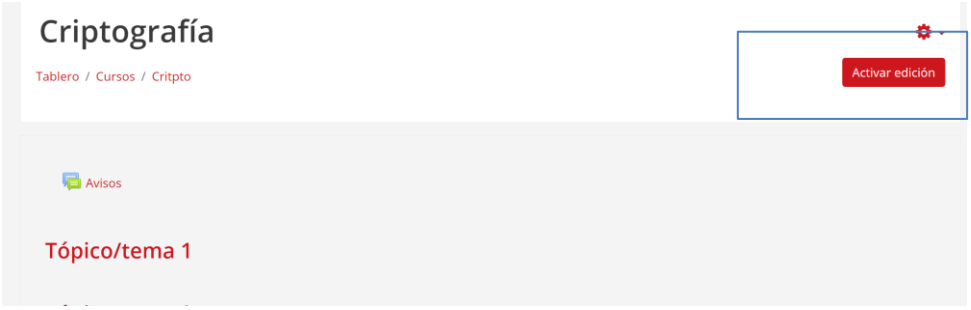


Fig. 24 botón de edición

Al activar la edición podemos modificar, agregar, eliminar los temas y actividades del curso. En este caso nos enfocamos en las secciones de temas.

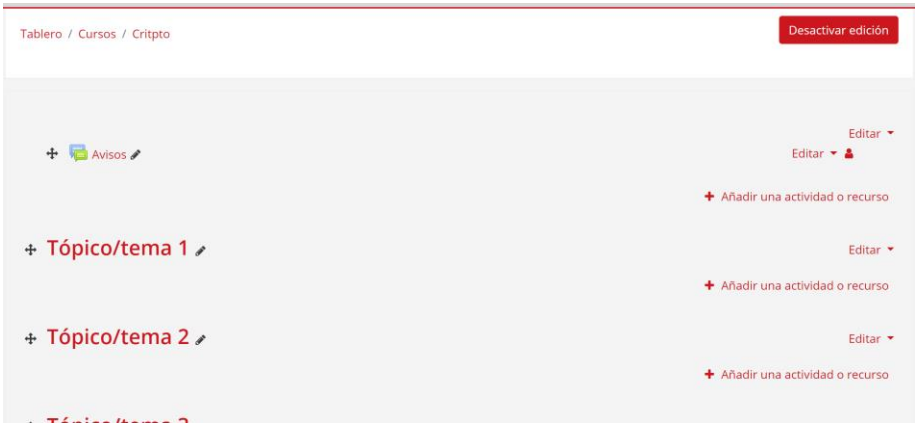


Fig. 25 Visión de edición en curso

Para la edición de nombre podemos seleccionar el icono del lápiz al lado del nombre del tema, ahí nos da la capacidad de cambiar el nombre:

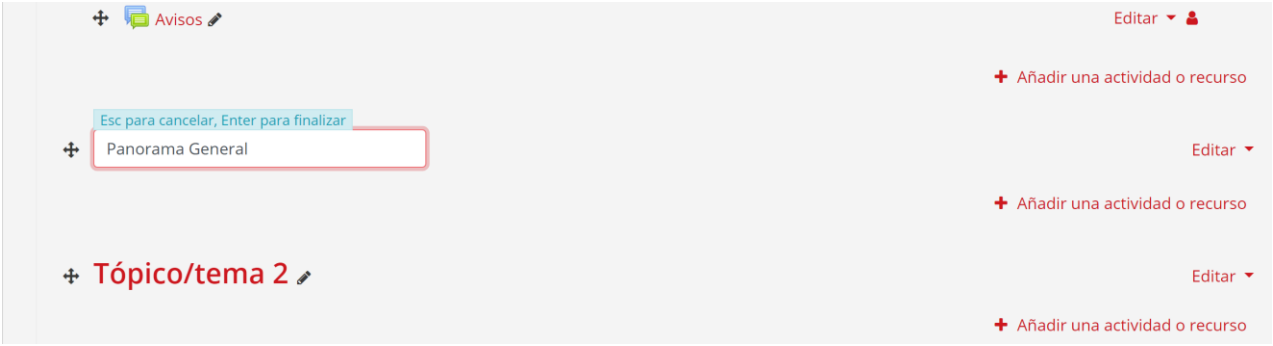


Fig. 26 Edición de nombre en un tema

Si se quiere agregar un nuevo tema, en la parte inferior de la página se encuentra el botón que lo permite "Añadir tópico"

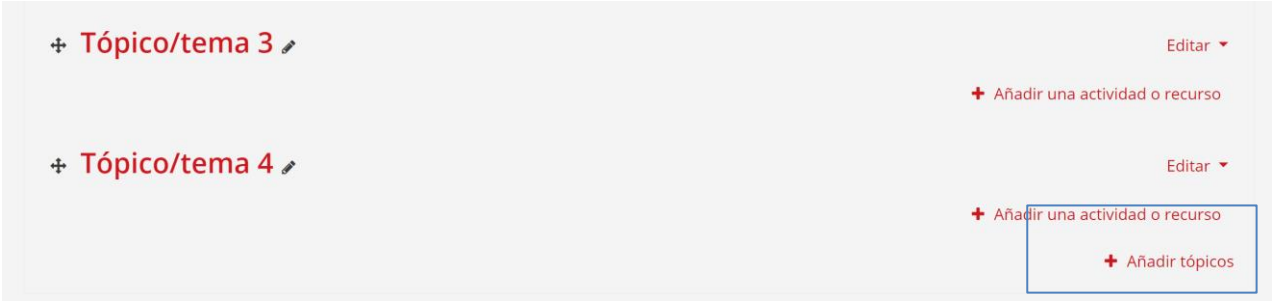


Fig. 27 Botón de añadir tópico

4.1.6. Páginas

Las páginas son la fuente de información para los temas del curso, para añadir nuevas páginas seleccionamos el botón de "añadir una actividad o recurso" que se encuentra al lado derecho de los tópicos:

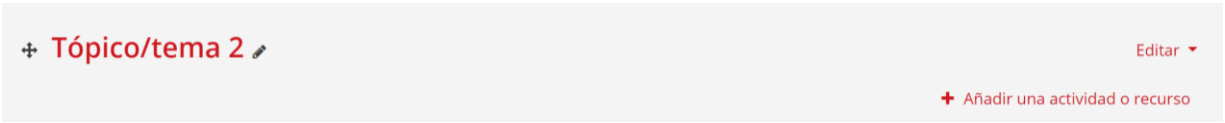


Fig. 28 Botón de añadir recurso

Una vez abierta la pestaña de recursos buscamos la opción de "página" si únicamente se quiere hacer una página de algún tema o la opción de "Lección" si se quiere hacer varias páginas para explicar un tema.

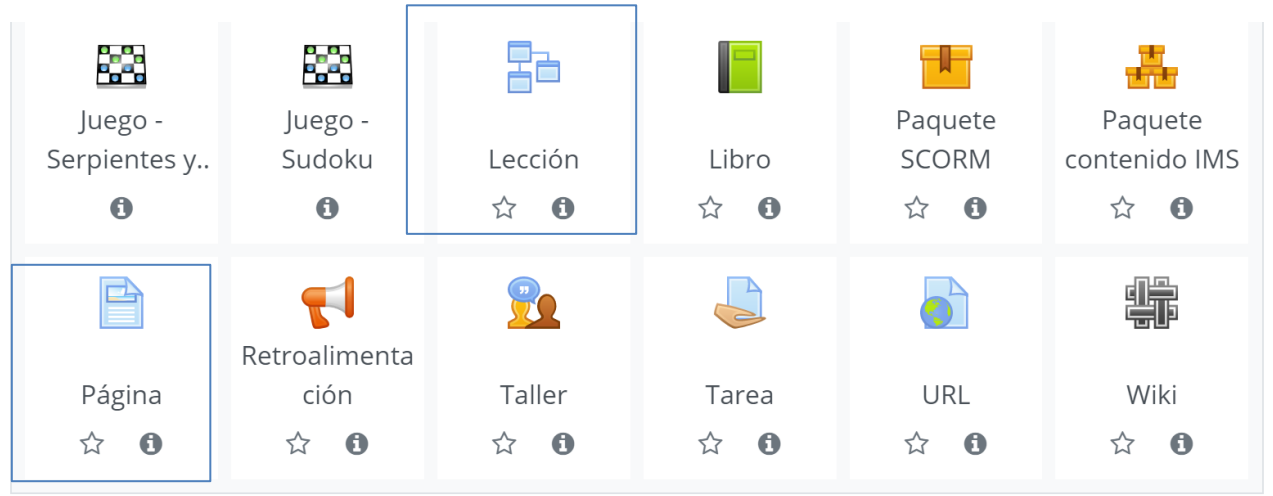


Fig. 29 Pestaña de selección de recursos

En este caso generaremos una nueva lección, para esto seleccionamos “Lección” y dentro le asignamos un nombre y si lo deseamos una descripción. Al terminar guardamos y mostramos cambios:

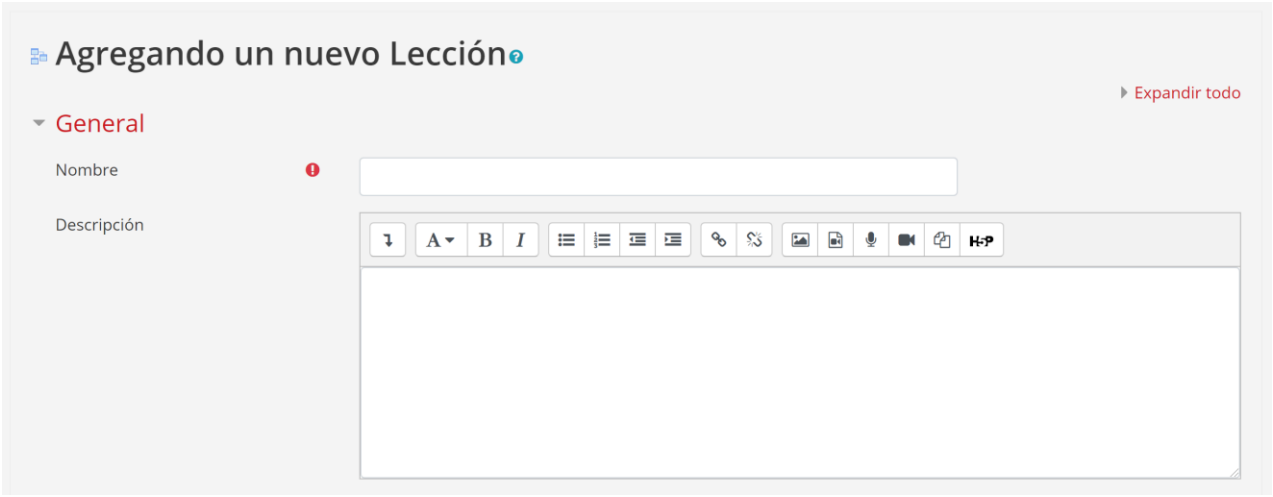


Fig. 30 Generar nueva lección

Una vez generada la lección podemos ir agregando actividades o páginas a la lección, para esto nos vamos a la pestaña de *Edición* en modo *colapsado* para una mejor visualización y comenzamos a añadir las páginas utilizando el *dropdown* del lado derecho:

Fig. 31 Página de lección para añadir páginas

Dentro de cada página de la lección añadimos el contenido de la lección y las vamos ligando con respecto a las siguientes páginas:

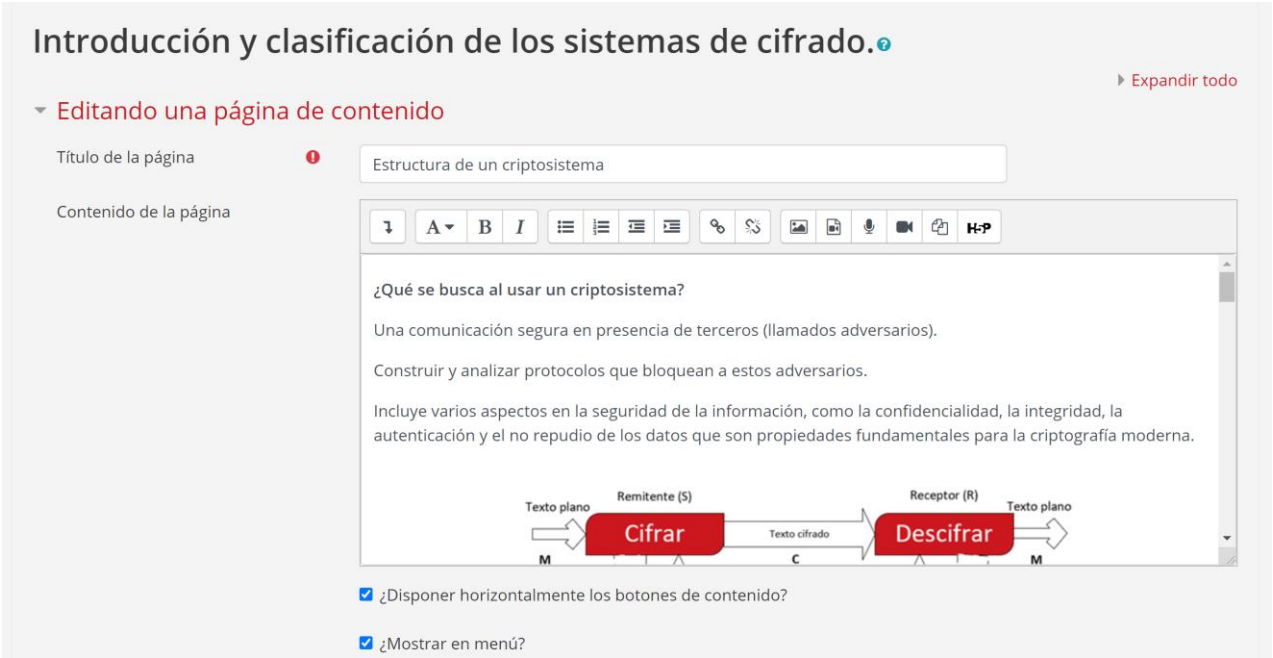


Fig. 32 edición de contenido de una página

Al terminar de agregar el contenido guardamos y podemos tener una previsualización del contenido, seguido podemos regresar a la página principal del curso para ver la lección como se vería por un estudiante:

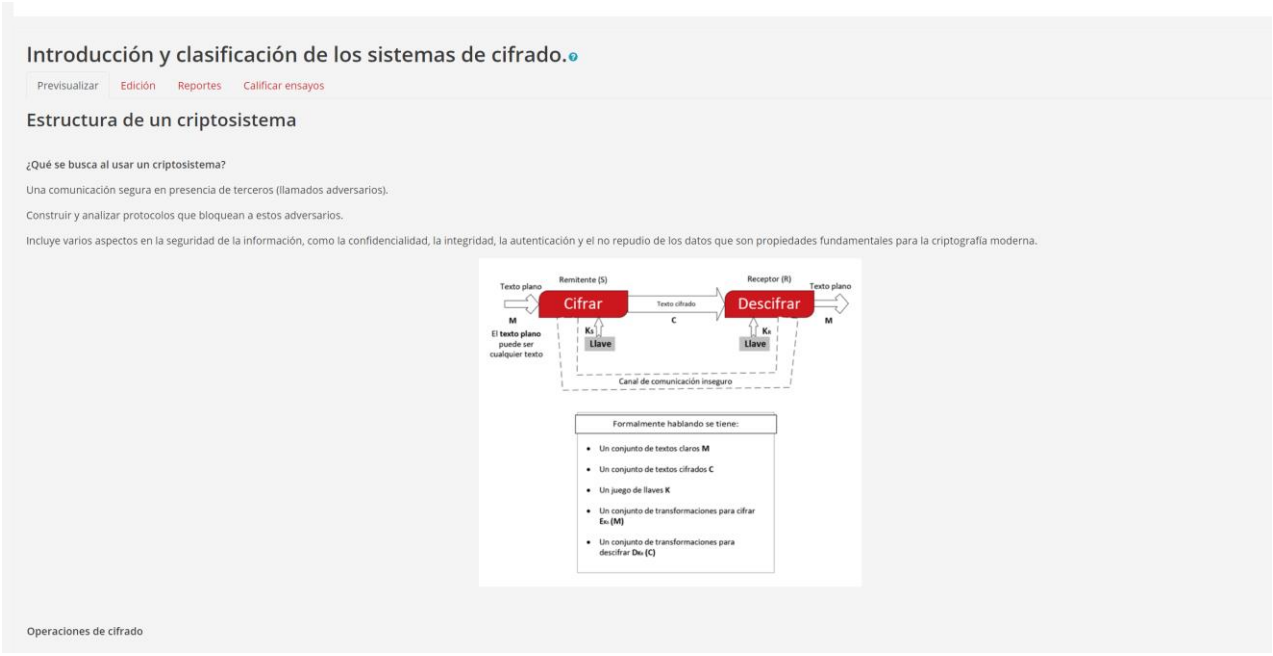


Fig. 33 Previsualización de una página de lección

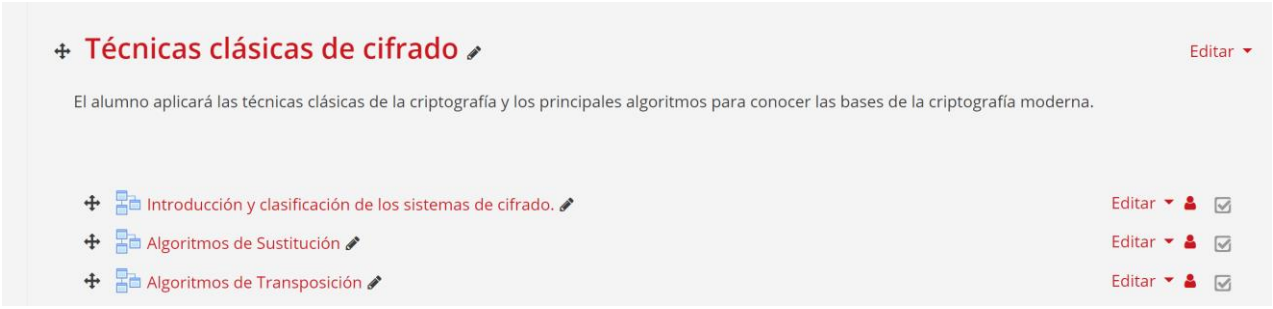


Fig. 34 Previsualización de un Tema con lecciones añadidas a el

4.1.7. Evaluaciones y exámenes

Abrimos la pestaña de recursos buscamos el de “Examen”, este recurso lo podemos usar tanto como autoevaluaciones o como exámenes, dependiendo de las necesidades:

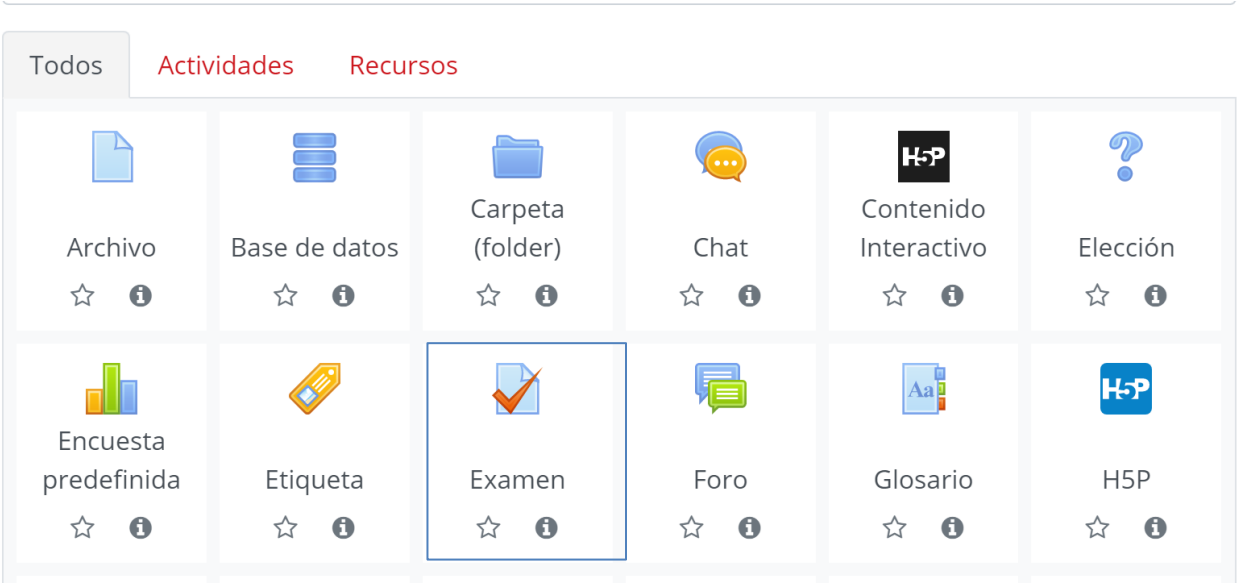


Fig. 35 Pestaña de selección de recursos

Dentro del Examen podemos modificar todo aspecto del mismo, pero es necesario crear un nombre y una descripción del mismo:

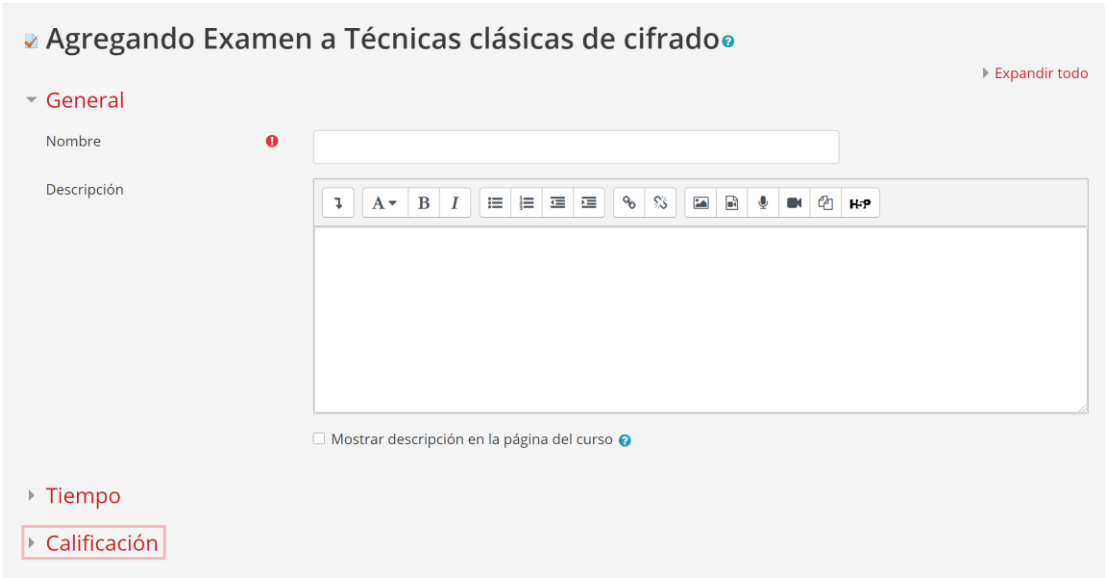


Fig. 36 Generar un examen

Al ingresar al examen podemos añadir preguntas dentro de categorías, para esto se selecciona la edición del examen y en la parte inferior presionar el botón de **añadir:**

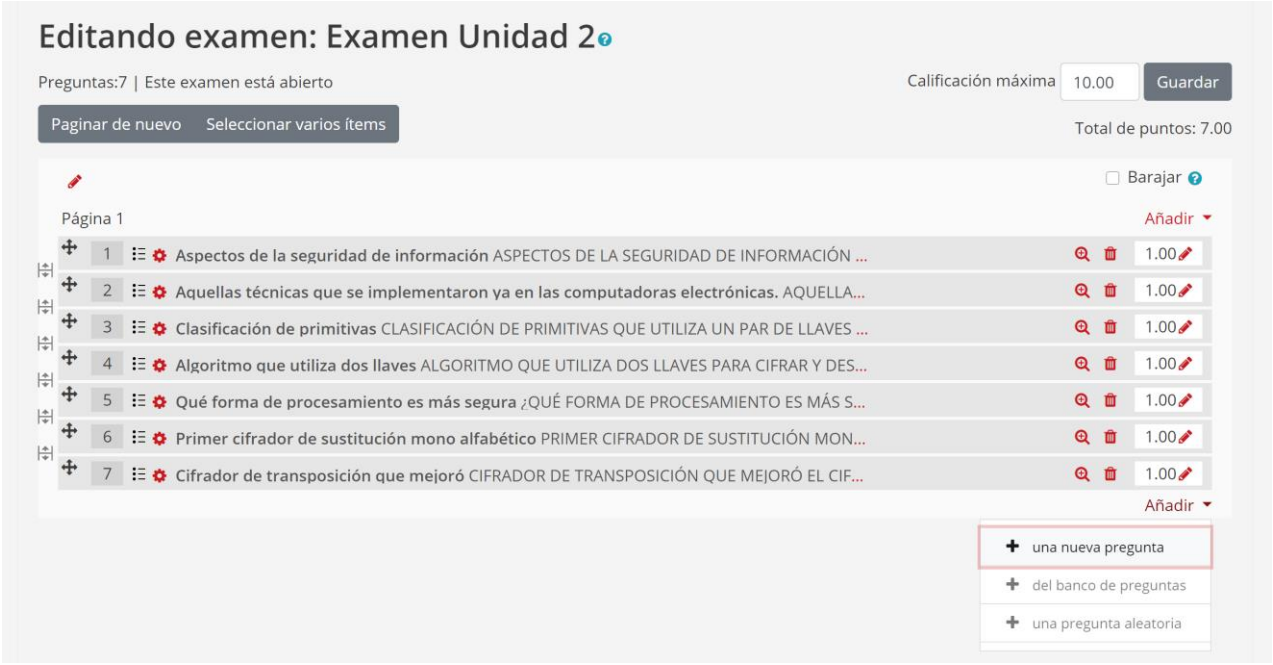


Fig. 37 Pestaña de edición de examen y añadir nueva pregunta

Una vez dentro podemos seleccionar entre varios tipos de preguntas:

Fig. 38 Selección de tipo de preguntas

Dependiendo del tipo de pregunta seleccionada saldrán diferentes formas de llenado y evaluación, en todas se debe seleccionar a que grupo de preguntas pertenece y se puede poner retroalimentación de las respuestas:

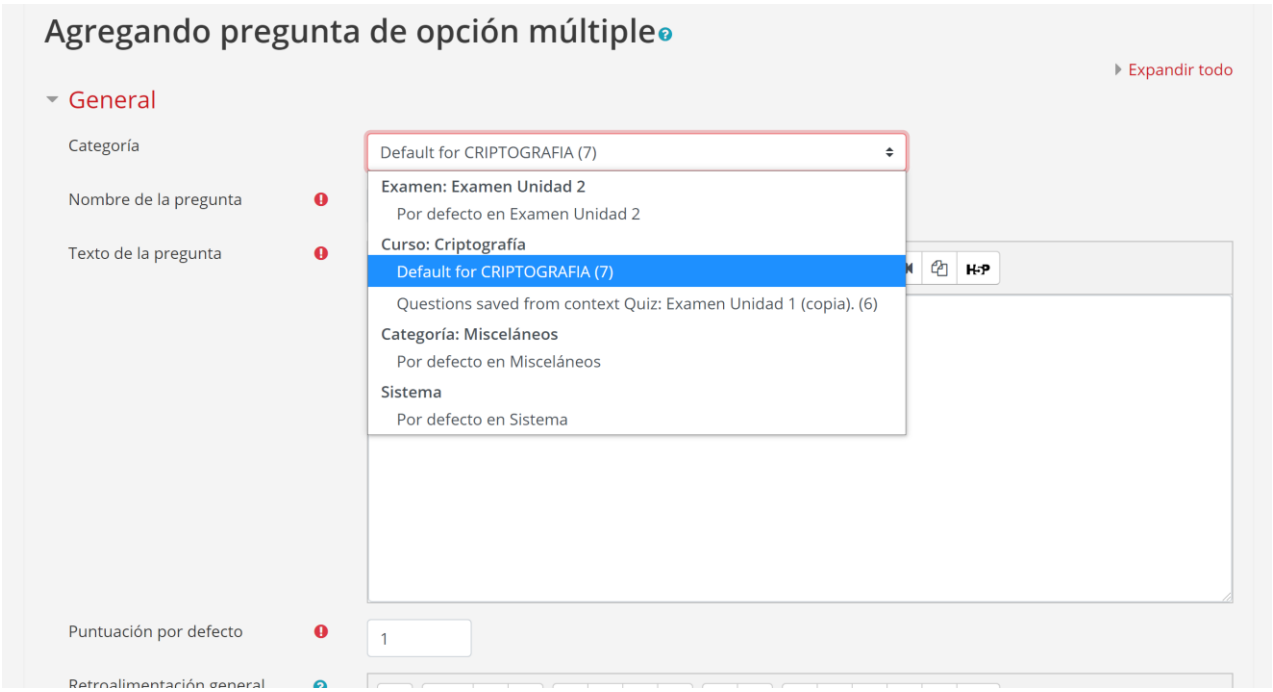


Fig. 39 Creación de una pregunta para examen

Con el examen creado podemos visualizarlo y comprobar el funcionamiento del mismo:

Fig. 40 Previsualización de un examen o evaluación

4.1.8. Ejercicios

Abrimos la pestaña de recursos buscamos el de “Contenido Interactivo” este recurso lo podemos usar con el fin de agregar contenido interactivo o dinámico a el curso, de igual manera el contenido creado con este recurso puede ser evaluado:

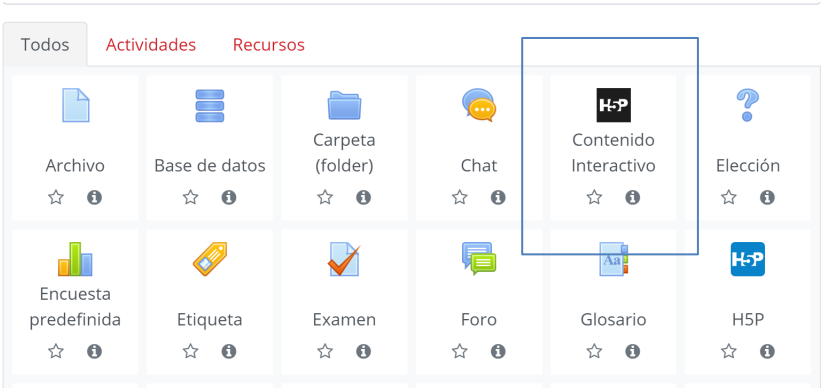


Fig. 41 Pestaña de selección de recursos

Estos recursos son de gran utilidad para generar contenido más inmersivo para la plataforma, entre ellos podemos encontrar juegos, actividades, publicación de contenido multimedia, entre otras cosas:

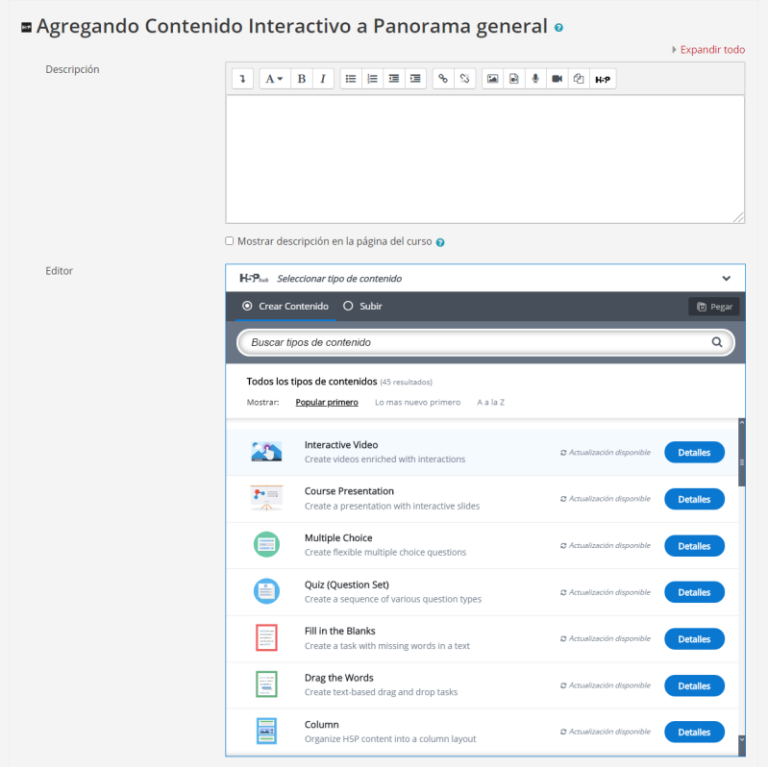


Fig. 42 Recurso de contenido dinámico

Para trabajos que requieren un entregable podemos seleccionar el recurso de *Tareas.* Para esto regresamos al curso y abrimos la pestaña de recursos buscamos el de “Tareas”:

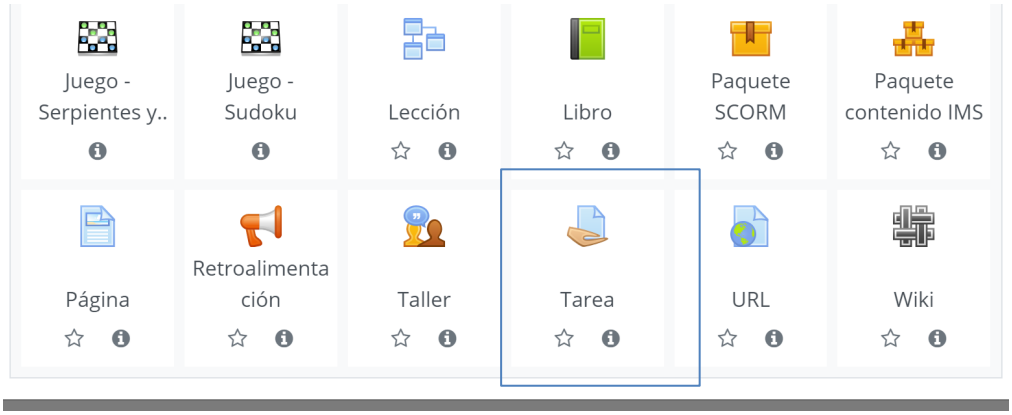


Fig. 43 Pestaña de selección de recursos

Con este recurso es posible crear actividades con entregables los cuales serán evaluados por profesores de forma manual:

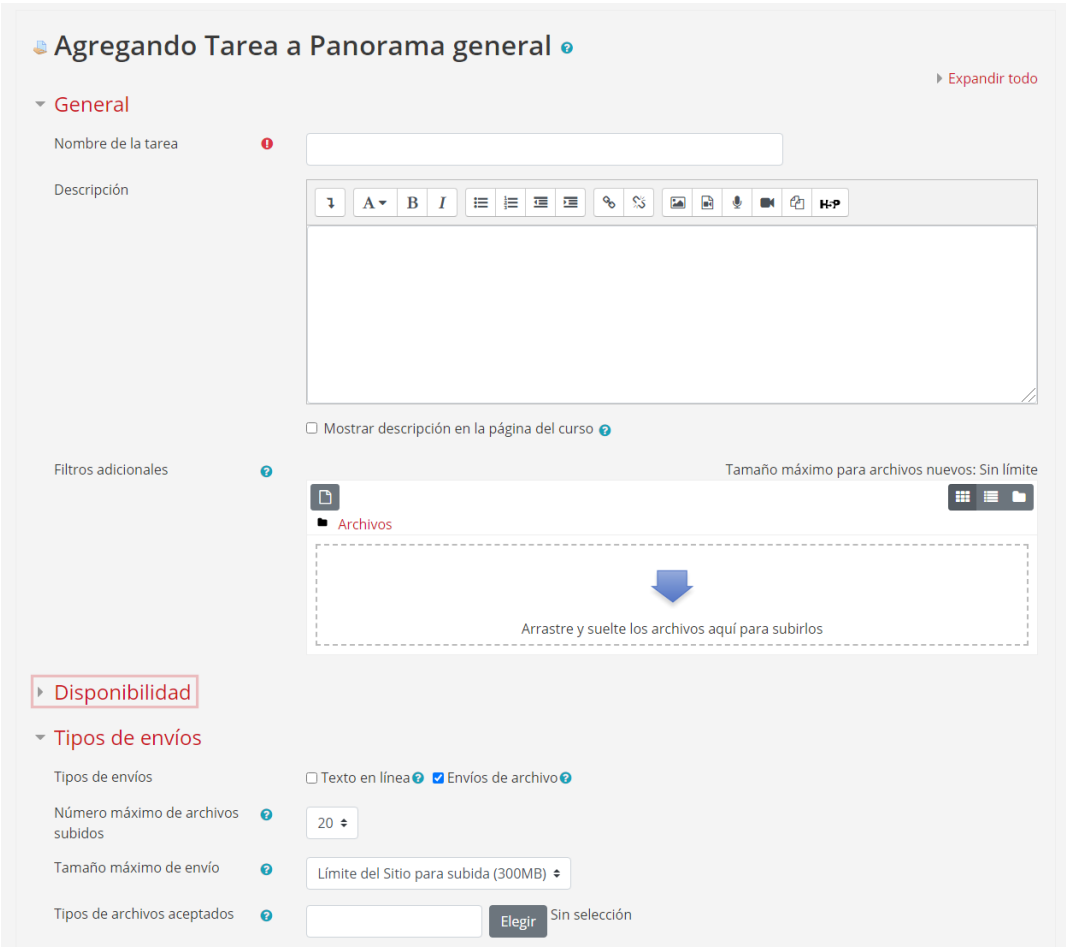


Fig. 44 Creación de tareas para entregables

4.1.9. Administrar archivos

Para la gestión de archivos es necesario poner límites y tamaños máximos para así no sobrecargar el servidor, esto se puede hacer desde la pestaña de *Administración del sitio* en la subpestaña de seguridad y *políticas de seguridad del sitio*. Aquí se pueden modificar todas aquellas preferencias que son referentes a la privacidad de los usuarios y la seguridad de la información del sitio y usuarios.

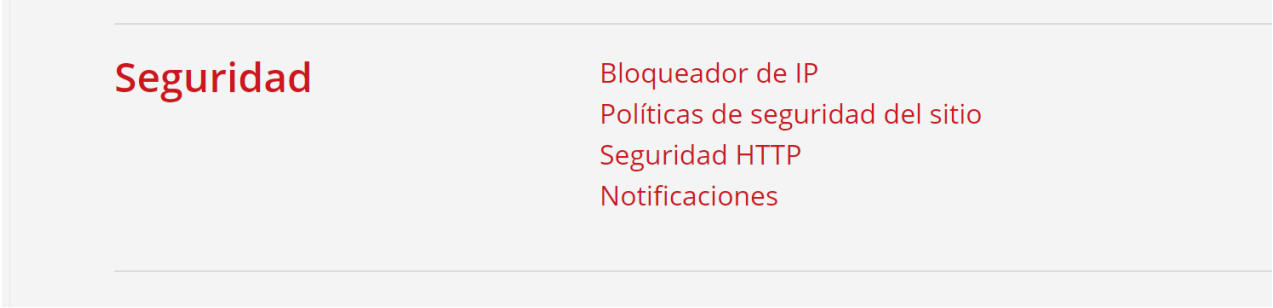


Fig. 45 Pestaña de seguridad dentro de Administración de Sitio

En la sección de tamaño máximo y espacio para archivos privados los modificamos para adecuarse a las necesidades del curso. En el curso no se espera que los alumnos suban archivos de mayor tamaño a 10 MB, una vez modificados los valores se deben guardar los cambios en la parte inferior de la ventana.

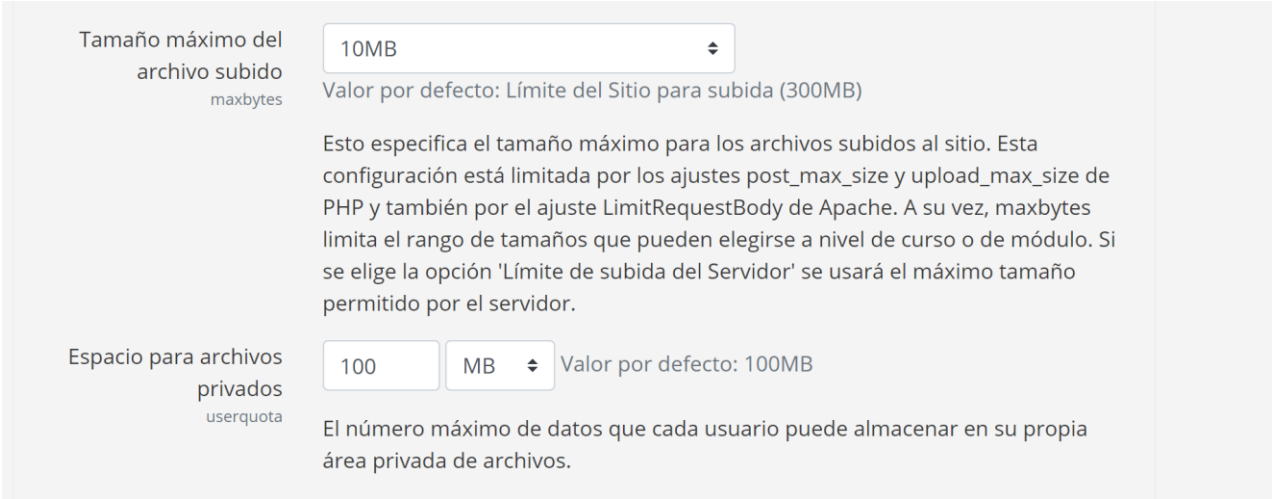


Fig. 46 Sección de tamaño de archivos

4.1.10. Servicio de Correos electrónicos

Moodle permite tener una comunicación con los usuarios mediante email, pero es necesario configurar dicho servicio en el servidor o por medio de un tercero, si el tamaño de los usuarios que usaran la plataforma es menor a 100 es posible usar un servicio gratuito como es el caso de Gmail donde podemos integrar el SMTP por medio de una llave y así no tener que preocuparnos de la seguridad o envío de correos por parte de la plataforma.

Para esto se pueden seguir los pasos en la guía de Moodle
https://docs.moodle.org/39/en/Email_setup_gmail

Es altamente recomendado tener un servicio de email de salida configurado con Moodle ya que este no solo permite la interacción con los usuarios sin necesidad de ingresar a la plataforma, de igual manera es utilizado para recuperar contraseñas entre otras cosas.

Para configurar el servicio SMTP en Moodle, se puede hacer desde la pestaña de *Administración del sitio* en la subpestaña de Servicio → Email → Configuración de correo de salida*:*

Fig. 47 Configuración de servicios

En esta sección configuramos todos los datos requeridos para la salida de correos electrónicos por parte de la aplicación, dependiendo de si se decide usar el servidor para enviar correos o un tercero como Gmail, es necesario llenar los datos que se piden y presionar el botón de guardado al final de la página. Si todo se configuró de forma correcta después de unos minutos el sitio web podrá enviar emails sin ningún problema:

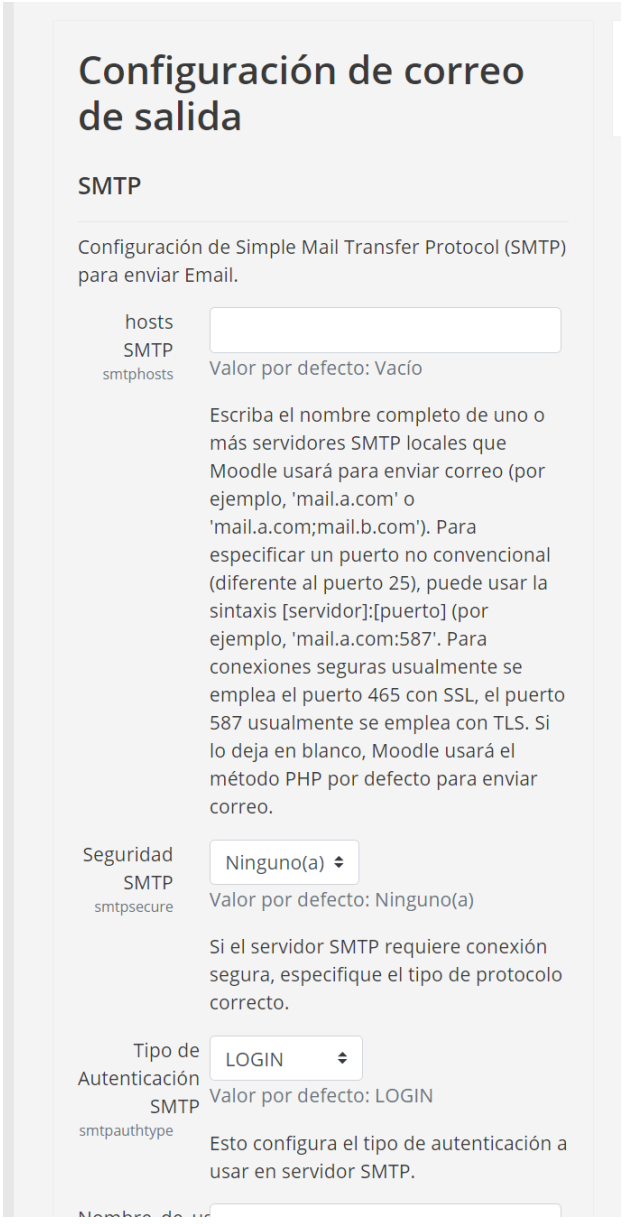


Fig. 48 Página de datos para la configuración de correos electronicos

4.2. Manual de usuario

El usuario final de la plataforma podrá ser un docente o un alumno, tendrán acceso al curso y a funciones básicas de la plataforma. La siguiente sección contiene ejemplificaciones de material final donde se muestra lo necesario para utilizar la plataforma como usuario final.

4.2.1. Inicio de sesión

Para acceder a la aplicación de Moodle se debe ingresar usando la cuenta creada por un administrador, dependiendo de los permisos que le fueron asignados será el tipo de perfil que se usará en la plataforma. Los tipos de usuarios que se manejan son de tipo profesor/docente y estudiante.

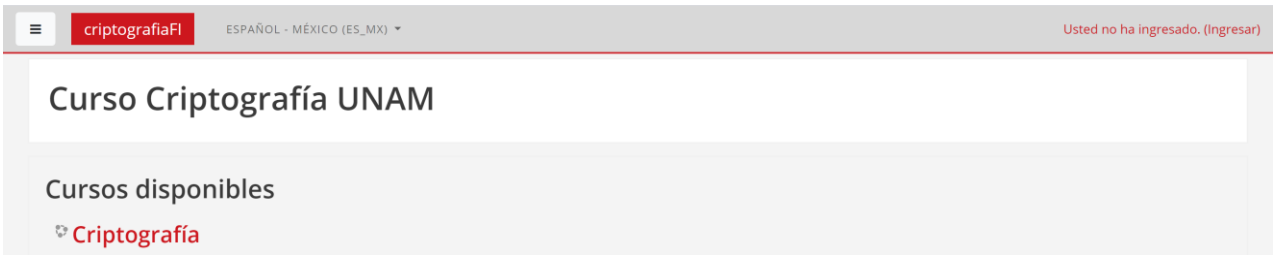


Fig. 49 Pantalla principal del curso, con botón de ingreso a la plataforma en la parte superior derecha de la pantalla

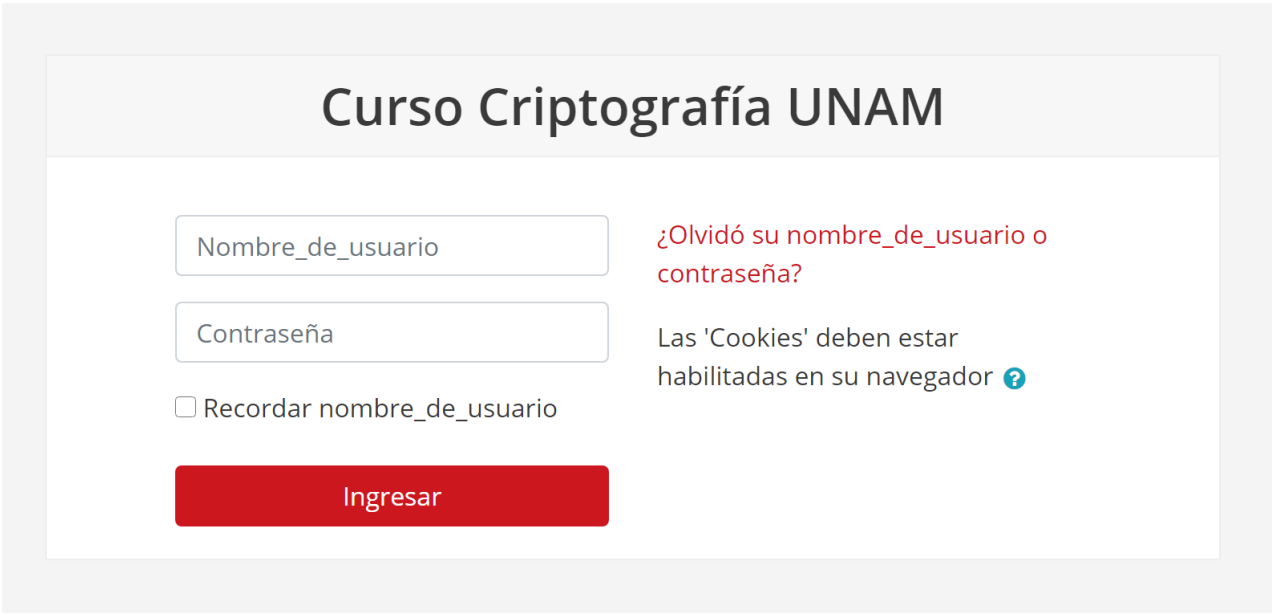


Fig. 50 Pantalla de inicio de sesión

Los usuarios de tipo estudiante tendrán acceso a la plataforma para cursar la materia; esto incluye ver el contenido, realizar los ejercicios, los exámenes y recibir calificaciones. Los usuarios de tipo profesor tendrán acceso a los cursos que les fueron asignados, con el mismo acceso de los usuarios estudiantes, más la capacidad de editar los cursos y calificar actividades realizadas por los estudiantes.

4.2.2. Perfil de usuario

Los usuarios de tipo profesor y estudiante tendrán una visualización similar al ingresar a la plataforma, donde se mostrarán el curso o cursos a los que se encuentran asignados.

Dentro de la página principal, después de realizar el inicio de sesión, es posible dirigirse a los cursos, seleccionando alguno en la barra lateral izquierda o dentro del bloque de "Vista general del curso".

La apariencia de la página puede ser modificada de manera personal por cada usuario presionando el botón de "Personalizar esta página".

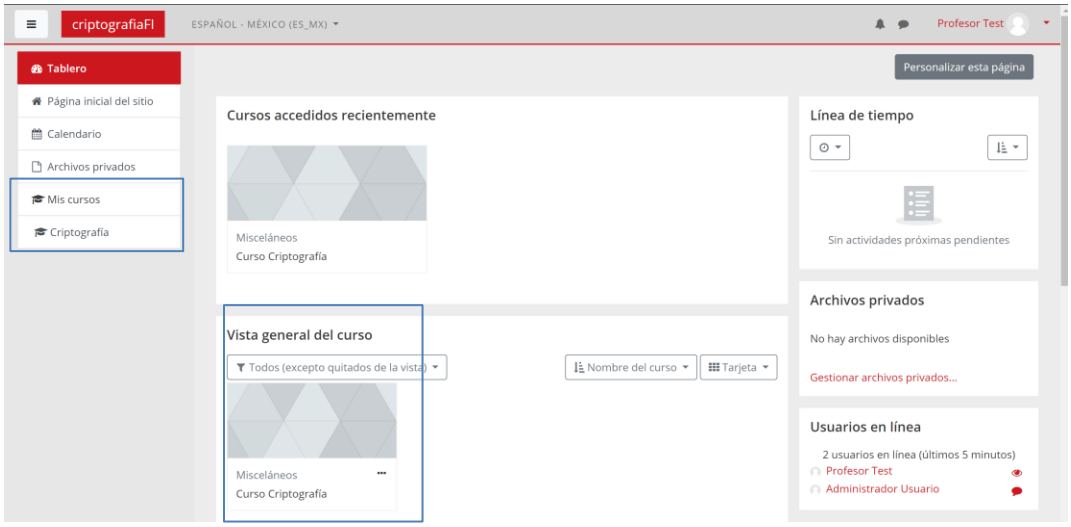


Fig. 51 Vista de página principal al realizar un inicio de sesión

4.2.3. Ingresar al curso como estudiante

Una vez dentro de la página del curso se presentarán las actividades que se pueden realizar para los estudiantes, dichas actividades pueden ser lecturas, ejercicios, exámenes, etc. Dependiendo de la configuración del curso pueden mostrarse todas las actividades al mismo tiempo o pueden estar ocultas e ir mostrándose dependiendo del progreso del estudiante.

Fig. 52 Página principal de curso para usuarios estudiantes

Actividad teórica

Las actividades teóricas son aquellas que se consideran completadas al ingresar a ellas, se recomienda que los estudiantes tomen apuntes como parte del proceso de aprendizaje. De igual manera, se puede requerir que el estudiante complete actividades didácticas dentro de ellas con el fin de poder avanzar en el material del curso.

Fig. 53 Pantalla de ejemplo de actividad teórica

Actividad de entrega

Existen actividades que requieren una entrega, como puede ser un documento escrito; los cuales no serán evaluados de forma automática, estos archivos serán revisados y evaluados por el profesor asignado. Las actividades de entrega permiten al estudiante subir archivos a la plataforma.

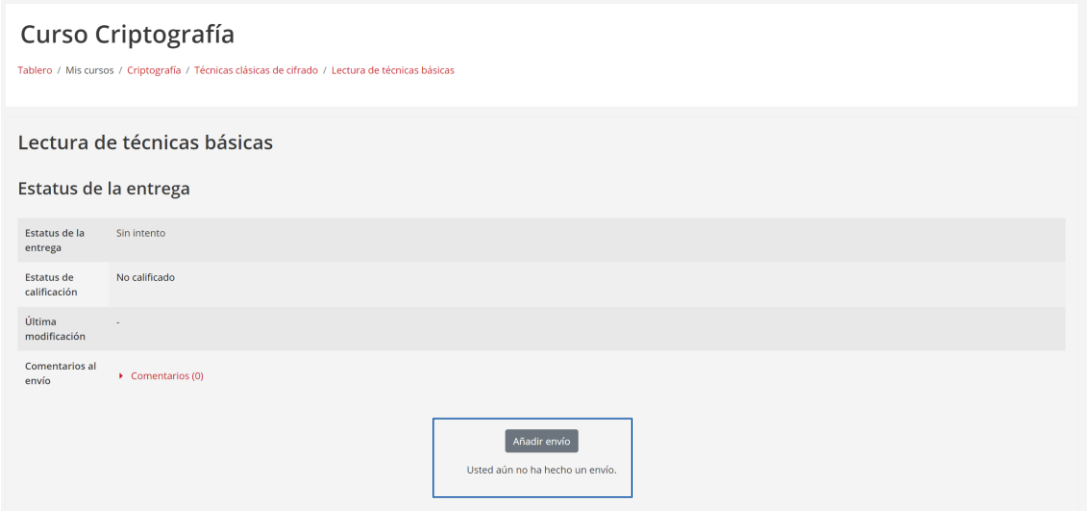


Fig. 54 Página principal de actividad de entrega

Al presionar "Añadir envío" se presentará una nueva página, donde se podrá seleccionar un archivo del ordenador para subir a la plataforma, arrastrando el archivo o seleccionándolo desde el explorador de archivos.

Dependiendo en las restricciones de la actividad se podrá subir archivos de cierto tamaño y cierto tipo.

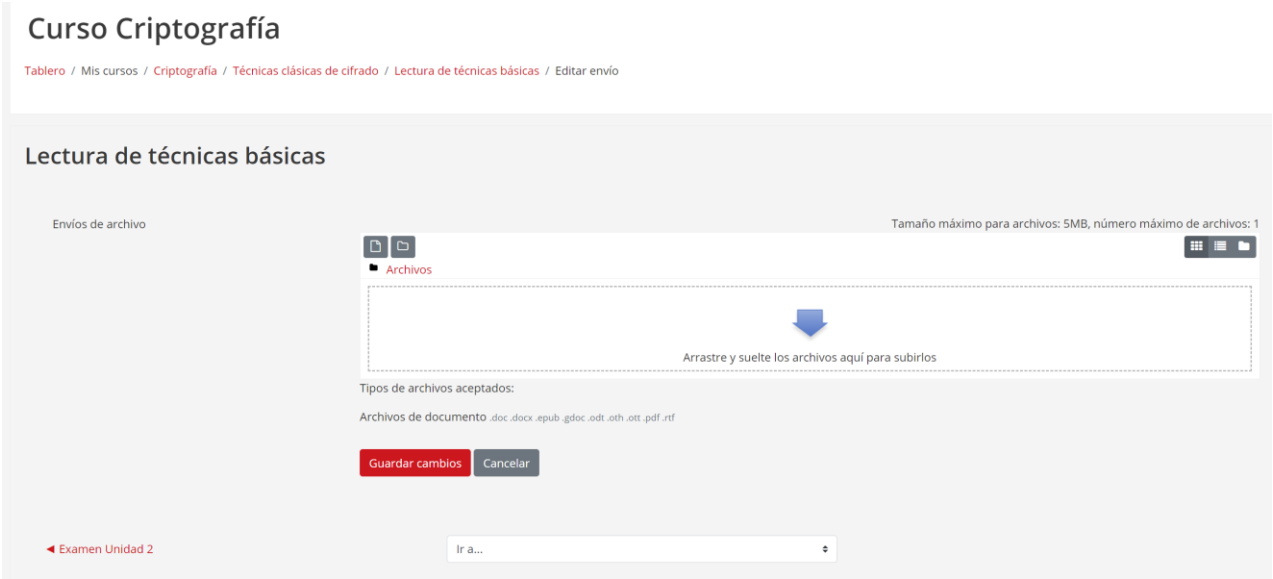


Fig. 55 Pantalla de envío de archivos

Cuando el archivo haya sido seleccionado, se subirá a la plataforma y se mostrará una pre-visualización. Es necesario seleccionar "Guardar cambios" para que sea marcado como enviado.

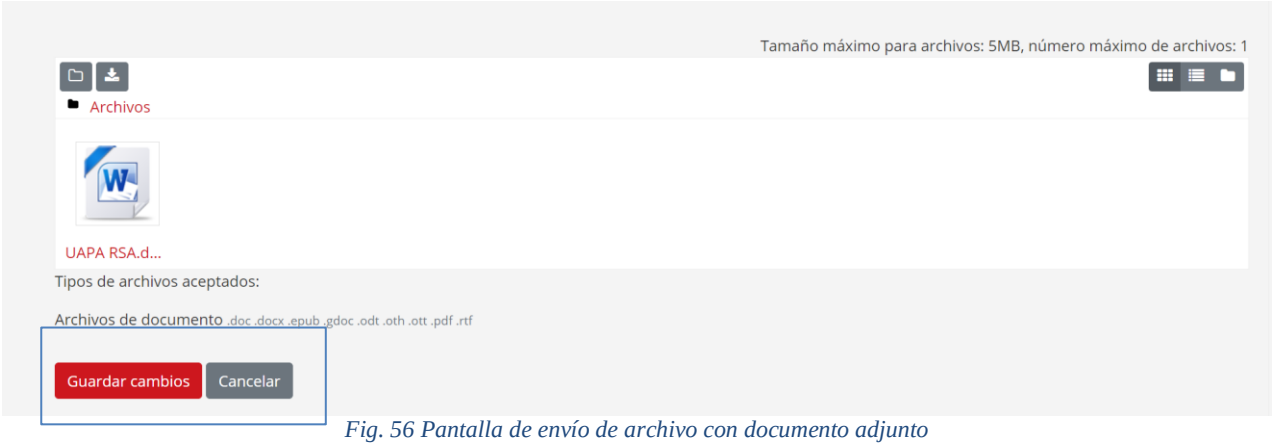


Fig. 56 Pantalla de envío de archivo con documento adjunto

Al guardar los cambios se regresará a la actividad, donde se mostrará el archivo que se ha enviado y será posible descargarlo, editar o eliminar el envío.

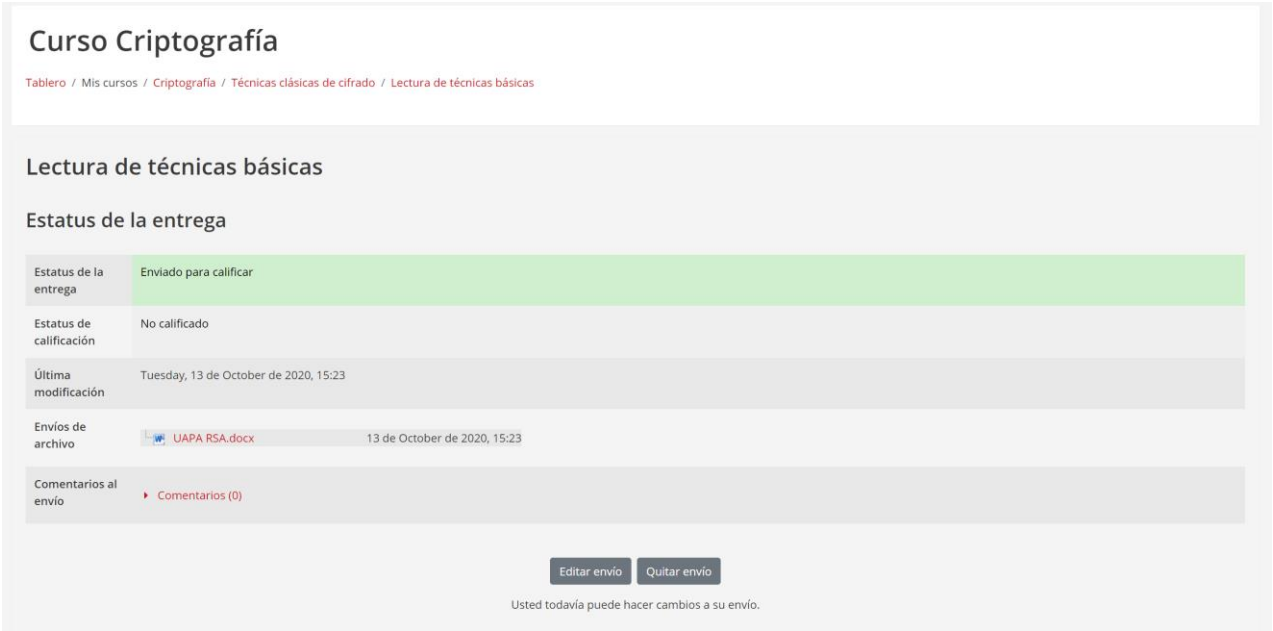


Fig. 57 Pantalla con archivo enviado

Actividades didácticas

Para reforzar los conocimientos adquiridos existen actividades didácticas, las cuales pueden constar de pequeños juegos o ejercicios para poner en práctica las habilidades adquiridas. Dichas actividades tendrán una calificación que será parte de la evaluación total del curso.

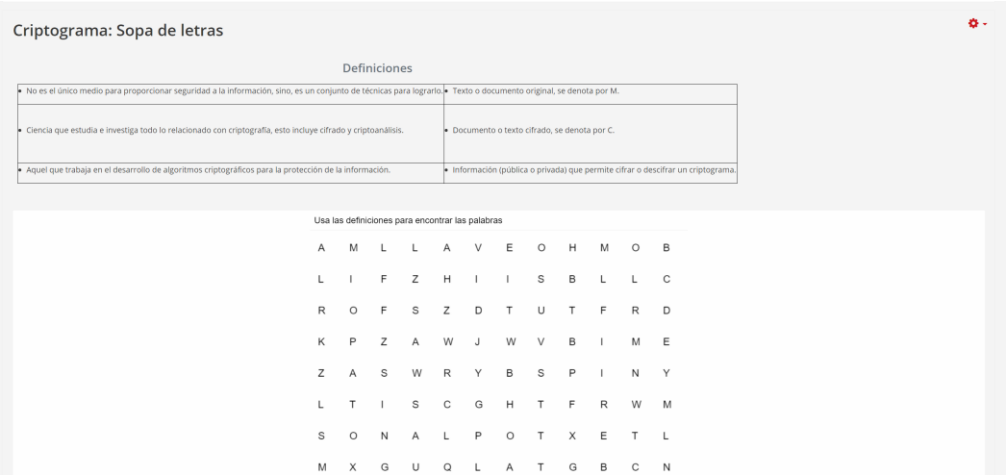


Fig. 58 Pantalla con ejemplo de actividad didáctica

Actividad de evaluación/examen

Las *actividades de evaluación* son aquellas usadas para validar que conocimiento aprendido durante el curso. Éstas constan de varias subactividades, las cuales tendrán un puntaje especifico y se mostrará a un costado de ellas.

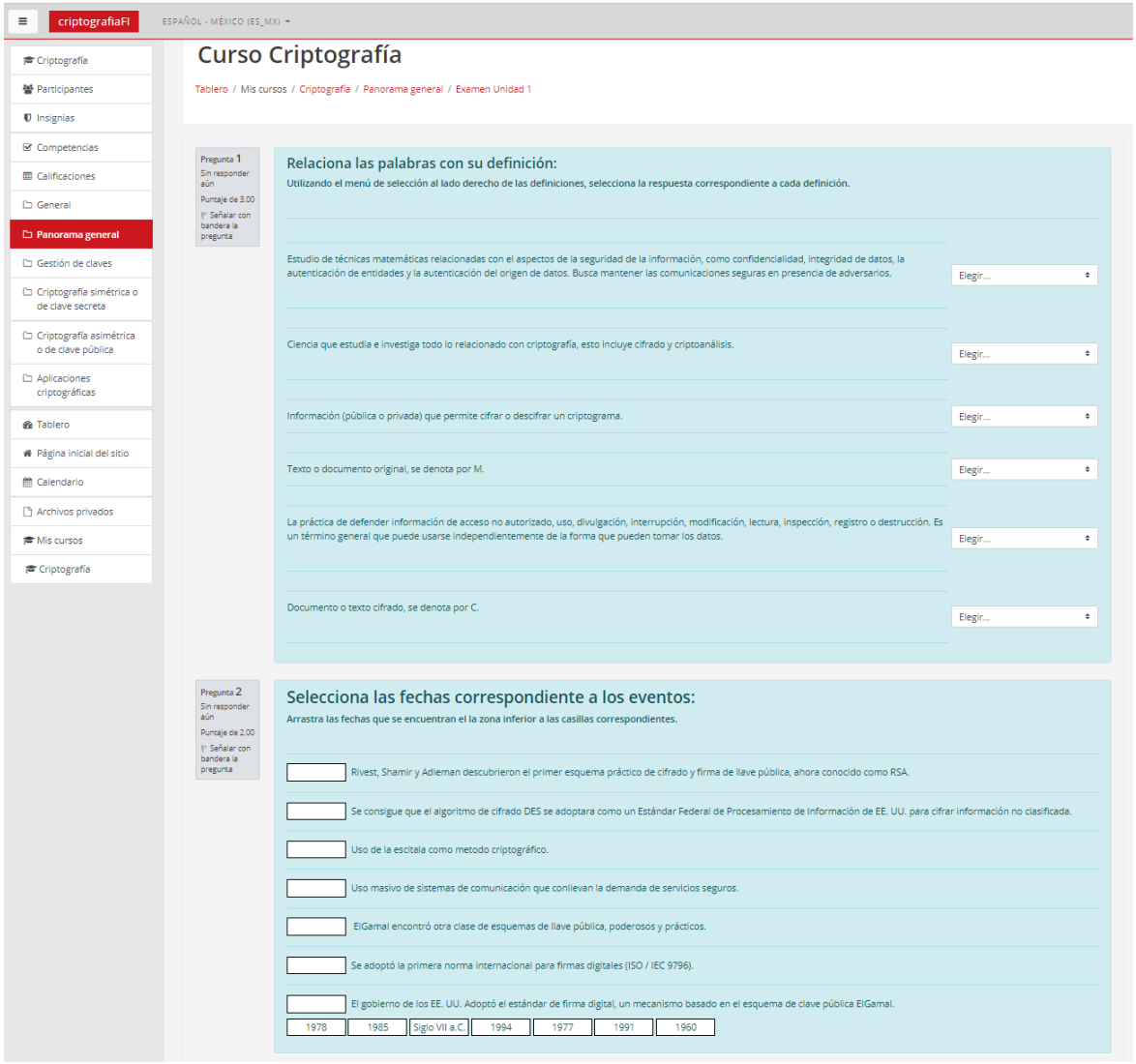


Fig. 59 Pantalla ejemplo con actividad de evaluación

Al terminar una evaluación se darán las opciones de regresar a modificar las respuestas o enviar las respuestas y terminar.

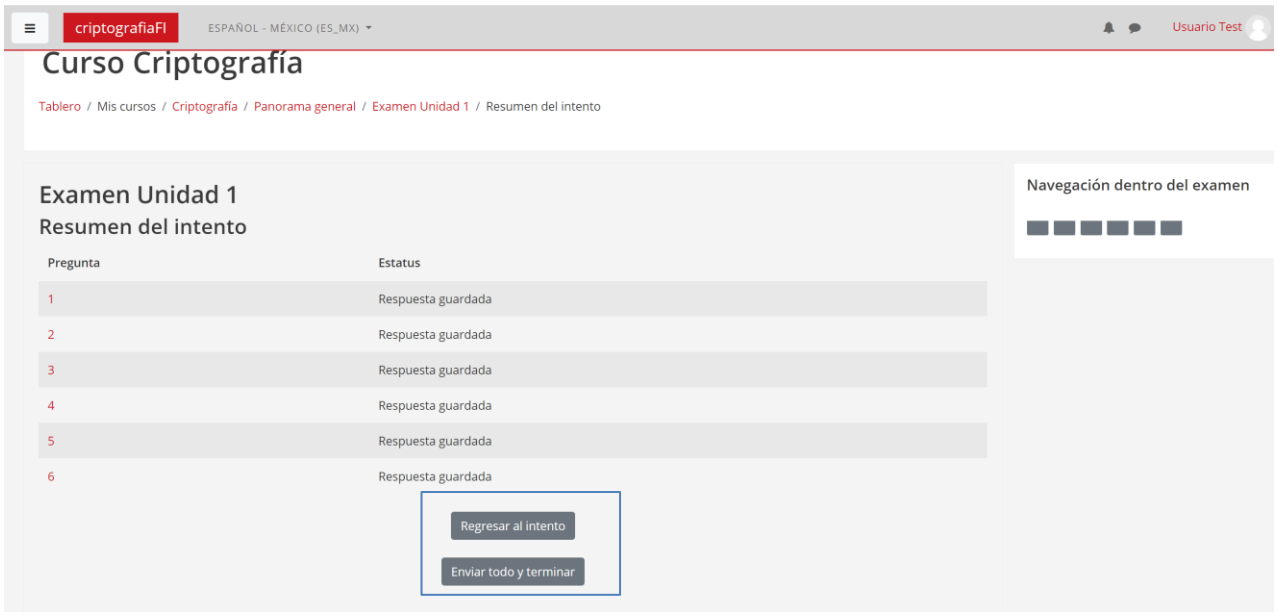


Fig. 60 Pantalla de envío de evaluación

Terminando la evaluación y, si se habilita la opción, se mostrará la calificación al instante y junto con ellas una retroalimentación de lo contestado, dependiendo de si fueron o no correctas las respuestas.

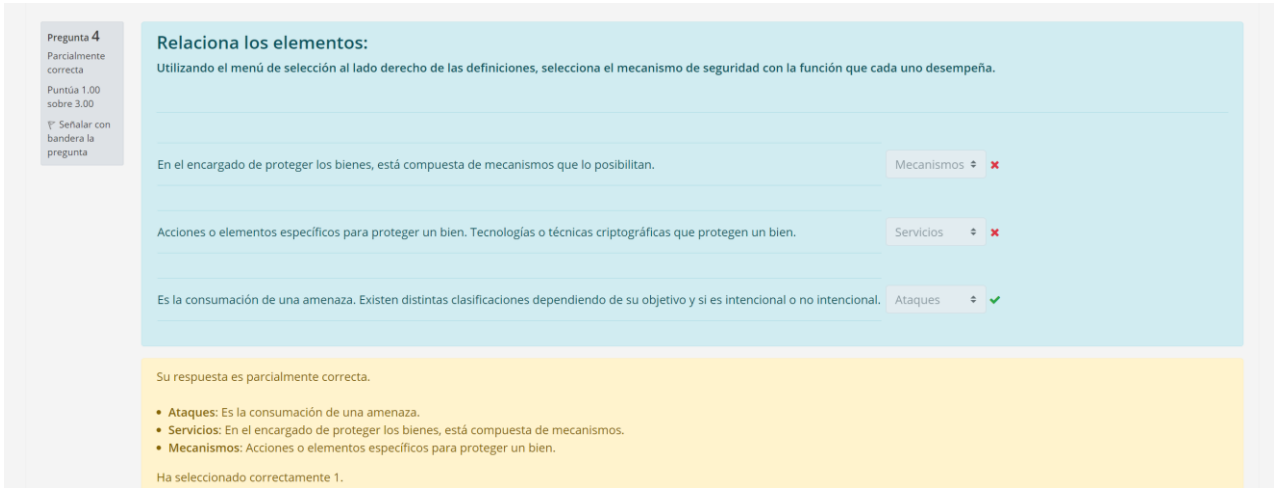


Fig. 61 Pantalla de evaluación con retroalimentación

Al volver a ingresar a la actividad de evaluación se mostrarán los intentos que se realizaron (si la actividad permite varios intentos) y la calificación de la actividad.

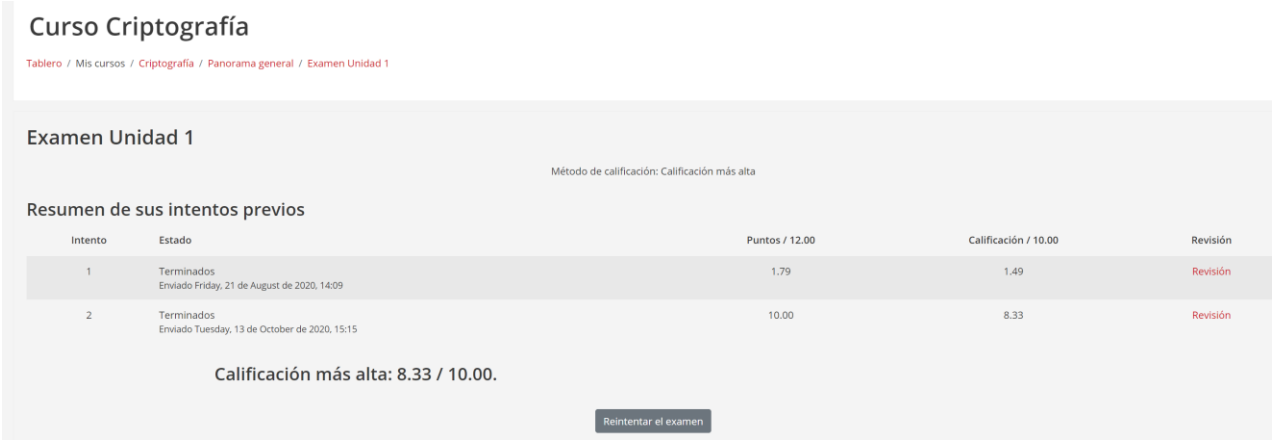


Fig. 62 Pantalla de calificaciones de la actividad de evaluación

Avances en el curso

El curso puede contar con actividades ocultas que serán mostradas una vez se cumplan ciertos requisitos; un ejemplo puede ser finalizar una evaluación, que mostrará el siguiente tema del curso una vez aprobado.

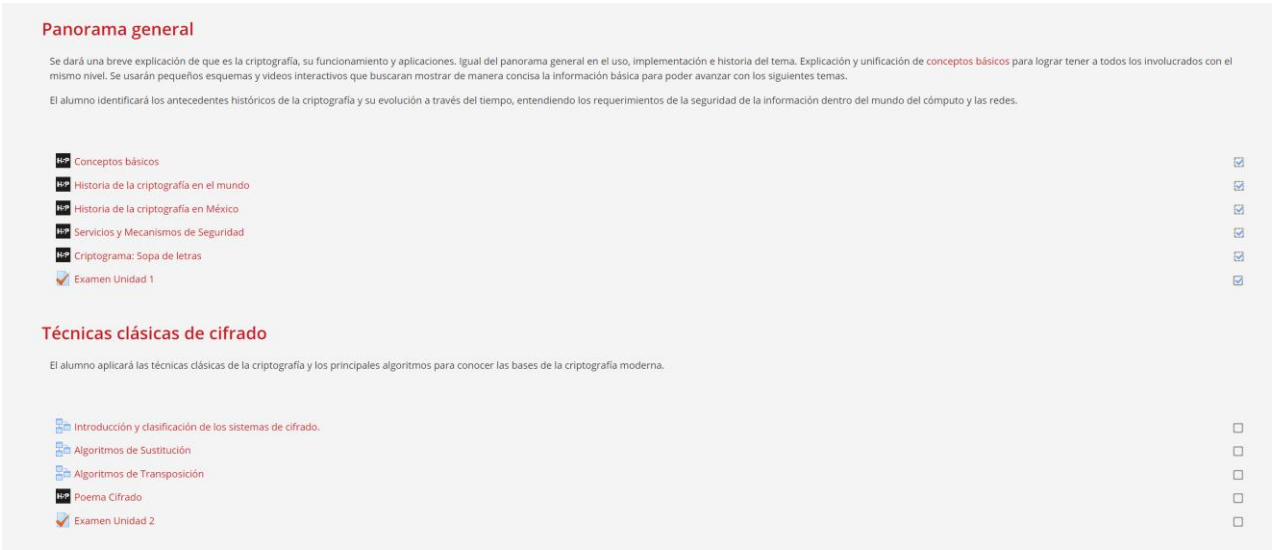


Fig. 63 Pantalla de avance de curso

Para poder visualizar todas las actividades y sus calificaciones se puede ingresar a la pestaña dentro del menú a la izquiera de "*Calificaciones*".

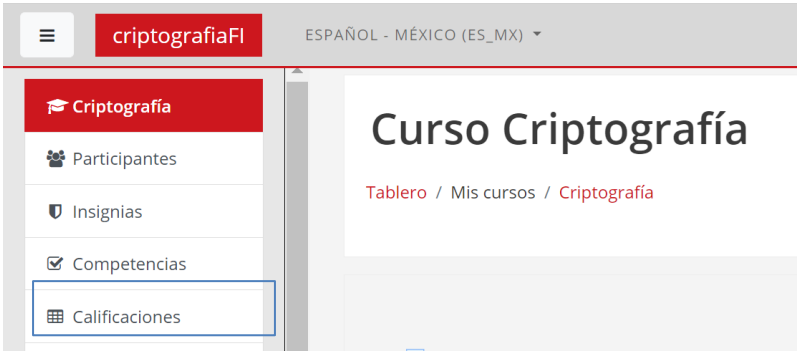


Fig. 64 Menú de navegación dentro del curso

En la página de calificaciones se podrá ver un desglose de las actividades, las calificaciones de cada una y el porcentaje que contribuye a la evaluación final.

Fig. 65 Página de calificaciones del alumno

4.2.4. Configuración de curso

Los usuarios de tipo profesor cuentan con la habilidad de modificar actividades y el curso. Se recomienda que cualquier modificación sea autorizada por el administrador y solo se haga si es necesario.

Para realizar una modificación se debe ingresar al curso y presionar el botón de "*activar edición*"

Fig. 66 Página principal del curso

Una vez activada la vista de edición se mostrarán diferentes iconos en la página que permitirán la modificación del curso.

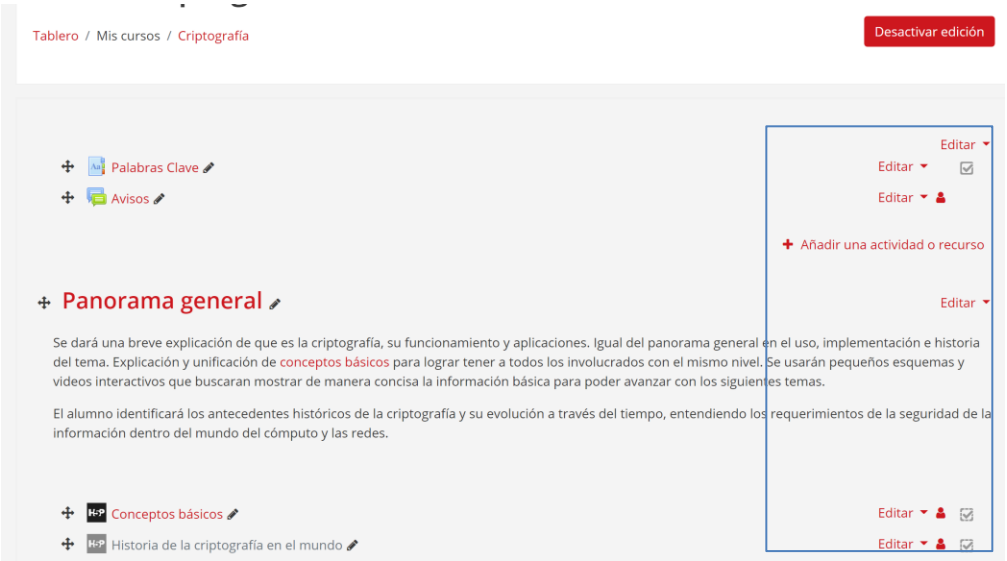


Fig. 67 Página principal del curso en modo edición

Para modificar una actividad es necesario presionar el menú de "*Editar*" que se encuentra junto a las actividades y dentro del menú de despliegue seleccionar "*Editar ajustes*"

Fig. 68 Menú de despliegue para editar actividad

El acceso a las actividades puede ser restringido de diferentes maneras, las más comunes son la restricción de acceso al botón "*finalización de actividad*", de igual manera se pueden agregar varias restricciones a una misma actividad.

Fig. 69 Opciones de restricción de acceso a actividades

Para mantener un flujo de evaluación del curso se puede marcar las actividades como "*finalizada*" dependiendo de ciertas restricciones.

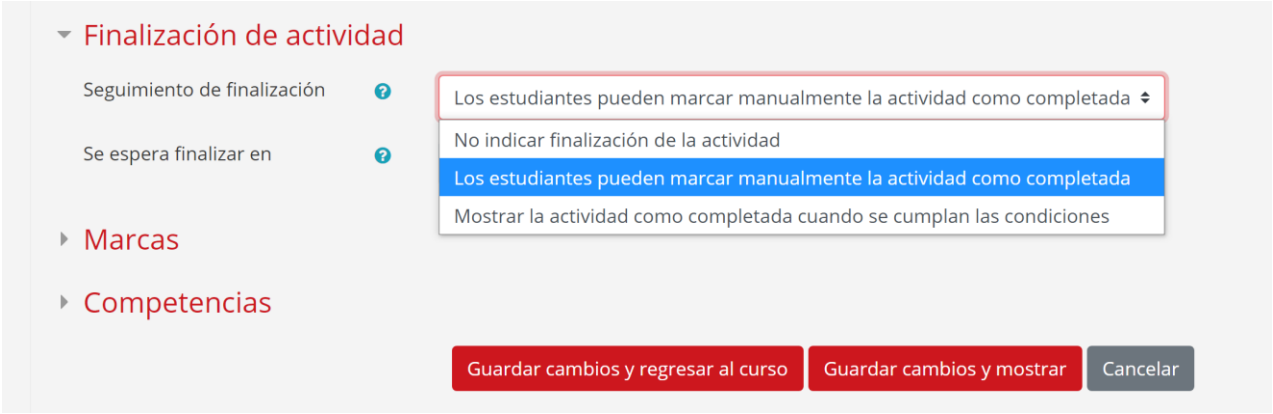


Fig. 70 Menú de restricciones para marcar una actividad como finalizada

4.2.5. Inscripción de curso

Para inscribir estudiantes al curso es necesario ingresar como profesor, una vez dentro del curso seleccionamos el icono de tuerca en el lado superior derecho y la opción "*Más*…"

Fig. 71 Pestaña de editar de curso

Dentro de "*administracón del curso*" ingresamos a la sección de "*Usuarios incritos*", donde podremos ver una lista de los usuarios que tienen acceso al curso, en dicha sección es posible modificar el acceso para los estudiantes.

Fig. 72 Menú de administración del curso para profesores

En la página de "*Usuarios inscritos*" seleccionamos "*Inscribir usuarios*", se desplegará un menú con un selector y búsqueda de usuarios dentro de la plataforma, al seleccionar a los usuarios se puede dar permiso de ser *estudiante* o *profesor sin permiso de edición*.

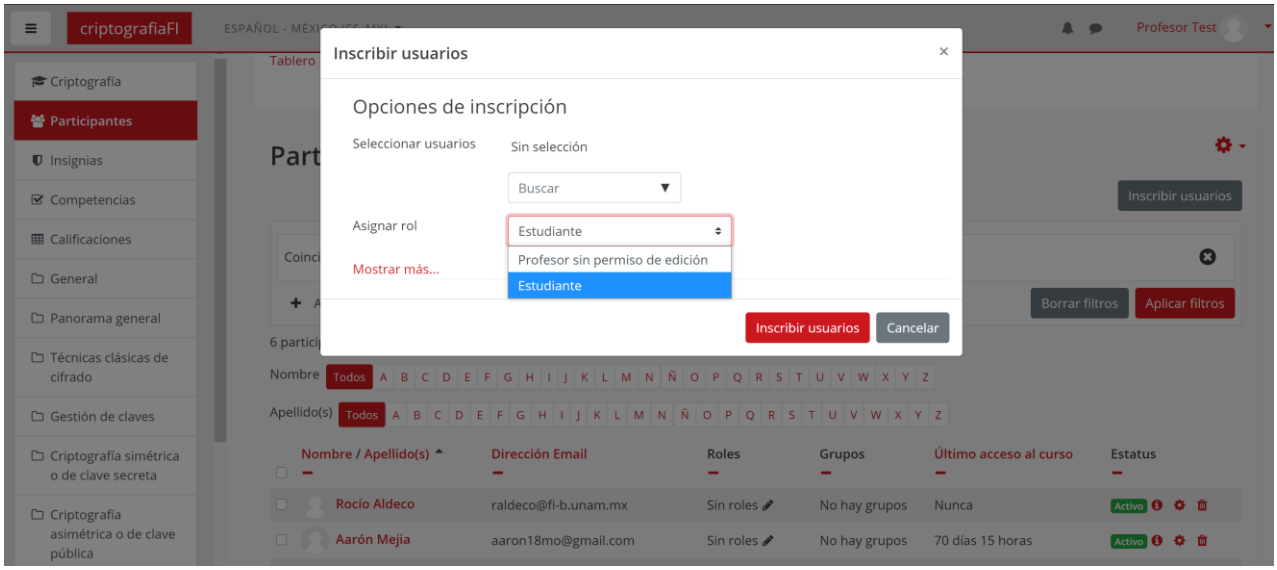


Fig. 73 Menú de inscripción al curso

4.2.6. Calificar

Revisar calificaciones de estudiantes

Los profesores del curso tendrán la capacidad de ver y modificar las calificaciones de los estudiantes, para esto se puede ingresar a la pestaña de "*calificaciones*" en el menú del curso.

En la página se mostrará un listado con los estudiantes inscritos y las calificaciones correspondientes a las actividades que se han realizado.

Fig. 74 Página de "reporte de calificaciones" para profesores

Es posible ingresar al perfil de cada estudiante y ver un desglose más detallado de las calificaciones, también es posible anular o excluir las calificaciones si así se requiere.

Fig. 75 Página de desglose de calificaciones de alumno

Modificar calificaciones

Dentro de la página de "*Reporte del calificador*" es posible ver las calificaciones de cada actividad, al presionar el icono de editar se dará la posibilidad de realizar una revisión a cualquier actividad y modificar la calificación si es necesario.

Fig. 76 Revisión de evaluación con avaluación automática

Al realizar una revisión es posible cambiar la calificación y dar un comentario de retroalimentación. Los cambios guardados se verán reflejados tanto para el estudiante como para el profesor.

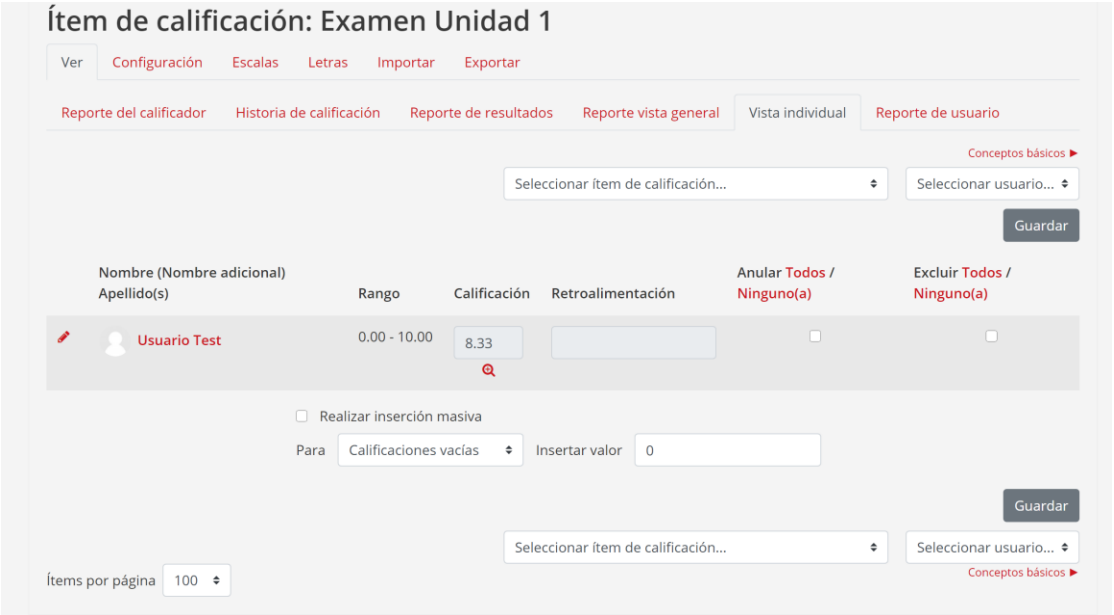


Fig. 77 Pantalla de modificación de calificación de actividades

Calificar actividades de entrega

Las actividades que requieren archivos entregables deben ser calificadas de forma manual, al ingresar a estas actividades se verá un botón con la leyenda "*ver todos los envíos*", al presionar se desplegará una lista de los archivos enviados y el estudiante que los envió.

Fig. 78 Pantalla de entrega de archivos para profesores

En la pantalla de "*Ver todos los envíos*" se muestra una lista de los alumnos y las entregas que se han realizado. Para poder hacer una revisión y calificar las actividades se debe seleccionar el botón de "*calificación*" al lado del nombre del estudiante.

Fig. 79 Pantalla de visualización de entregas de trabajos

La página de evaluación permite la descarga del archivo que el estudiante subió a la plataforma, habrá una sección para poner la calificación de la actividad y un recuadro donde se podrán escribir comentarios personalizados sobre la entrega realizada.

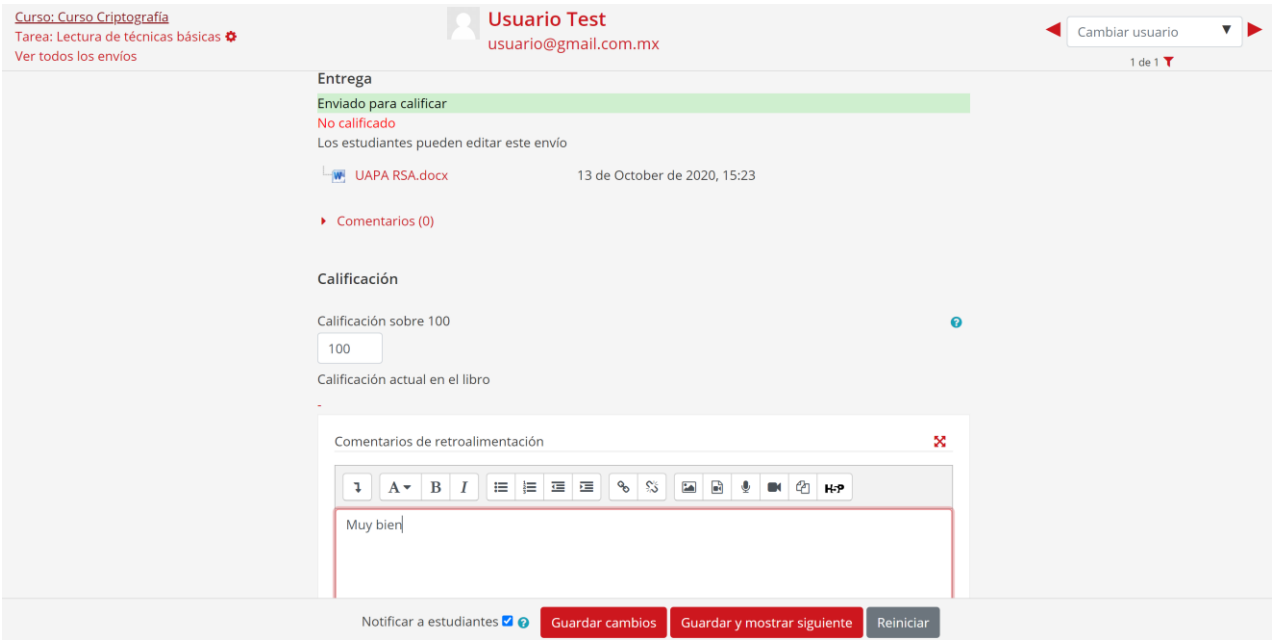


Fig. 80 Página de evaluación de entrega de archivos para los profesores

4.2.7. Monitoreo de progreso

Los alumnos podrán visualizar el avance de las actividades en el curso con ayuda de las casillas que se muestran junto a las actividades. A los alumnos se les puede permitir o no que las casillas sean marcadas manualmente; una vez que se haya cumplido con los criterios de una actividad se mostrará una palomita. Con ayuda de esto y según la configuración del curso, se puede mostrar al curso como finalizado.

Fig. 81 Sección con cajas de finalización

4.3. Ilustraciones y diagramas de apoyo

En el proceso de creación del curso se generaron diagramas que, de apoyo al contenido teórico, ya que la mayor parte de los conceptos no son de fácil comprensión de forma escrita para todos los usuarios, por tanto, se crearon diferentes elementos dependiendo de los temas de cada unidad en la materia.

A continuación se presentan estos diagramas organizados por unidad:

4.3.1. Unidad 1: Panorama general

La primera unidad de la asignatura hace referencia a los temas de la seguridad de la información, buscando dar a entender los requerimientos de la misma dentro del mundo de la computación y las redes. Es necesario comprender los diferentes tipos de ataques y amenazas que pueden ser hallados dentro de la Informática para así ver la importancia de la criptografía en el área de seguridad Informática. Al igual comprender la estructura de la organización de las tecnologías de la comunicación nos ayuda a saber donde se puede implementar la criptografía.

A continuación se muestran diagramas que representan dichos conceptos:

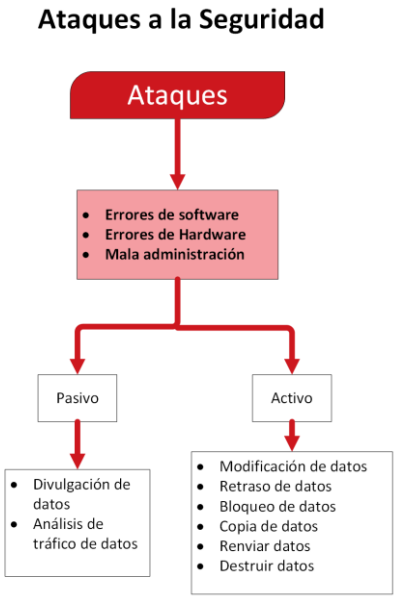


Fig. 82 Diagrama de los tipos de ataques en la seguridad Informática

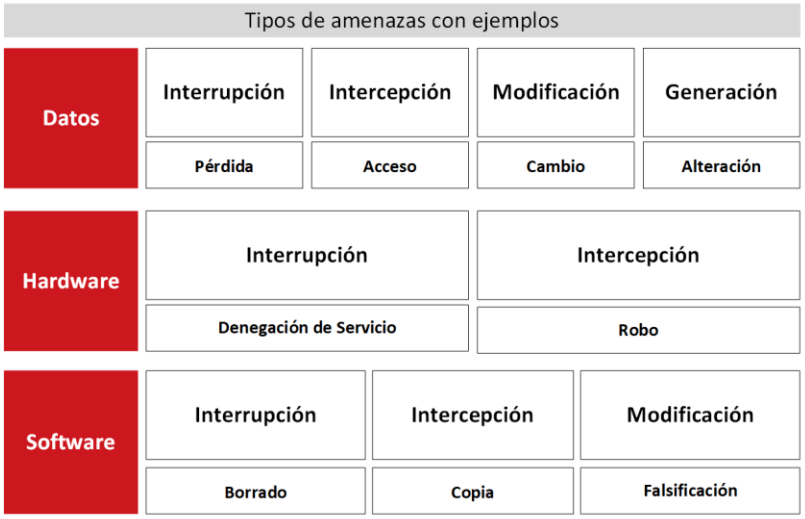


Fig. 83 Diagrama de muestra de tipos de amenazas y ejemplos en la seguridad Informática

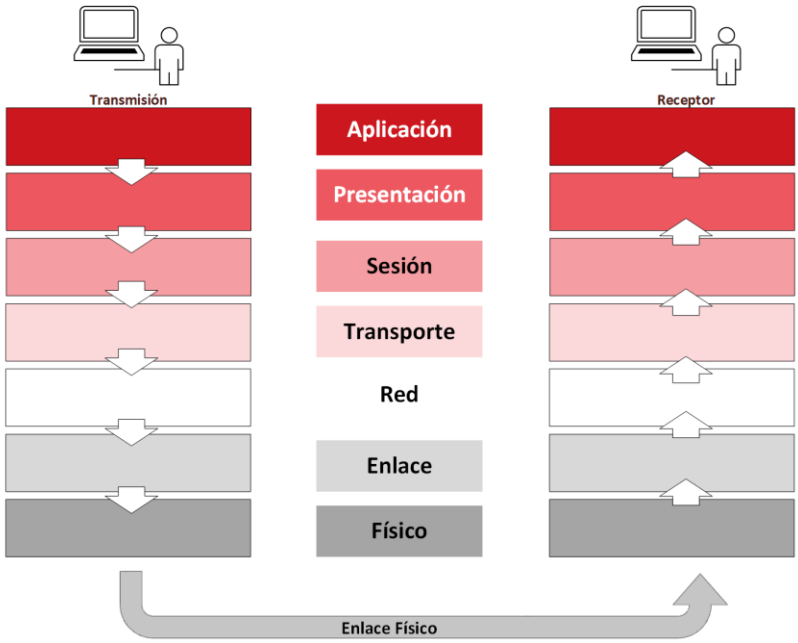


Fig. 84 Diagrama de comunicación entre transmisor y receptor con Modelo OSI

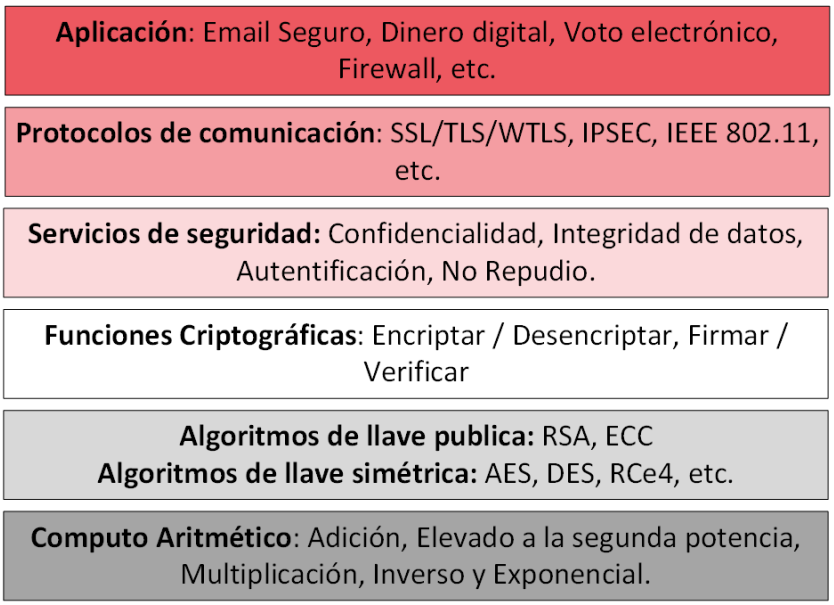


Fig. 85 Modelo OSI con ejemplos de cada nivel

4.3.2. Unidad 2: Técnicas clásicas de cifrado

La unidad 2 presenta los principios fundamentales para la criptografía, términos básicos de uso frecuente en la literatura, técnicas clásicas y principales algoritmos para conocer las bases de la criptografía moderna.

A continuación se muestran diagramas que representan los conceptos de las técnicas clásicas de cifrado:

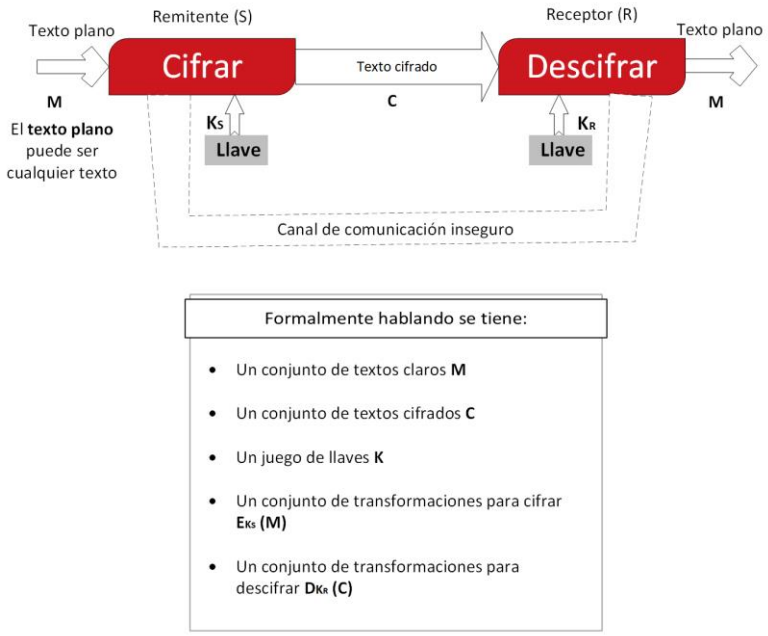


Fig. 86 Diagrama de única comunicación básica utilizando un cifrado y descifrado mediante un canal de comunicación inseguro.

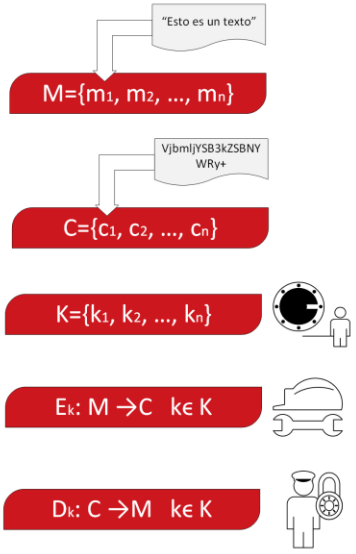


Fig. 87 Diagramas de explicación de los elementos que se envían en la comunicación segura por medio de un cifrado

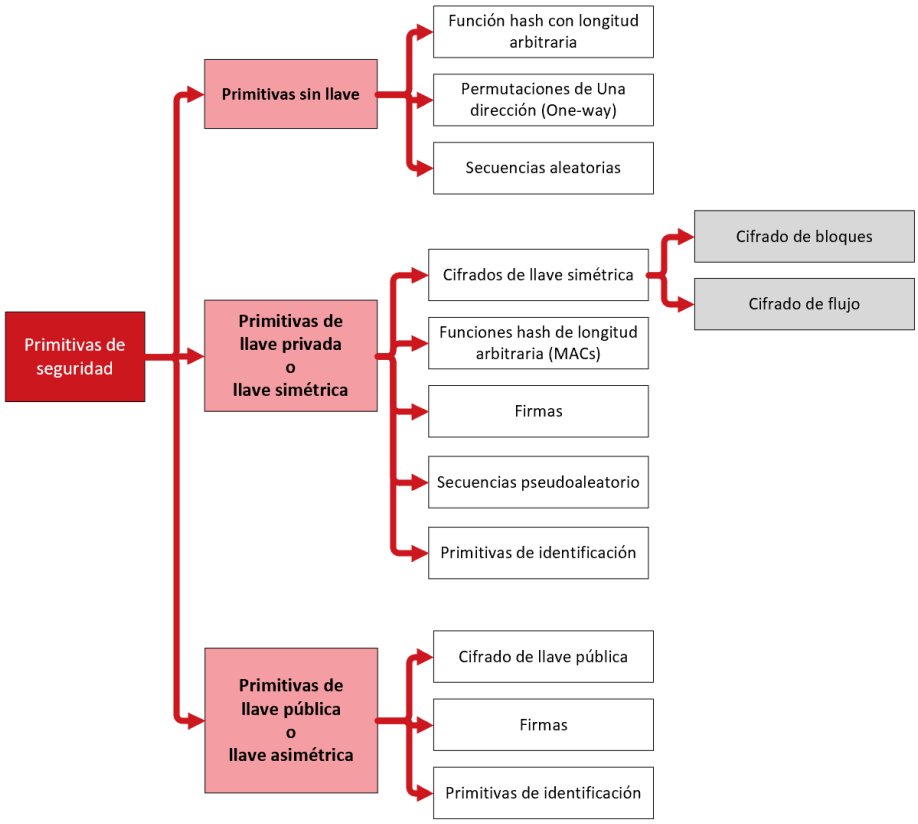


Fig. 88 Diagrama de tipos de cifrados y usos de llaves en los cifrados

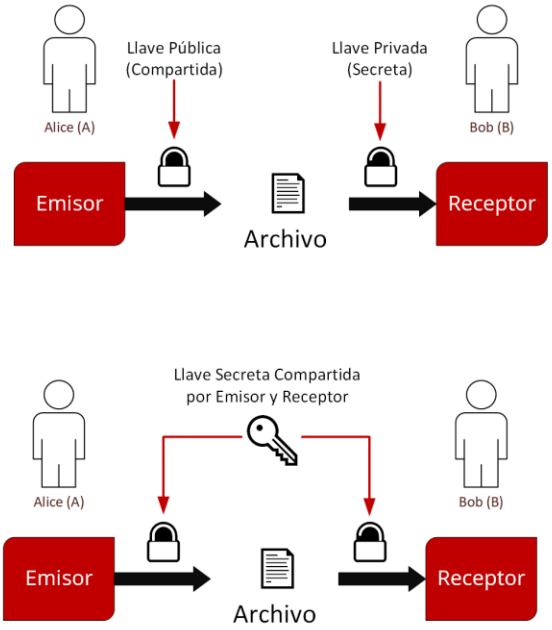


Fig. 89 distinción entre cifrado de llave pública (primera imagen) y cifrado de llave privada (segunda imagen)

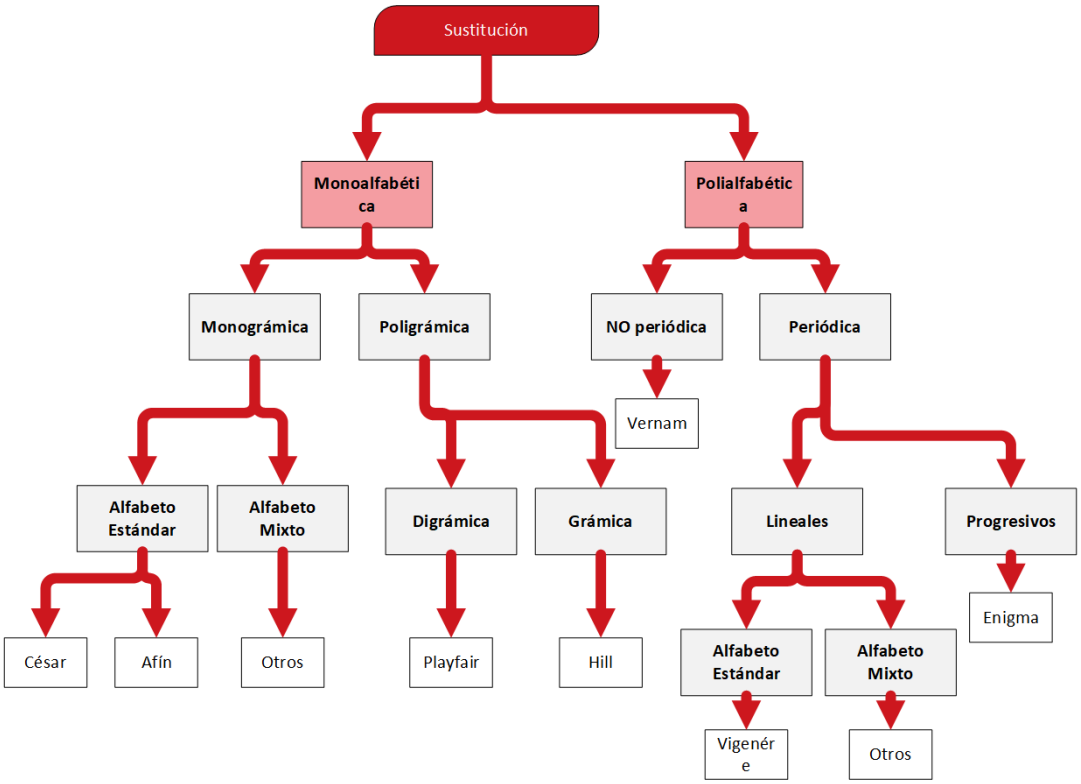


Fig. 90 Cifrados por sustitución, sus subtipos y ejemplos de cifrados que son usados

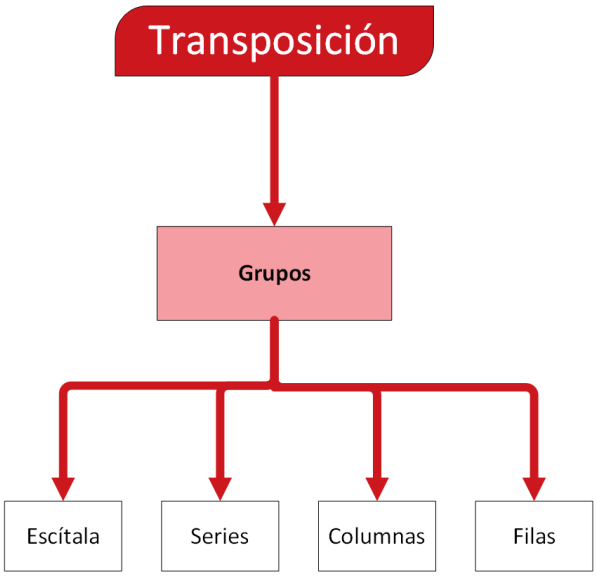


Fig. 91 Cifrados por transposición con ejemplos

4.3.3. Unidad 4: Criptografía simétrica o de clave secreta

La unidad 4 presenta los inicios de la criptografía moderna, así introduciendo los cifrados de simétricos, los cuales constan de una llave secreta que es compartida por los participantes de la comunicación.

En la unidad se comienzan a ver temas de difícil comprensión debido al nivel de baja abstracción que se puede ver en los algoritmos de cifrado, el caso de temas como generadores de secuencias y movimientos de registros donde se trabaja a nivel bit y byte en la computadora pueden llegar a ser de difícil comprensión y por lo tanto un material visual es de gran ayuda para los mismos.

A continuación se muestran diagramas que representan dichos conceptos:

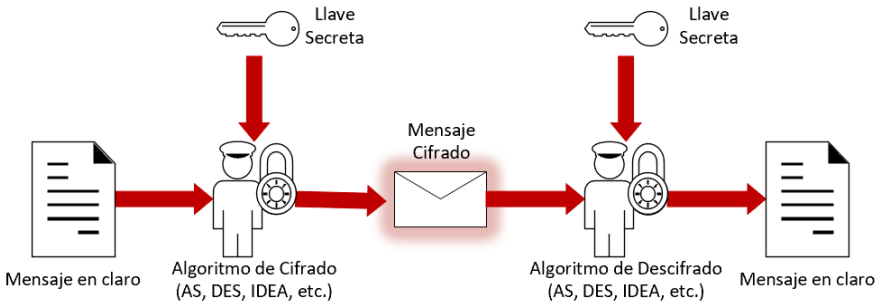


Fig. 92 Diagrama de comunicación con un cifrado de llave privada/llave secreta

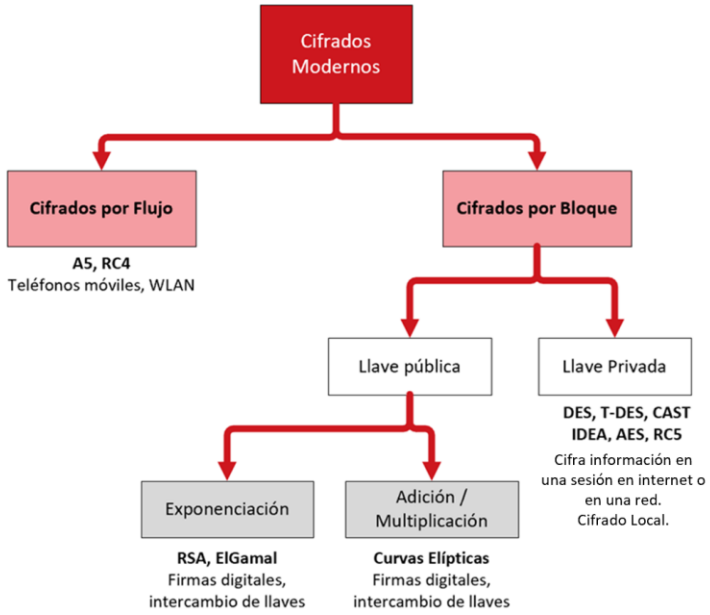


Fig. 93 Estructura de los cifrados modernos de flujo y de bloque con ejemplos

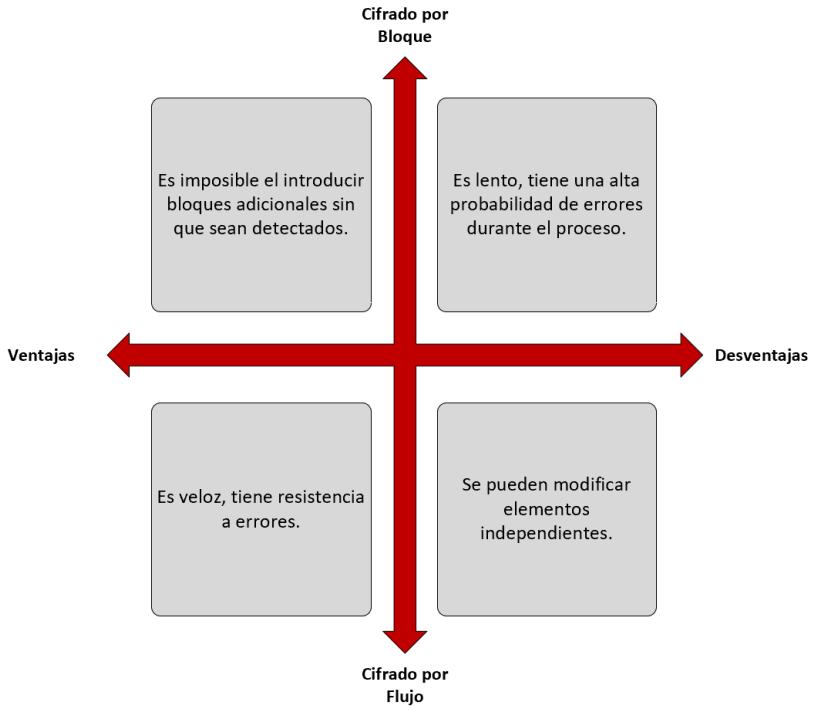


Fig. 94 Cuadro comparativo entre ventajas y desventajas en el uso de cifrados por flujo o por bloque

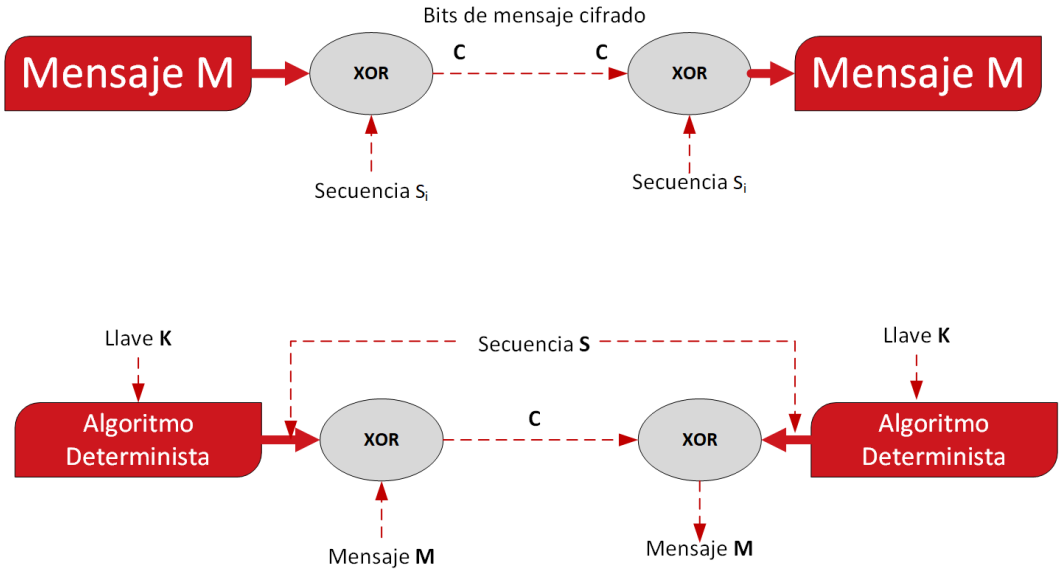


Fig. 95 Cifrado por flujo, versión general y versión detallada en el uso de llave privada

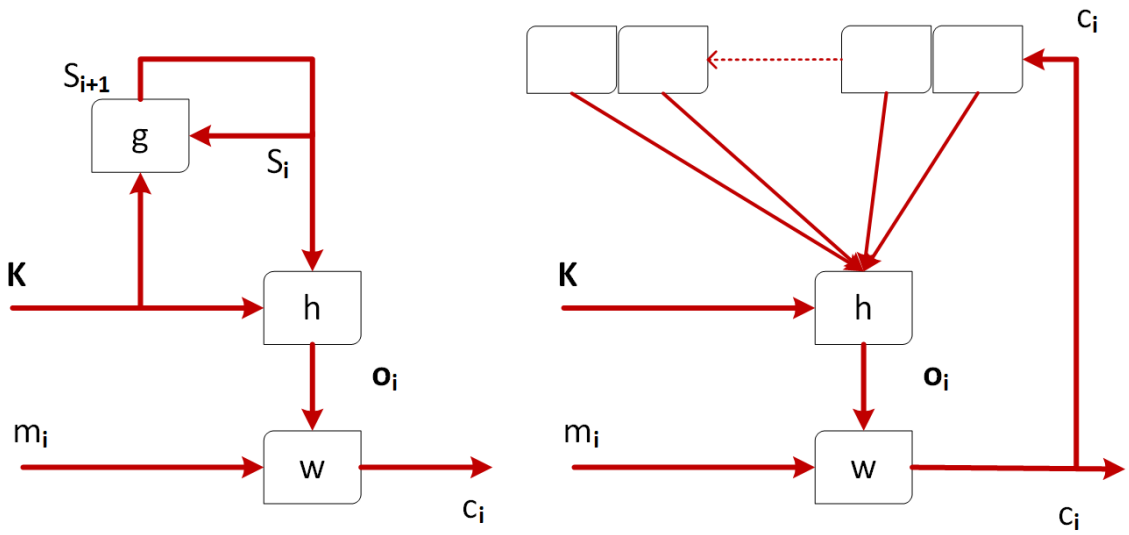


Fig. 96 Diagramas de tipos de generadores de secuencias, (izquierda) Generador síncrono (Derecha) Generador asíncrono

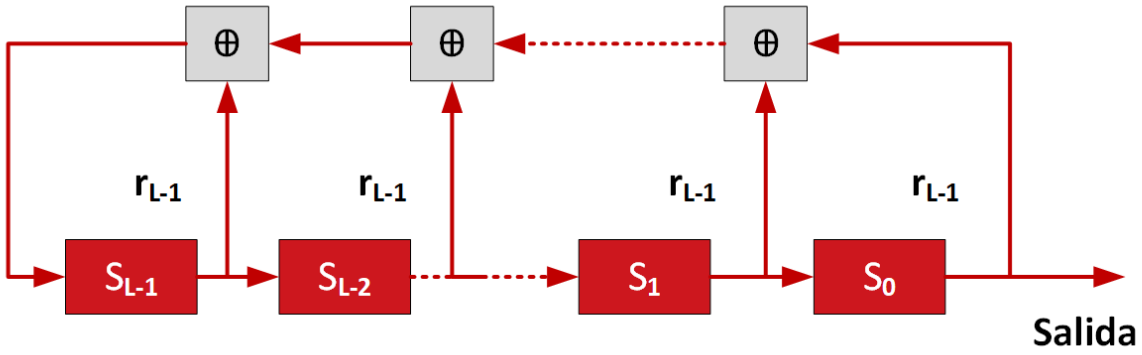


Fig. 97 Funcionamiento de Registros de Desplazamiento Retroalimentados Lineales

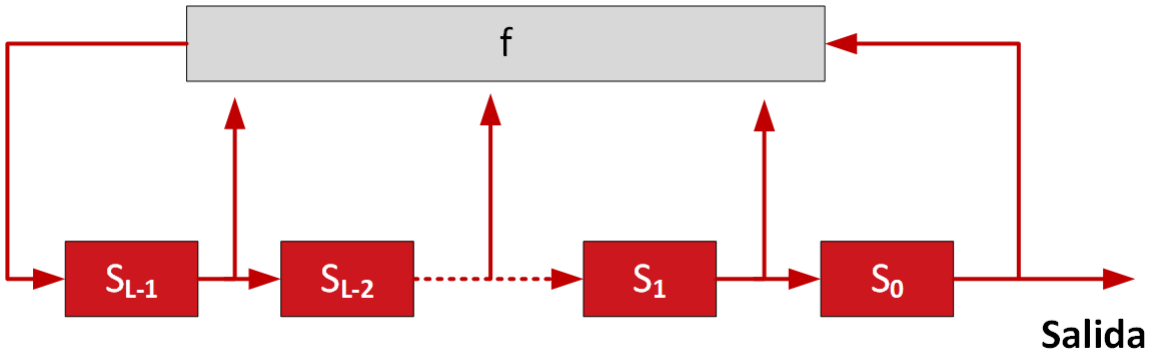


Fig. 98 Funcionamiento Registros de Desplazamiento Retroalimentados No Lineales

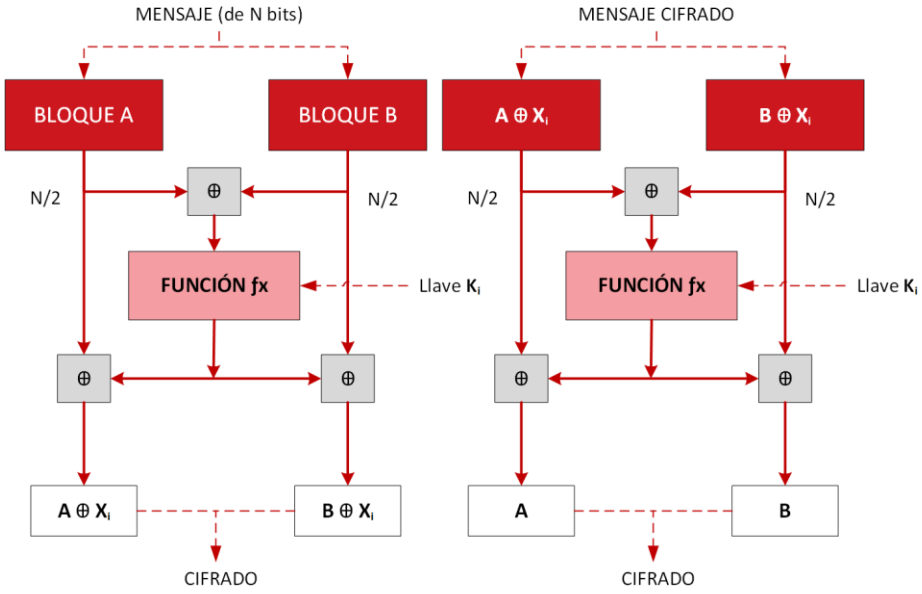


Fig. 99 Esquema de cifrado y descifrado por bloques

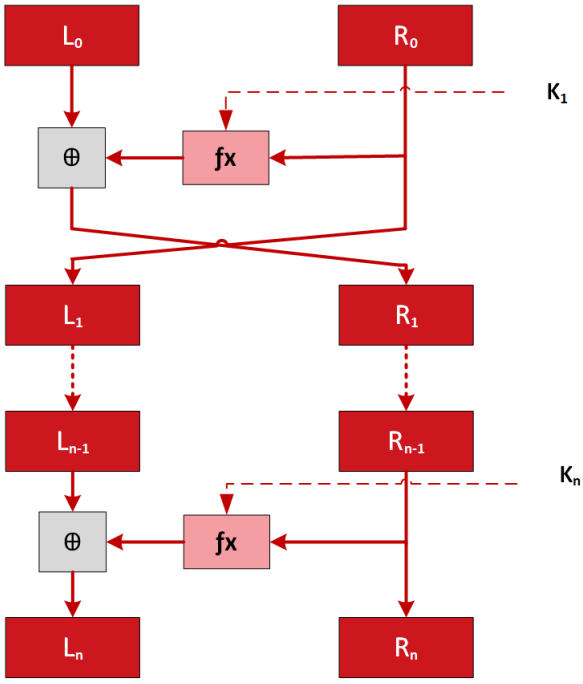


Fig. 100 Esquema de Redes de Feistel (Cifrado por Bloques)

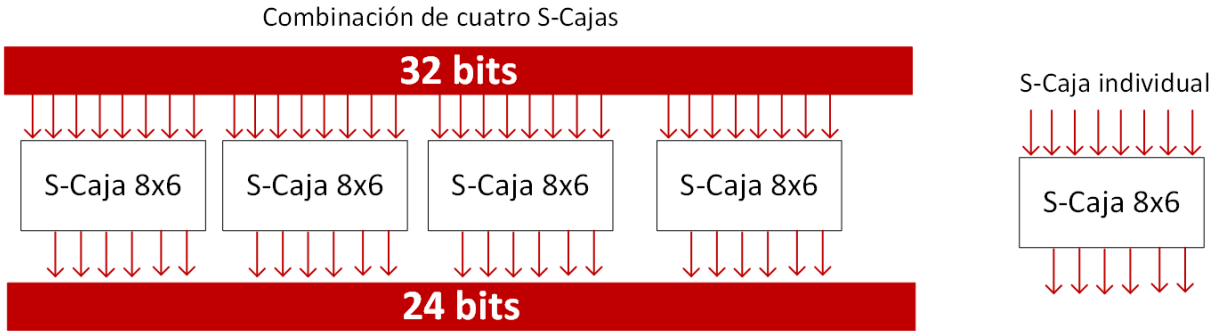


Fig. 101 Esquema de una S-Caja y una combinación de ellas, usadas principalmente en los cifrados por bloques

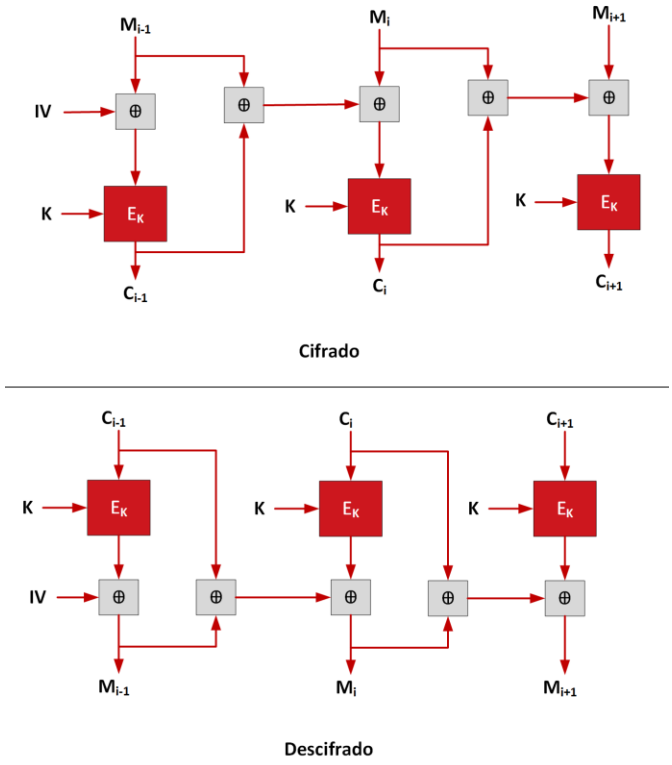


Fig. 102 Modo de Operación en cifrado por bloques tipo PCBC

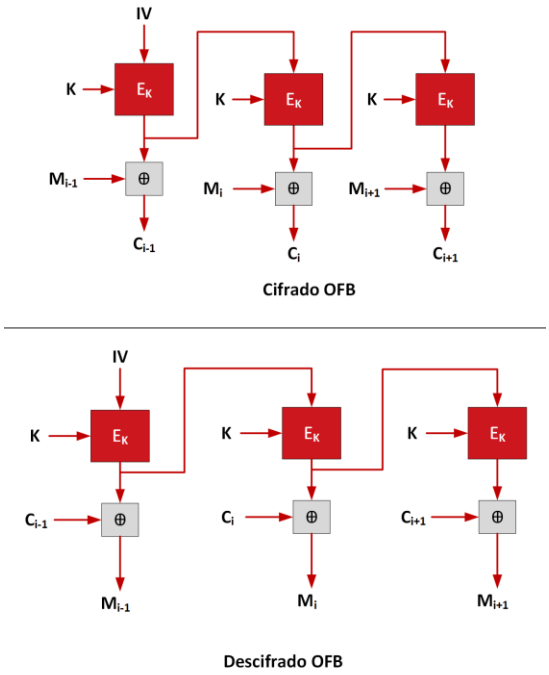


Fig. 103 Modo de Operación en cifrado por bloques tipo OFB

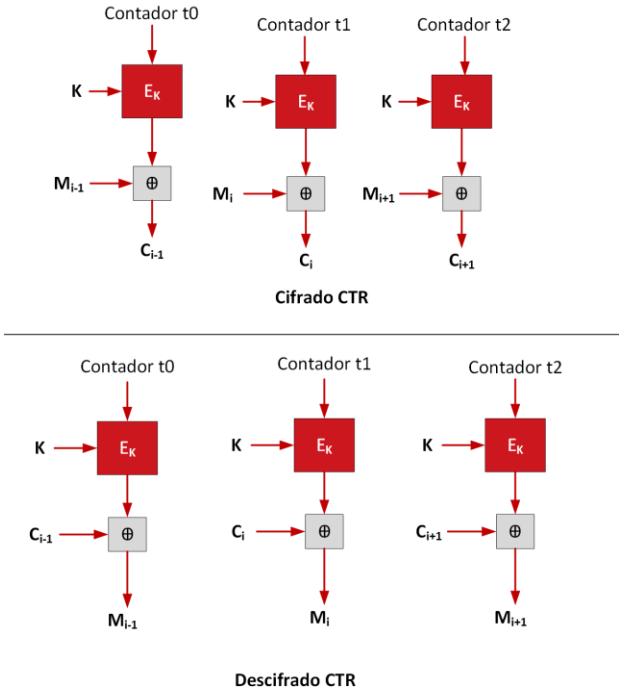


Fig. 104 Modo de Operación en cifrado por bloques tipo CTR

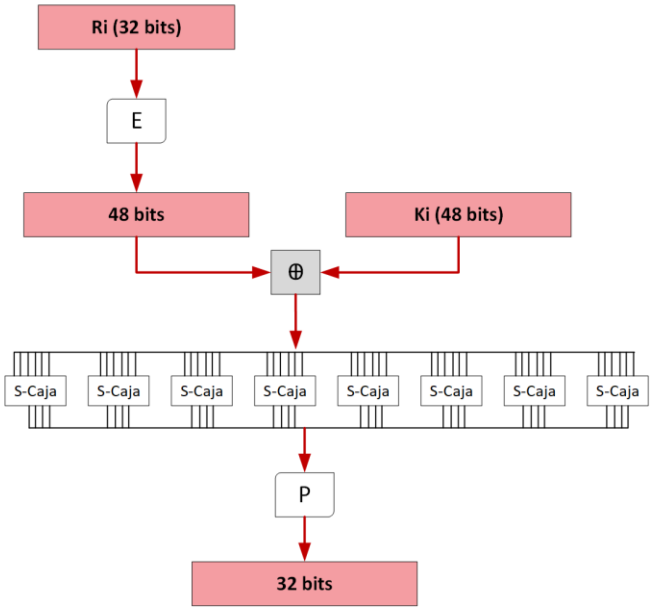


Fig. 105 Esquema de la función f del algoritmo DES

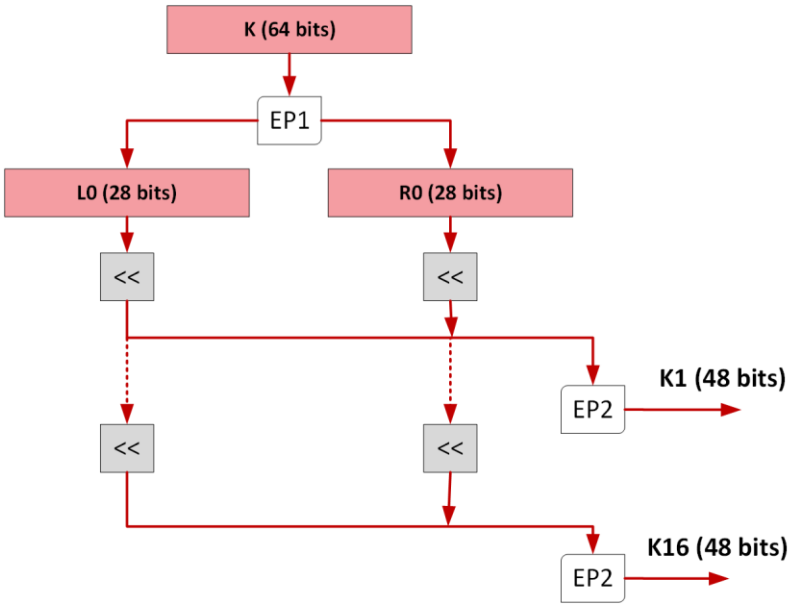


Fig. 106 Cálculo de las llaves (K i) para el algoritmo DES.

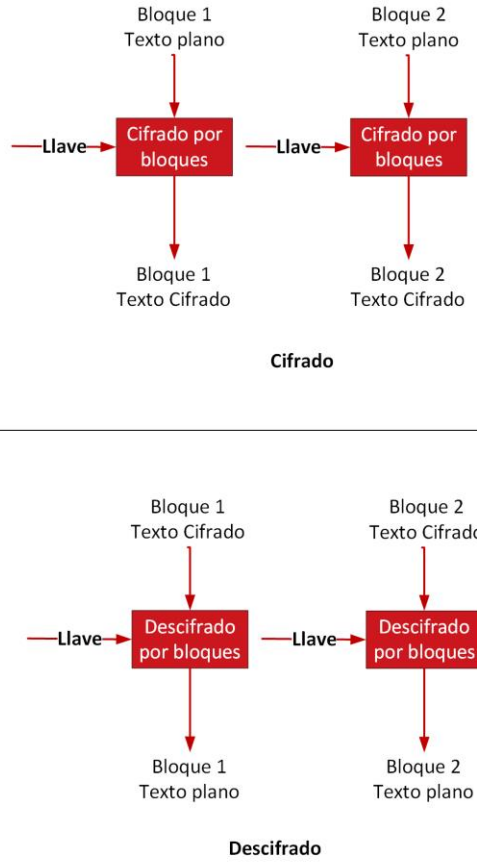


Fig. 107 Modo de Operación en cifrado por bloques tipo ECB

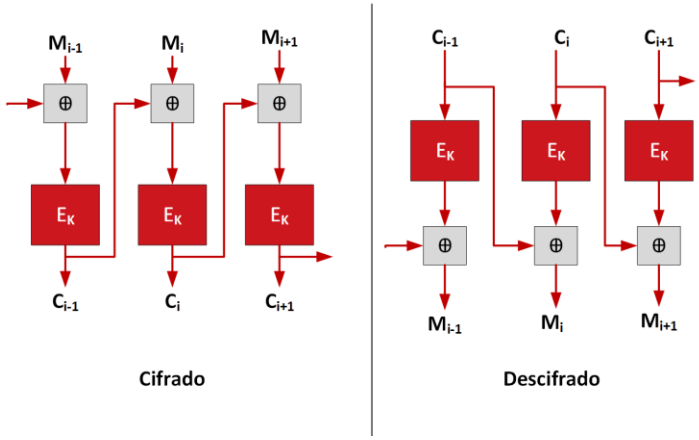


Fig. 108 Modo de Operación en cifrado por bloques tipo CBC

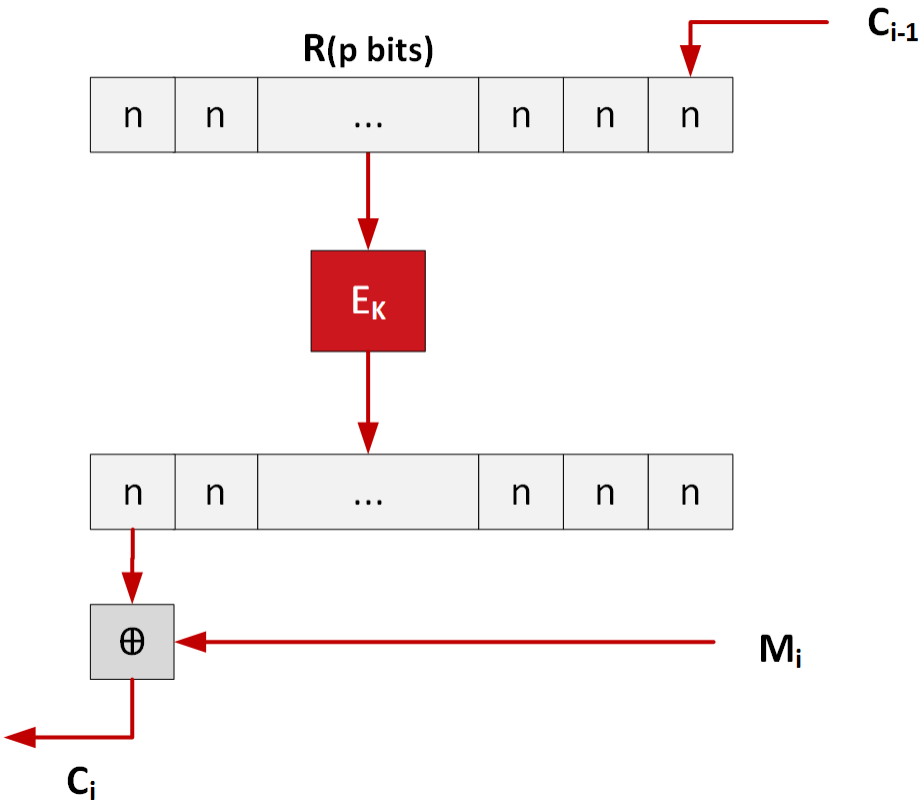


Fig. 109 Modo de Operación en cifrado por bloques tipo CFB

4.3.4. Unidad 5: Criptografía asimétrica o de clave pública

La unidad 5 continúa expandiendo el tema de criptografía moderna, con la introducción de los cifrados asimétricos, los cuales son conocidos por contar con una llave pública y una llave privada para realizar el cifrado y descifrado de los mensajes. El movimiento de datos dentro de las comunicaciones se vuelve más complejo y los diagramas ayudan a comprender los sucesos para realizar comunicaciones de forma segura.

A continuación se muestran diagramas:

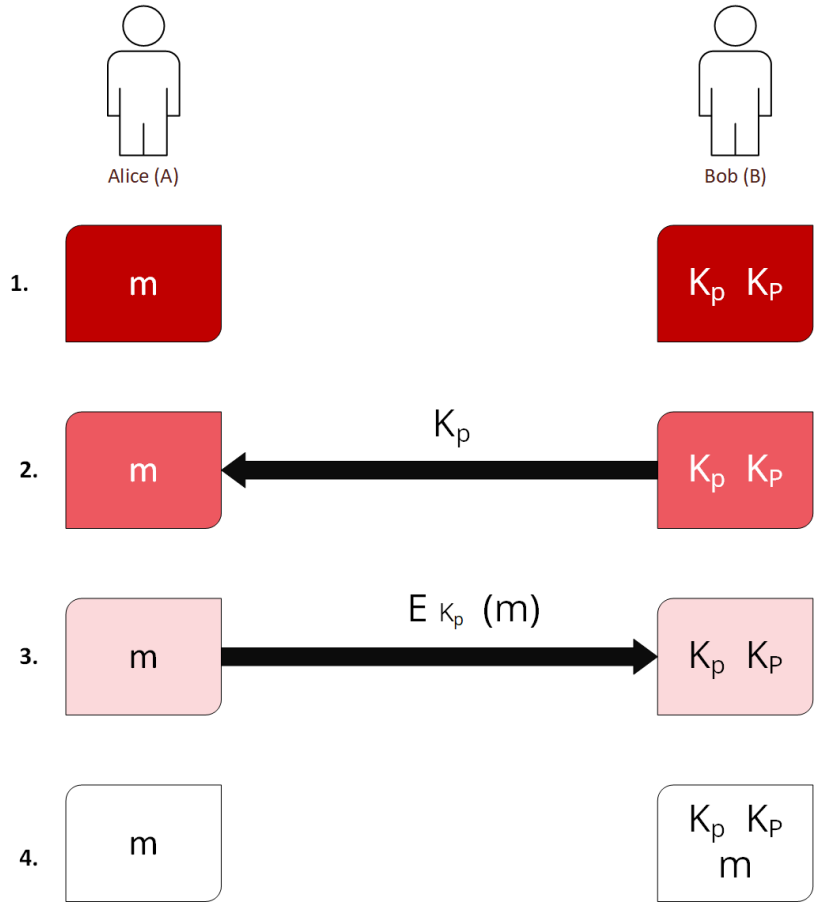


Fig. 110 envío de mensaje empleando algoritmo asimétrico:

Donde:
- m: mensaje en texto plano a enviar.
- K p : llave pública.
- K P : llave privada
- E: mensaje cifrado por K p

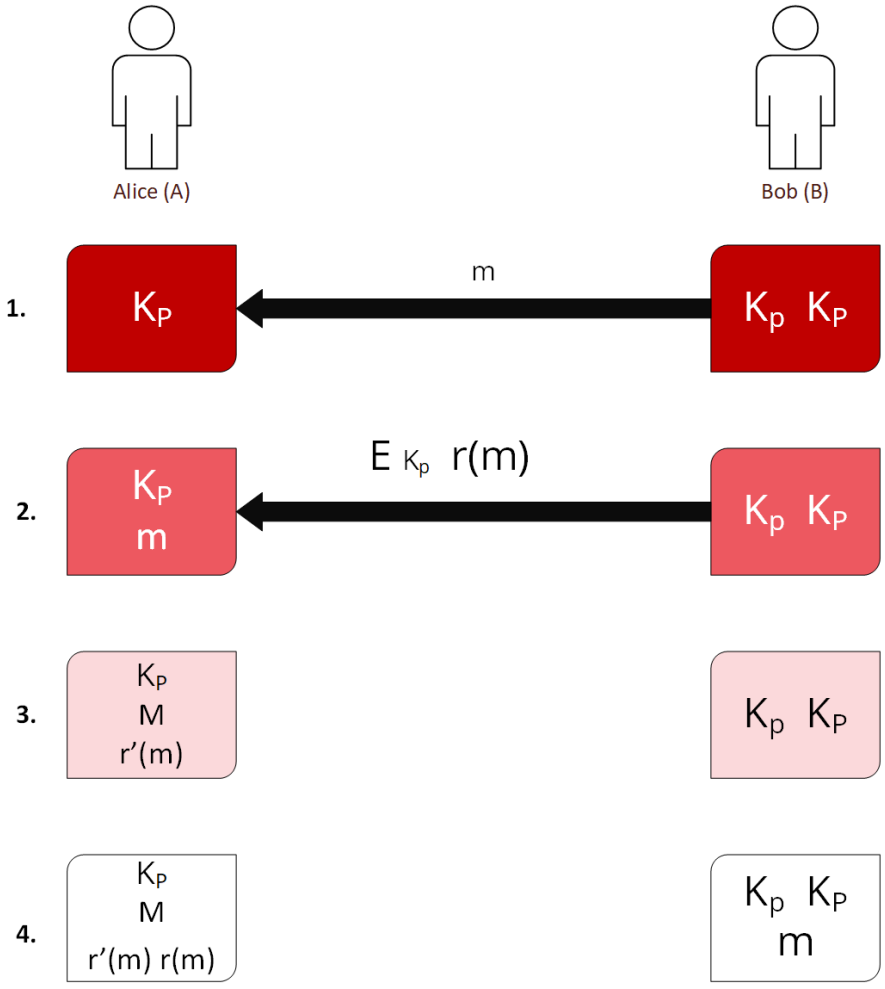


Fig. 111 Ejemplo de firma digital de mensaje empleando algoritmo asimétrico:

Donde:
- m: mensaje en texto plano a enviar.
- K p : llave pública.
- K P : llave privada
- r: resumen del mensaje

Fig. 112 Estructura de una Función Resumen

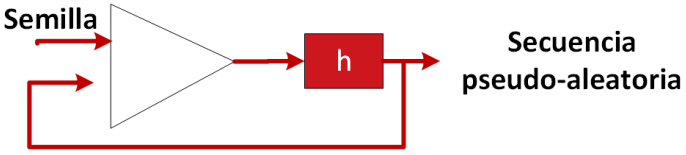


Fig. 113 Estructura de una Función Resumen con semilla

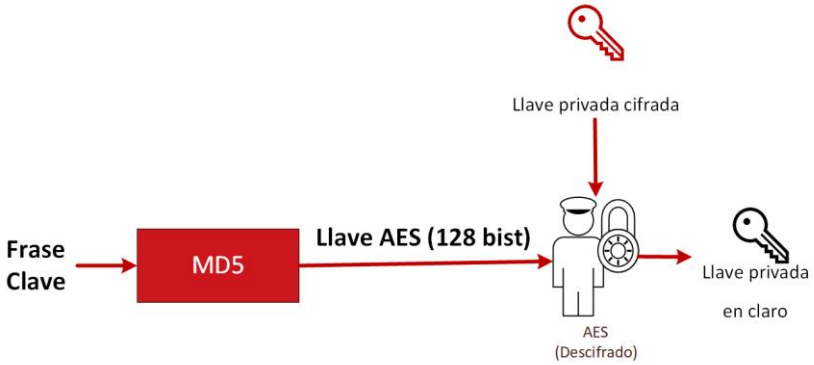


Fig. 114 Esquema de uso hash para obtener una Llave Privada

4.4. Animaciones

El uso de una plataforma web nos da la capacidad de implementar no solo texto e imágenes, sino también contenido dinámico. La implementación de animaciones se ha comprobado como de gran utilidad en la enseñanza en estudiantes a nivel universitario a la hora de memorizar, atender, almacenar y recuperar información adquirida.[2]

Con el uso de animaciones se logra promover una mejor comprensión de la materia, se busca captar la atención de los alumnos que aprenden de diferente manera a la tradicional que se basa en lectura.

La investigación sobre el uso de animaciones en el ámbito educativo (Mayer 2002) muestra que las animaciones acompañadas con narraciones de información mejoran el aprendizaje de los estudiantes. Al igual se comprobó cómo la explicación de temas complejos acompañados de animaciones facilitan la comprensión de los mismos. Con esta implementación se busca capturar la mayor cantidad de sentidos de los estudiantes y evitar la distracción al momento de estudiar.

4.4.1. Apoyo al material

Este tipo de animaciones apoyan el contenido de la materia, no cuentan con audio y son cortas. Se presenta en las unidades de la materia para explicar temas donde la abstracción de la información puede llegar a ser complicada si no se explica de manera visual. Se busca facilitar la comprensión de dichos conceptos mediante artefactos visuales.

A continuación se presentan estas animaciones organizadas por unidad:
Véase en: http://45.33.120.30/gif-gallery

4.4.1.1. Unidad 2: Técnicas clásicas de cifrado

La unidad 2 presenta algoritmos clásicos de cifrado, los cuales son la base de la criptografía moderna, dichos algoritmos son fáciles de comprender utilizando métodos visuales que muestren el movimiento de los caracteres al realizar los pasos de los algoritmos.

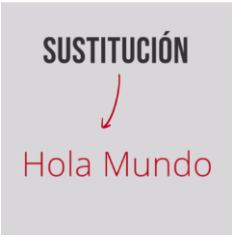


Fig. 115 animación de algoritmos de sustitución, se muestra una palabra la cual modifica las letras del texto original por otras.

[2] Mayer, R. E., Moreno, R. (2002, March), *Animation as an Aid to Multimedia Learning*, Educational Psychology Review, Vol. 14, No. 1.

Fig. 116 Animación de cifrados por transposición, donde las letras de un texto original son intercambiadas de lugar, sin modificar las letras.

Fig. 117 Cifrado Alberti, donde con ayuda de dos círculos con letras es posible cifrar un mensaje utilizando un cifrado de tipo sustitución.

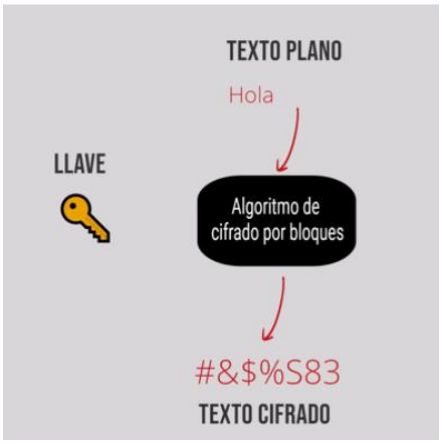


Fig. 118 Animación del funcionamiento básico de un cifrado por bloque

Fig. 119 Animación del funcionamiento básico de un Cifrado Cesar

Fig. 120 Uso de una llave y un texto a cifrar mediante un XOR para producir un cifrado de flujo.

4.4.1.2. Unidad 3: Gestión de claves

La unidad 3 se basa en la seguridad, generación, manejo, procesamiento y administración para las llaves seguras que son utilizadas en los algoritmos de cifrado. Existen diferentes métodos para poder realizar una buena gestión de llaves, estos constan de varios pasos y tienen la facilidad de ser explicados con animaciones donde se simula el inicio de una comunicación segura.

Fig. 121 Animación del protocolo de Diffie-Hellman para el intercambio de llaves privadas

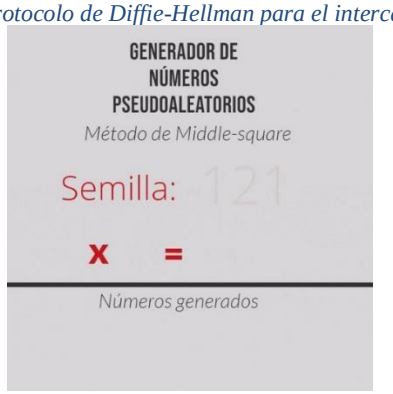


Fig. 122 Animación de un método de generación de números pseudoaleatorios

Fig. 123 Animación de handshake de protocolo TLS/SSL (certificados digitales)

4.4.1.3. Unidad 4: Criptografía simétrica o de clave secreta

Dentro de la unidad 4 se aborda el tema de AES, un algoritmo de cifrado simétrico altamente usado en la actualidad. Los pasos para realizarlo llegan a ser confusos y el uso de animaciones presenta una oportunidad para comprender su funcionamiento de forma visual.

Los pasos para el algoritmo de cifrado AES se presentan a continuación:

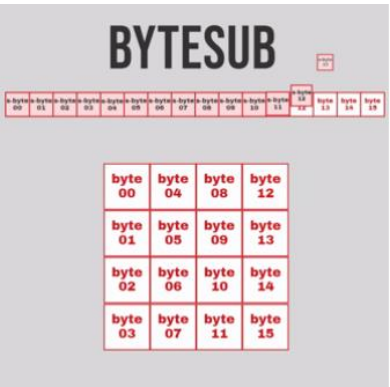


Fig. 124 ByteSub es una sustitución no lineal que se aplica a cada byte de la matriz de estado S.

Fig. 125 Desplaza cíclicamente hacia la derecha las filas de la matriz de estado S.

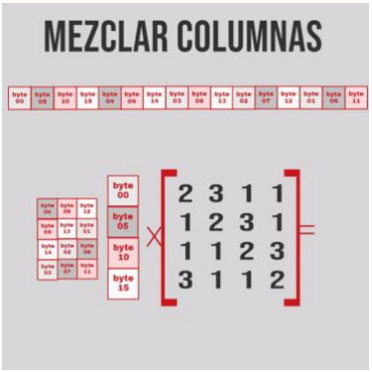


Fig. 126 Se toma cada columna de la matriz y se realiza una multiplicación del vector columna c i con una [¿]

4.4.1.4. Unidad 5: Criptografía asimétrica o de clave pública

En la unidad 5 se presenta el concepto de llaves públicas. Igual es posible abordar los temas de cifrado cuántico, debido a que los algoritmos asimétricos requieren compartir llaves de forma segura y la computación cuántica busca solucionar el problema de las comunicaciones inseguras.

A continuación se muestran capturas de las animaciones que explican los conceptos de llave pública y parte del cifrado cuántico:

Fig. 127 Animación del proceso de una comunicación entre A y B utilizando un cifrado por llave pública

Fig. 128 Animación de los tipos de giros para los qubits

Fig. 129 Animación del protocolo en computación cuántica BB84, parte del servidor

Fig. 130 Animación del protocolo en computación cuántica BB84, parte del cliente

Fig. 131 Animación de la forma en medición de qubits

4.4.2. Videos explicativos

Al inicio de la pandemia de COVID-19 en México se tomó la decisión por parte de la UNAM de realizar las actividades educativas de manera remota, entre ellas las clases. Esto dio paso a que muchos docentes decidieran grabar sus clases, generando material educativo que es posible ser editado y modificado para compartir información en diferentes presentaciones más aptas para la educación a distancia.

Con el permiso de la Dra. Rocío Alejandra Aldeco Pérez, que imparte la asignatura de Criptografía en la Facultad de Ingeniería de la UNAM se tomaron las grabaciones de las clases correspondientes al último periodo del semestre 2020-2. De las grabaciones de las clases en video se extrajo el audio, se editó dicho audio para mantener los temas claros y cortos, posteriormente se añadieron animaciones utilizando el software de *Create Studio.*

El material generado por causas de la pandemia y la necesidad de crear nuevo contenido que sea de apoyo a docentes y estudiantes para el aprendizaje de manera remota permitió crear una serie de 5 videos animados donde se explican temas de interés para la asignatura de criptografía.

El material de los videos forma parte de la unidad 6 (Aplicaciones criptográficas) del temario de la asignatura de Criptografía, donde se evalúan aplicaciones reales, como es el caso de protocolos de comunicación SSH y WPA, al igual, se agrega aplicaciones de la criptografía que han surgido en los últimos años como temas de interés, estos siendo: blockchain, hash chain y la criptografía post-cuántica.

A continuación se presentan las ligas donde se encuentran estos videos y algunas imágenes representativas de ellos:

4.4.2.1. Blockchain

UNAM FI Criptografía: Blockchain
Véase en: https://www.youtube.com/watch?v=UMAXctR6Z2g

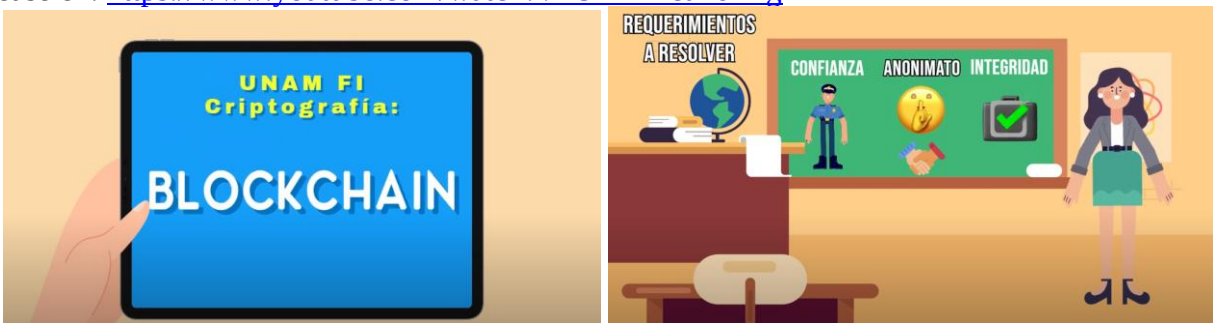


Fig. 132 Capturas de pantalla del video explicativo de blockchain

4.4.2.2. SSH

UNAM FI Criptografía: SSH
Véase en: https://www.youtube.com/watch?v=BgGMzifnNeU

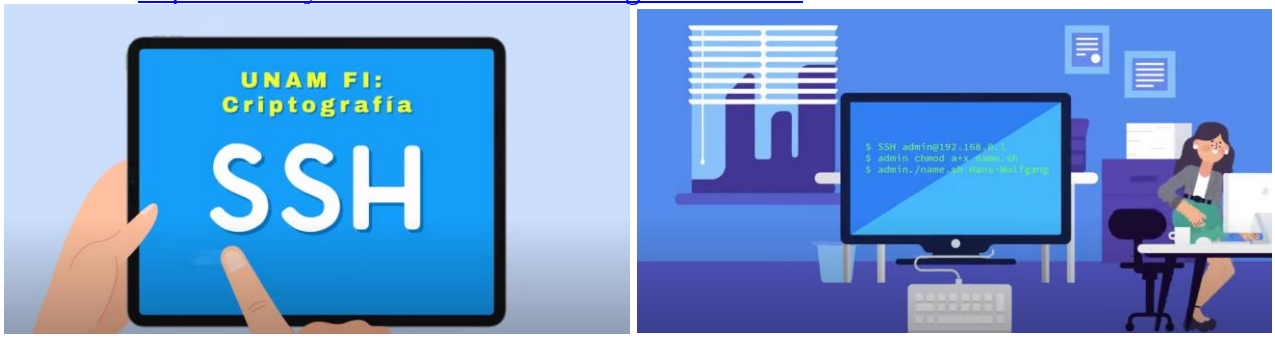


Fig. 133 Capturas de pantalla del video explicativo de SSH

4.4.2.3. WPA

UNAM FI Criptografía: WPA
Véase en: https://www.youtube.com/watch?v=3i1a_JPjyxk

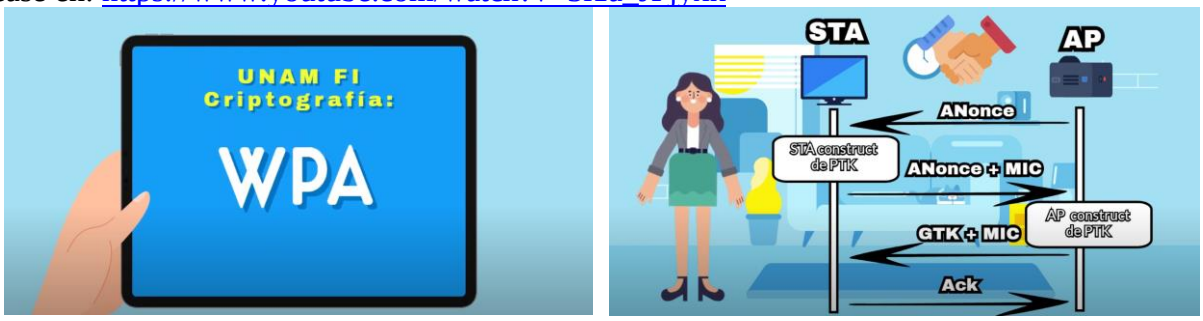


Fig. 134 Capturas de pantalla del video explicativo de WPA

4.4.2.4. Criptografía Post-Cuántica

UNAM FI Criptografía: Post-cuántica
Véase en: https://www.youtube.com/watch?v=O8LuKPKC-4w

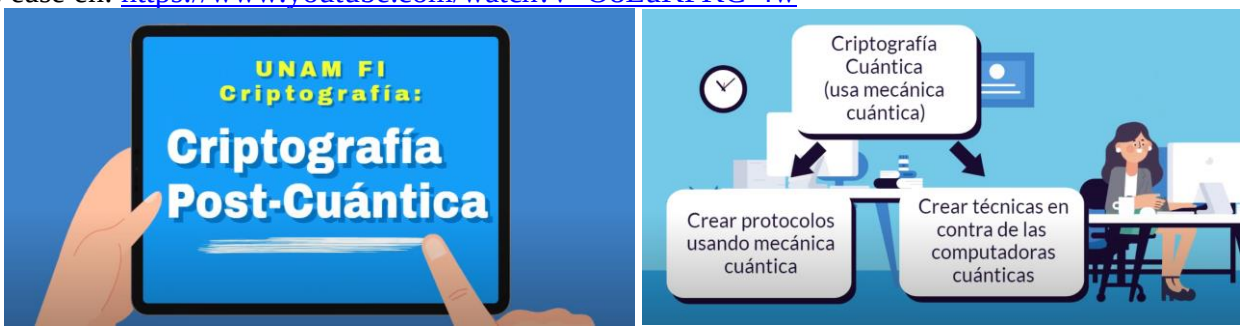


Fig. 135 Capturas de pantalla del video explicativo de Criptografía Post-Cuántica

4.4.2.5. Hash Chain

UNAM FI Criptografía: Hash Chain
Véase en: https://www.youtube.com/watch?v=UkcdmahiJcY

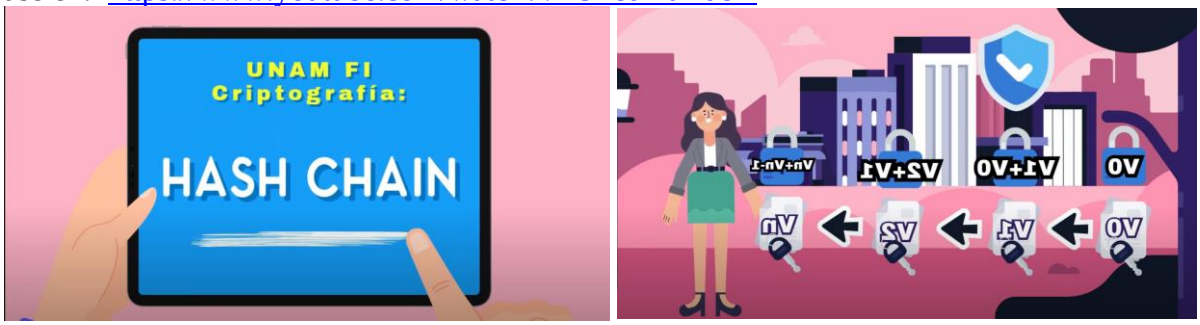


Fig. 136 Capturas de pantalla del video explicativo de Hash Chain

4.5. Actividad didáctica

Se generaron recursos de evaluación para la asignatura con el objetivo de contar con contenido para así proporcionar diferentes tipos de aprendizaje a los estudiantes. Las actividades didácticas más allá de proveer un proceso diferente al aprendizaje de manera teórica, ayuda a reforzar el conocimiento adquirido.

Las actividades didácticas generan calificación y ayudan al refuerzo de los conocimientos adquiridos hasta ese momento en el curso. Se busca que estas actividades sean autogestivas, para que el alumno reciba retroalimentación inmediata. En algunos casos la actividad a realizar debe ser revisada por el profesor de la asignatura, en tal caso se da una rúbrica para el profesor.

La primera actividad, creada para la Unidad 1: Panorama general, presenta los conceptos básicos para la criptografía. Se decidió implementar una sopa de letras donde es necesario buscar la palabra según su definición, con la idea de reforzar un aprendizaje asociativo, al usar al tener una sopa de letras se tiene un apoyo visual y buscando que se comprenda el concepto que expresa la palabra, más allá de soló memorizar el significado de cada una.

La segunda actividad, la cual es parte de la Unidad 2: Técnicas clásicas de cifrado, busca implementar un aprendizaje significativo, al utilizar técnicas vistas anteriormente de cifrados clásicos para encontrar palabras faltantes de un poema y así poner en práctica los puntos teóricos de la unidad.

A continuación se presentan las actividades didácticas organizadas por unidad:

4.5.1. Actividad 1: Sopa de letras (Unidad 1: Panorama general)

RL02-19
Sopa de letras.
Recurso juego para encontrar palabras en cualquier dirección.
Vale dos puntos cada acierto

Definiciones:

1. No es el único medio para proporcionar seguridad a la información, sino, es un conjunto de técnicas para lograrlo.
2. Texto o documento original, se denota por M.
3. Ciencia que estudia e investiga todo lo relacionado con criptografía, esto incluye cifrado y criptoanálisis.
4. Documento o texto cifrado, se denota por C.
5. Aquel que trabaja en el desarrollo de algoritmos criptográficos para la protección de la información.
6. Información (pública o privada) que permite cifrar o descifrar un criptograma.

Usa las definiciones para encontrar las palabras

Respuestas:

1. Criptografía
2. Criptología
3. Criptólogo
4. Textoplano
5. Criptograma
6. Llave

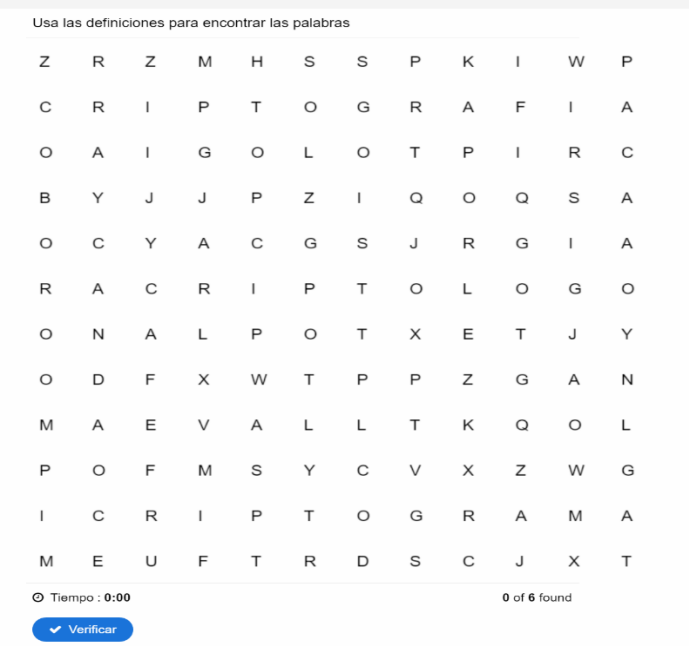


Fig. 137 Ejemplo de intento de Sopa de letras de actividad 1, la sopa de letras cambia con cada intento dentro de la plataforma.

Retroalimentación:
Mostrar el texto que corresponda según la puntuación obtenida

De 8 a 12 puntos
¡Correcto! Tienes los conceptos básicos de criptografía claros, estos son de gran importancia para poder comprender sobre lo que se habla durante la asignatura.

De 4 a 6 puntos
Regular. No olvides repasar los conceptos básicos de criptografía que se desarrollan durante este tema, si tienes duda revisa las fuentes dadas. Al principio puede ser confuso, pero con práctica se aclaran.

De 0 a 2 puntos
Incorrecto. Revisa los temas de la unidad para que quede claro los conceptos básicos de criptografía, esos son de gran importancia para seguir avanzando con la asignatura.

4.5.2. Actividad 2: Poema Cifrado (Unidad 2: Técnicas clásicas de cifrado)

RA06-19 (antes RA-02V2)
Completar escribiendo.
Recurso para textos con respuestas en blanco para completar texteando.
Vale dos puntos cada acierto
Textos en 5 recuadros, pueden ser con bordes de colores diferentes, de acuerdo al diseño de la materia.

Las técnicas clásicas de cifrado sientan las bases de las aplicaciones criptográficas modernas. Conocer los tipos de cifrados clásicos permite una mejor comprensión de temas actuales y avanzados, una mejor comprensión de las técnicas lleva a una mejor aplicación de seguridad en las Tecnologías de la Información y Comunicación.

En el siguiente poema hacen falta algunas palabras, dichas palabras se encuentran cifradas utilizando algunos algoritmos de sustitución y transposición vistos en los temas anteriores. Los criptogramas están asignados a un algoritmo en específico y tienen las llaves necesarias para lograr descifrarlos. Utiliza los métodos necesarios para conseguir las palabras y completar el poema.

Insertar recurso RA06-19
ENTRAÑABLE SECRETO[3].
Por: Adan el 10 de septiembre de 2013
No puedo decir tu nombre
en voz alta, no debo mirarte
como mis ojos _____ [Polybios] verte,
debo guardar el secreto

Es un secreto que quiero
gritar al cielo, es un secreto que
pesa tanto en mi alma, y es por
ti que debo cerrar **_____** [grupos y de series]

Me pides **_____** [columnas con clave] este secreto
y cambio del silencio de mis
labios me **_____** [Afín] besos
y caricias que a nadie más has dado,

Es un entrañable secreto,
un secreto que **_____** [Vigenère] compartir
contigo, que debo callar,
si te quiero mantener a mi lado.

[3] Adan, (Septiembre 10 2013) ENTRAÑABLE SECRETO. Recuperado de https://poesialibre.wordpress.com/2013/09/10/entranable-secreto/

Polybios:
Llave: secreto
Criptograma: BB DB CB AB CE BB DB
Filas y columnas: A B C D E
La i y la j se encuentran en la misma casilla

Solución: anhelan

	A	B	C	D	E
A	S	E	C	R	T
B	O	A	B	D	F
C	G	H	i/j	K	L
D	M	N	P	Q	U
E	V	W	X	Y	Z

Afín:
Llave: a = 5, b = 3, n = 26
Criptograma: kxhdgdp uzp
Alfabeto: a b c d e f g h i j k l m n o p q r s t u v w x y z

Solución
Números
10 23 7 3 6 3 15 20 25 15
a valor de letras
17 4 6 11 6 18 19 20 18
A texto
regalas tus

Vigenère:
Llave: secreto
Criptograma: iykvvh

Solución: quiero

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
B	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a
C	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b
D	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c
E	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d
F	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e
G	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f
H	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g
I	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h
J	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i
K	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j
L	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k
M	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l
N	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m
O	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n
P	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
Q	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p
R	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q
S	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r
T	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s
U	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t
V	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u
W	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v
X	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w
Y	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x
Z	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y

Fig. 138 Tabla de Vigenère

Cifrado por columnas con clave:
Llave: 4, pesa
Criptograma: RXUAGDAR

Solución:

P	E	S	A
G	U	A	R
D	A	R	X

A	E	P	S
R	U	G	A
X	A	D	R

guardar

Cifrado transposición de grupos y de series:
Llave: bloques de 4**,** posiciones 2431
Criptograma: ilsmboiaxxxs

Solución:
Ilsm boia xxxs
2431 2431 2431

misl abio sxxx
mis labios

Retroalimentación:
Mostrar el texto que corresponda según la puntuación obtenida

De 8 a 10 puntos
¡Correcto! Al aplicar las diferentes técnicas para descifrar es posible comprender mejor el funcionamiento de los criptosistemas debido a que se aplican los conocimientos aprendidos y es fácil entender funciona un cifrado y cómo se puede invertir. Los criptosistemas vistos en este tema no son lo suficientemente seguros para usarse hoy en día debido al poder computacional actual, ya que los algoritmos clásicos basan su seguridad en ocultar el proceso que utilizan más que en ocultar las llaves. Es decir, si se sabe el proceso para cifrar es muy fácil descifrar los mensajes.

De 4 a 6 puntos
Regular. No olvides repasar los algoritmos vistos durante este tema, si tienes duda revisa las fuentes dadas. La criptografía clásica permite utilizar algoritmos para cifrar y descifrar, para ello es importante poner atención al proceso y seguir los pasos en el orden descrito. Al principio puede ser confuso, pero con práctica se aclara el funcionamiento.

De 0 a 2 puntos
Incorrecto. Revisa los temas de la unidad para que quede claro el procedimiento para cifrar y descifrar con diversas técnicas clásicas. La criptografía clásica sienta las bases de la criptografía moderna, por ello es importante su estudio y comprensión.

4.6. UAPA

Al hacer el desarrollo de la información para el curso se dio la oportunidad de crear contenido de carácter abierto para el apoyo al aprendizaje, con apoyo de la entidad CUAIEED se planeó integrar contenido de la asignatura de Criptografía a la plataforma de UAPA (https://uapa.cuaieed.unam.mx/) la definen como "Un portal donde se encontrarán recursos educativos llamados ***Unidades de Apoyo para el Aprendizaje (UAPA)***, estas unidades abordan diversos temas y están diseñadas para estudiarse de manera autónoma y gratuita por cualquier persona interesada en ampliar sus conocimientos." Una UAPA cuenta con una introducción al tema a explicar, un desarrollo que pueda ser entendido por cualquier interesado en el tema y actividades que ayuden a reforzar el conocimiento adquirido.

En este proyecto se desarrollaron dos UAPA's de los temas *AES (Advance Encryption Standar)* y *Funciones Hash* que a continuación se presentan:

4.6.1. AES (Advanced Encryption Standard)

Introducción

Desde la antigüedad el humano ha ideado maneras de comunicarse y mantener la confidencialidad de la información, sobre todo en tiempos de conflictos. En un inicio se crearon sistemas criptográficos los cuales se agrupa dentro de la *Criptografía Clásica*. Con el auge de las computadoras surge lo conocido como *Criptografía Moderna* que utiliza los mismos principios que en la Criptografía Clásica, sin embargo, ahora no se opera sobre los caracteres de un texto, si no sobre bits (1,0), siendo estos la forma en la que los textos se representan dentro de una computadora.

Dentro de la Criptografía Moderna existen dos grandes grupos, los cifrados asimétricos y los cifrados simétricos. Los cifrados simétricos consiste en utilizar una misma llave para cifrar y descifrar información. Funciona como un candado que tiene una llave con la que se abre y cierra.

Un algoritmo de Cifrado Simétrico puede funcionar de dos formas: por flujo y por bloque. Cuando el algoritmo es por flujo se cifra bit a bit hasta cifrar todos los bits que conformen la información. En un algoritmo por bloque se toma un bloque o conjunto de bits y los opera para generar un nuevo bloque de cifrado.

En este tema se hablará del algoritmo de cifrado AES, el cual es un cifrado simétrico y por bloque, se buscará dar a conocer sus orígenes y funcionamiento para la comprensión básica del mismo.

Orígenes

AES es un es un conjunto de especificaciones que sientan las bases para el desarrollo de algoritmos de cifrados seguros creado y administrado por el *National Institute of Standards and Technologies* (NIST) el cual promueve la innovación en el área de criptografía. En octubre del año 2000, NIST adoptó un nuevo algoritmo de cifrado con fines no militares con el fin de sustituir a DES. Este algoritmo fue Rijndael, incluido en el ISO/IEC 18033-3 y disponible actualmente en todas las bibliotecas de cifrado, además de actualmente ser el único aprobado por la NSA.

Rijndael es una familia de cifradores por bloque con varios tamaños de llave, AES toma un subconjunto de estos algoritmos para integrarlos el estándar. Fueron desarrollados por Vincent Rijmen y Joan Daemen quienes presentaron su algoritmo ante NIST durante el proceso de selección AES. Durante el resto del curso utilizaremos AES para referirnos a la Rijndael.

AES es un algoritmo de cifrado por bloque diseñado para utilizar una llave de tamaño variable. Utiliza bloques de 16 bytes y una llave de 128, 192 o 256 bits. Las operaciones que realiza son a nivel byte dentro de un **Campos de Galois**. Las demás operaciones se efectúan sobre registros de 32 bits, pero estos pueden interpretarse como polinomios de grado inferior a 4 (polinomios en 2^8 en Campos de Galois).

Campos de Galois

Los Campos de Galois (conocido como GF por sus siglas del inglés *Galois Field*) son una estructura algebraica que define una cantidad finita de números que existen dentro del campo, en este caso el campo tiene un tamaño de un byte, y los números existentes en el campo, son la combinación de bits en un byte.

0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 1
…
1 1 1 1 1 1 1 1

Todas las operaciones que se utilizan en estos campos dan como resultado números dentro del mismo. Es decir, jamás tendremos un resultado con más o menos bits que 1 byte. Las operaciones deben cumplir con ciertas reglas que hagan posible lo anterior. Cabe mencionar que una operación puede existir en un campo, sin embargo, no necesariamente se refiere a la operación que utilizamos comúnmente. Por ejemplo: una *multiplicación* en un Campo de Galois puede significar un XOR con módulo.

Estructura

AES no sigue la estructura Feistel, en cambio establece rondas con 4 operaciones invertibles, estas se realizan en forma matricial en vez de arreglo como en DES. Las operaciones forman tres capas diseñadas para dificultar el criptoanálisis lineal y diferencial.

- **Capa de mezcla lineal:** *DesplazarFila* y *MezclarColumna*. Proporciona alto nivel de difusión a lo largo de varias rondas.
- **Capa no lineal:** *ByteSub*. Aplicación paralela de S-Cajas con propiedades de no linealidad.
- **Capa de adición de clave:** XOR exclusivo entre el estado intermedio y la subclave de cada ronda.

Elementos

AES es un algoritmo que se basa en someter a un texto plano a un número determinado de rondas para obtener un bloque de texto cifrado. A este bloque se le llama *estado*, y puede representarse mediante una matriz rectangular de bytes con 4 filas y N_b columnas. Por ejemplo, si nuestro bloque tiene 160 bits entonces N_b=5, como se muestra en la siguiente tabla:

$a_{0,0}$	$a_{0,1}$	$a_{0,2}$	$a_{0,3}$	$a_{0,4}$
$a_{1,0}$	$a_{1,1}$	$a_{1,2}$	$a_{1,3}$	$a_{1,4}$
$a_{2,0}$	$a_{2,1}$	$a_{2,2}$	$a_{2,3}$	$a_{2,4}$
$a_{3,0}$	$a_{3,1}$	$a_{3,2}$	$a_{3,3}$	$a_{3,4}$

Tabla 1

La llave tiene una estructura similar a la del estado, 4 filas y N_k columnas. Si nuestra llave tiene, por ejemplo 128 bits, N_k = 4, como se muestra en la siguiente tabla:

$k_{0,0}$	$k_{0,1}$	$k_{0,2}$	$k_{0,3}$
$k_{1,0}$	$k_{1,1}$	$k_{1,2}$	$k_{1,3}$
$k_{2,0}$	$k_{2,1}$	$k_{2,2}$	$k_{2,3}$
$k_{3,0}$	$k_{3,1}$	$k_{3,2}$	$k_{3,3}$

Tabla 2

En algunos casos, tanto el estado como la clave se consideran como vectores de registros de 32 bits, estando cada registro constituido por los bytes de la columna correspondiente, ordenados de arriba a abajo.

El bloque que se pretende cifrar o descifrar se traslada directamente byte a byte sobre la matriz de estado, siguiendo la secuencia $a_{0,0}\, a_{1,0}\, a_{2,0}\, a_{3,0}\, a_{0,1}\ldots$, al igual que los bytes de la llave con la misma secuencia, $k_{0,0}\, k_{1,0}\, k_{2,0}\, k_{3,0}\, k_{0,1}\ldots$

En la siguiente tabla se muestran los tamaños válidos de llave, así como su respectivo número de filas y número de columnas para las matrices de llave y bloque, y el número de rondas que se van a operar.

	$N_b = 4$ (128 bits)	$N_b = 6$ (192 bits)	$N_b = 8$ (256 bits)
$N_k = 4$ (128 bits)	10	12	14
$N_k = 6$ (192 bits)	12	12	14
$N_k = 8$ (256 bits)	14	14	14

Tabla 3

Cada ronda utiliza una sub-llave generada de la expansión de la llave original, el algoritmo AES con n rondas tiene la siguiente estructura:

1. Calcular sub-llaves K_0, K_1, K_2, … K_n a partir de llave K
2. $S \leftarrow B \oplus K_0$
3. Para i = 1 hasta n:
   Aplicar ronda i-ésima del algoritmo con sub-llave K_i

Donde:

- **B:** Es el bloque a cifrar
- **S:** Es la matriz de estado
- **K:** Es la llave principal

Como cada ronda es una sucesión de funciones reversibles, el proceso de descifrado consiste en aplicar las inversas de cada función en el orden contrario con las mismas sub-llaves K_i que, en el cifrado, solo que comenzado por el último bloque.

Proceso de cifrado y descifrado (bloques de 128, 192 y 256 bits)

Ya que AES nos permite utilizar diferentes longitudes de bloque como de llave, el número de rondas varía. Reutilizando la tabla de la sección anterior donde se muestran cuántas rondas son necesarias en función de N_b y N_k.

	$N_b = 4$ (128 bits)	$N_b = 6$ (192 bits)	$N_b = 8$ (256 bits)
$N_k = 4$ (128 bits)	10	12	14
$N_k = 6$ (192 bits)	12	12	14
$N_k = 8$ (256 bits)	14	14	14

Tabla 4

Todas las rondas tienen la siguiente estructura (menos la última ronda):

1. $S \leftarrow ByteSub(S)$
2. $S \leftarrow DesplazarFila(S)$
3. $S \leftarrow MezclarColumnas(S)$
4. $S \leftarrow K_i \oplus S$

Donde:

- **S:** Es la matriz de estado
- **K_i:** Es la sub-llave correspondiente a la ronda i–ésima

La última ronda es igual que los anteriores, pero no realiza el paso 3. A continuación una descripción detallada de cada una de las operaciones que se realizan durante las rondas.

- *ByteSub* (Fig. 124, p. 78)

  ByteSub es una sustitución no lineal que se aplica a cada byte de la matriz de estado ***S***. Utiliza una S-caja de 8*8 invertible, que se obtiene de dos transformaciones.

  Esta S-caja garantiza que sea reversible y que las sustituciones no son inversas. El mapeo que genera esta S-caja es un conjunto de operaciones lineales dentro del campo finito.

  La función inversa de *ByteSub* sería la aplicación de la inversa de la s-caja correspondiente a cada byte de la matriz de estado.

- *DesplazarFila* (Fig. 125, p. 78)

  Desplaza cíclicamente hacia la derecha las filas de la matriz de estado *S*. Cada fila f_i se desplaza c_i número de veces, donde cada c_i es diferente. Los desplazamientos también están en función del tamaño de N_b como se muestra a continuación:

N_b	c_1	c_2	c_3
4	1	2	3
6	1	2	3
8	1	3	4

Tabla 5

  Para el proceso inverso, se realiza el mismo número de desplazamientos para cada fila, pero hacia la izquierda.

- MezclarColumnas (Fig. 126, p. 78)

  Se toma cada columna de la matriz y se realiza una multiplicación del vector columna c_i con una matriz definida. Esta multiplicación es definida por el campo finito, no una multiplicación normal. Para el proceso de descifrado se realiza el mismo procedimiento, pero la matriz utilizada es la inversa del proceso de cifrado. La matriz tiene la siguiente forma:

Los coeficientes de la matriz se obtienen a partir de un polinomio, y la multiplicación de la columna c_i es una multiplicación de polinomios entre el mismo y el vector columna que se toma como un polinomio con coeficientes dentro del campo finito. El polinomio se multiplica módulo x^4+1 por (Ecu. 1):

$$c(x) = 03x^4+01x^2+01x+02$$
Ecu. 1

donde 03 es el valor hexadecimal que se obtiene concatenando los coeficientes binarios del polinomio correspondiente en 00000011, es decir, $x+1$ y así sucesivamente. La operación inversa se obtiene multiplicando cada columna c_i de la columna de la matriz de estado *S* por el polinomio (Ecu. 2):

$$c(x) = 0Bx^4+0Dx^2+09x+0E$$
Ecu. 2

- Cálculo de sub-llaves

Para generar las sub-llaves que serán utilizadas en cada ronda se realiza una expansión de la llave original, dando con resultado una secuencia de $4*(n+1)*N_b$ bytes. Las sub-llaves se forman al tomar consecutivamente un bloque del tamaño de la matriz de estado, esto para cada ronda.

La función tiene dos versiones, para $N_k <= 6$ y $N_k > 6$. Sea la llave I un vector de bytes de $4*N_k$ y W(i) un vector de $N_b*(n+1)$ registros de 4 bytes, y sea n el número de rondas. Las funciones **Sub** y **Rot** se refieren a aplicar una S-caja de AES a un byte y una rotación a la izquierda, respectivamente.

Rc(*j*) es una constante definida como Rc(*j*) = (R(*j*), 0, 0, 0) donde Cada R(*j*) es el elemento de un Campo de Galois correspondiente al valor $x^{(i-1)}$

Si Nk ≤ 6:

Para *i* de 0 a N_k-1:

$W_{(i)} = (K_{(4 \cdot i)}, K_{(4 \cdot i + 1)}, K_{(4 \cdot i + 2)}, K_{(4 \cdot i + 3)})$

Para *i* de N_k a $N * (n + 1)$:

tmp = $W_{(i - 1)}$

Si *i* mod $N_k = 0$

tmp = **Sub**(**Rot**(tmp)) ⊕ R(i/N_k)

$W_{(i)} = W_{(i - Nk)} \oplus$ tmp

Si Nk > 6:

Para *i* de 0 a $N_k - 1$:

$W_{(i)} = (K_{(4 \cdot i)}, K_{(4 \cdot i + 1)}, K_{(4 \cdot i + 2)}, K_{(4 \cdot i + 3)})$

Para *i* de N_k a $N_b \cdot (n + 1)$:

tmp = $W_{(i - 1)}$

Si i mod $N_k = 0$

tmp = **Sub**$($ **Rot**$($tmp$))$ ⊕ Rc(i/N_k)

Si i mod $N_k = 4$

tmp = **Sub**(tmp)

$W_{(i)} = W_{(i - Nk)}$ ⊕ tmp

En resumen:

1. La primera operación sustituye el byte original por un nuevo byte
2. La segunda operación mezcla los bytes de cada fila
3. La tercera operación mezcla los valores de las columnas entre ellas.

Modos de Operación

Un modo de operación es un algoritmo que utiliza un cifrador por bloques para proveer seguridad a la información. Un modo de operación describe cómo aplicar repetidamente una operación de cifrado de bloque simple para transformar de forma segura cantidades de datos mayores que un bloque.

Modo ECB (Fig. 107, p. 70)
El modo ECB (*Electronic Code Book*) es el método más sencillo y obvio de aplicar un algoritmo de cifrado por bloques. Simplemente se subdivide la cadena que se quiere codificar en bloques del tamaño adecuado y se cifran todos ellos empleando la misma clave

- Ventajas: Permite codificar los bloques independientemente de su orden. También es resistente a errores, pues si uno de los bloques sufriera una alteración, el resto quedaría intacto.
- Desventajas: Si el mensaje presenta patrones repetitivos, el texto cifrado también los presentará, y eso es peligroso, sobre todo cuando se codifica información muy redundante. Un atacante puede efectuar un ataque estadístico y extraer bastante información.

Modo CBC (Fig. 108, p. 70)
El modo CBC (*Cipher Block Chaining Mode*) incorpora un mecanismo de retroalimentación en el cifrado por bloques. Esto significa que el cifrado de bloques anteriores condiciona el cifrado del bloque actual, por lo que será imposible sustituir un bloque individual en el mensaje cifrado. Esto se consigue efectuando una operación XOR entre el bloque del mensaje que queremos cifrar y el último criptograma obtenido.

Modo CFB (Fig. 109, p. 71)
El modo de operación CFB (*Cipher-Feedback Mode*) permite codificar la información en unidades inferiores al tamaño del bloque, lo cual permite aprovechar totalmente la capacidad de transmisión del canal de comunicaciones, manteniendo además un nivel de seguridad adecuado.
El esquema de funcionamiento de este modo de operación se muestra en la siguiente figura:

Donde:

- **p** Es el tamaño de bloque del algoritmo simétrico.
- **n** Es el tamaño de los bloques que queremos transmitir (n es divisor de p).
- $\mathbf{M_i}$ Es el i−ésimo bloque del texto claro, de tamaño n.

Se emplea un registro de desplazamiento **R** de longitud **p** y lo cargamos con un vector de inicialización. Ciframos el registro **R** con el algoritmo simétrico y obtenemos en **r** sus **n** bits más a la izquierda. El bloque que deberemos enviar es $\mathbf{C}_i = \mathbf{r} \oplus \mathbf{M_i}$. Desplazamos **R** n bits a la izquierda e introducimos $\mathbf{C}_i$ por la derecha. Para descifrar basta con cargar el vector de inicialización en **R** y cifrarlo, calculando **r**. Entonces $\mathbf{M_i} = \mathbf{r} \oplus \mathbf{C}_i$. Desplazamos luego **R** e introducimos $\mathbf{C}_i$ por la derecha como hacíamos en el algoritmo de cifrado. Si **n = p**, el modo CFB queda reducido al modo CBC.

Aplicaciones

En sus inicios el objetivo de AES era ser utilizado para la protección de información por el gobierno de EE.UU. Actualmente AES es el algoritmo de cifrado por excelencia, es utilizado por el protocolo HTTPS para enviar archivos por internet, se utiliza en los *routers* junto con WPA 2 para las conexiones seguras a ellos, en el protocolo SSL/TSL para asegurar conexiones. Y por supuesto, cualquiera puede acceder a herramientas que permitan cifrar archivos con AES. Con el internet de las cosas, este algoritmo es una gran opción para el cifrado en estos sistemas gracias a su eficiencia y seguridad.

Análisis de seguridad

AES es considerado uno de los algoritmos más seguros gracias a su diseño, que prevé muchas de las técnicas de criptoanálisis que pudieran utilizarse para romper el algoritmo y las debilidades que presentaba DES y 3DES. Algunos de los puntos a considerar:

- El tamaño de la llave es lo suficientemente grande para hacer de un ataque de fuerza bruta algo muy costoso.
- No existe simetría en las llaves
- Resistente a criptoanálisis lineal y diferencial

COMPONENTE	DESARROLLO
Actividad 1	Busca una herramienta que te permita cifrar archivos utilizando AES a partir de una contraseña y contesta lo siguiente: 1. ¿Qué sucede con el archivo después de cifrarlo? 2. ¿Qué sucede si intentas descifrar el archivo utilizando una contraseña diferente? (Evaluado según la **Rúbrica de trabajos escrito, p. 137**)
Autoevaluación	Contesta si la afirmación es verdadera o falsa. 1. AES utiliza una llave simétrica. **a. Falso:** Incorrecto Recuerda que AES utiliza la misma llave para cifrar y descifrar, por eso se dice que es de llave simétrica. **b. Verdadero:** Correcto La llave simétrica es la que se utiliza para cifrar y descifrar información. 2. AES funciona cifrando bloques de bits. **a. Falso:** Incorrecto AES es un algoritmo de cifrado simétrico por bloque, cifra dividiendo el texto plano en bloques de tamaño fijo. **b. Verdadero:** Correcto AES es un algoritmo de cifrado simétrico por bloque**.** 3. En AES, aunque un bloque no esté completo se puede operar. **a. Falso:** Correcto Cuando el último bloque no está lleno por completo se agregan bits arbitrarios al final para completar. **b. Verdadero:** Incorrecto En AES los bloques son de tamaño fijo (16 bytes), cuando el texto cifrado no es suficiente para llenar el último bloque por completo se agregan bits arbitrarios al final para completar. 4. El cifrado por bloques utiliza rondas para operar los bits. **a. Falso:** Correcto

	En el cifrado por bloques no es requerimiento opera por rondas, depende de cada algoritmo. AES, por ejemplo, sí utiliza rondas. **b. Verdadero:** Incorrecto El cifrado por bloques no necesariamente opera por rondas. El cifrado AES sí utiliza rondas para procesar los bloques. 5. AES se puede utilizar para cifrar documentos. **a. Falso:** Incorrecto AES se puede utilizar para cifrar documentos, existen muchas herramientas que te permiten lograrlo a partir de una llave en forma de contraseña, haciendo que la información sea ilegible a menos de que se utilice la misma llave para descifrar. **b. Verdadero:** Correcto AES puede cifrar y descifrar texto de diferentes longitudes.

4.6.2. Funciones HASH

Introducción
Una de las aplicaciones fundamentales de la criptografía moderna es la Función Hash, igual conocida
como función solo de ida, función unidireccional o función resumen.
El contenido del tema va dirigido a cualquier persona que busque comprender el funcionamiento básico de las funciones hash y sus aplicaciones.

Origen[4]
Una de las aplicaciones fundamentales de la criptografía moderna es la Función Hash, igual conocida como función solo de ida, función unidireccional o función resumen. Los valores que regresa una función hash son conocidos como: **valor hash, código hash, *hash sum,*** o simplemente **hash.**

La función hash se puede definir como una función computacionalmente eficiente que asigna cadenas binarias o mensajes de longitud arbitraria a cadenas binarias de cierta longitud fija o constante, generalmente más pequeñas, llamadas valores hash. Entre sus aplicaciones están la emisión de certificados, las firmas digitales, la generación de llaves y la verificación de contraseñas. (Fig. 116)

Mensaje = **m** Función Resumen **= h(m)**
Resultado de la función resumen = $\mathbf{r_i}$

Las funciones hash generalmente son públicamente conocidas y no implican el uso de llaves secretas. Cuando se utilizan para detectar si la entrada del mensaje ha sido alterada, se denominan códigos de detección de modificación (por sus siglas en inglés MDC *modification detection codes*). De la misma manera existen funciones hash que involucran el uso de una llave secreta para proporcionar autenticación de origen de datos, así como integridad de datos, Esta técnica se denomina código de autenticación de mensajes (por sus siglas en inglés MAC *message authentication code*).

[4] Scott, T. [Computerphile]. (2013, Noviembre 8). Hashing Algorithms and Security - Computerphile. [Archivo de Video] Youtube. Recuperado de https://www.youtube.com/watch?v= b4b8ktEV4Bg

Funciones solo de ida y sus propiedades

La función hash debe cumplir con ciertas propiedades para ser considerada una función hash criptográfica segura, las cuales son:

1- **Unidireccionalidad**: Si tenemos un resumen **h(m),** debe ser computacionalmente imposible encontrar M a partir de dicho resumen. La función **h(m)** no tiene función inversa (no es una función uno a uno).
2- **Compresión**: A partir de un mensaje de cualquier longitud, el resumen **h(m)** debe tener una longitud fija. Normalmente la longitud de **h(m)** es menor que el mensaje **m.**
3- **Facilidad de cálculo:** Dado cualquier mensaje **m** debe computacionalmente fácil calcular **h(m).**
4- **Difusión:** Dado un resumen **h(m)** debe ser una función compleja de todos los bits del mensaje **m:** si se modifica un solo bit del mensaje **m, h(m)** deberá cambiar la mitad de sus bits aproximadamente.
5- **Colisión simple**: Debe ser computacionalmente imposible si se conoce **m**, encontrar otro **m**' tal que **h(m)** = **h(m')**. Esto se conoce como resistencia débil a las colisiones.
6- **Colisión fuerte**: Debe ser computacionalmente difícil encontrar un par **(m, m')** de forma que **h(m) = h(m').** Esto se conoce como resistencia fuerte a las colisiones.

Hash de mensajes

Una función hash es una función unidireccional que crea una huella digital de longitud fija del mensaje de entrada. La longitud de la entrada de la función no debe afectar el funcionamiento, pero la salida **h(m)** siempre tendrá una longitud fija (normalmente de 128 bits, 160 bits o 256 bits)
De esta manera una función hash deberá:

- Aceptar entradas de cualquier tamaño.
- Producir salidas de longitud fija a cualquier tipo de entrada.
- El hash no deberá revelar ninguna información sobre la entrada.
- Debería ser imposible producir un hash específico.
- Debería ser imposible encontrar dos mensajes diferentes que produzcan el mismo resultado hash.

Estructura de una Función Hash

En general, las funciones hash tienen una estructura similar a la de las funciones de compresión, que dan como resultado bloques de longitud n a partir de bloques de longitud m. Estas funciones se encadenan de forma iterativa, haciendo que la entrada en el paso **i** sea función del **i-ésimo** bloque del mensaje y de la salida del paso $\mathbf{i-1}$. En general, se suele incluir en alguno de los bloques del mensaje m (al principio o al final), información sobre la longitud total del mensaje. De esta forma se reducen las probabilidades de que dos mensajes con diferentes longitudes den el mismo valor en su resumen. (Fig. 113, p. 74)

Ahora veamos un ejemplo del uso de una función hash en un protocolo llamado protocolo de desafío-respuesta (en inglés challenge-response protocol). Estos protocolos tienen como objetivo la autenticación de entidades mediante el uso de un desafío o problema, si la entidad conoce la solución a este problema, entonces podemos confiar en ella.

Imaginemos el siguiente escenario.

1- Alicia plantea a Beto un problema matemático difícil y afirma que ya lo ha resuelto.
2- A Beto le gustaría resolver el problema, pero de la misma manera le gustaría asegurarse de que Alicia no esté mintiendo.
3- Por lo tanto, Alicia escribe su solución, calcula el hash de la misma y le envía a Beto este valor hash (mientras mantiene la solución en secreto).
4- Después de unos días, cuando Beto encuentra la solución al problema, Alicia puede demostrar que tenía la solución correcta al hacer que Beto calcule el hash de su resultado y verifique que coincida con el valor de hash que Alicia le envío. De esta manera podemos garantizar que ambos comparten un valor que es secreto para el resto.

Funcionamiento de MD5 y sus ataques

Se trata de uno de los más populares algoritmos de generación de hash, debido en gran parte a su inclusión en las primeras versiones de PGP[5]. Resultado de una serie de mejoras sobre el algoritmo MD4[6], diseñado por Ron Rivest, procesa los mensajes de entrada en bloques de 512 bits, y produce una salida de 128 bits.

Siendo **m** un mensaje de **b** bits de longitud, en primer lugar, se alarga **m** hasta que su longitud sea exactamente 64 bits inferior a un múltiplo de 512. El alargamiento se lleva a cabo añadiendo un 1 seguido de tantos ceros como sea necesario. En segundo lugar, se añaden 64 bits con el valor de **b**, empezando por el byte menos significativo. De esta forma tenemos el mensaje como un número entero de bloques de 512 bits, y además le hemos añadido información sobre su longitud.

Seguidamente, se inicializan cuatro registros de 32 bits con los siguientes valores hexadecimales[7]:

A = 67452301
B = EFCDAB89
C = 98BADCFE
D = 10325476

[5] Pretty Good Privacy (PGP privacidad bastante buena) es un programa desarrollado por Phil Zimmermann y cuya finalidad es proteger la información distribuida a través de Internet mediante el uso de criptografía de clave pública, así como facilitar la autenticación de documentos gracias a firmas digitales.
[6] MD4 es un algoritmo de resumen del mensaje (el cuarto en la serie) diseñado por el profesor Ronald Rivest del MIT. Implementa una función criptográfica de hash para el uso en comprobaciones de integridad de mensajes.
[7] Los números que aquí se indican son los valores enteros hexadecimales tal y como se introducirían en el código fuente de un programa, suponiendo que el byte menos significativo quede en la dirección de memoria más baja (*little endian*).

Se continúa con el lazo principal del algoritmo, que se repetirá para cada bloque de 512 bits del mensaje. En primer lugar, copiaremos los valores de A, B, C y D en otras cuatro variables, a, b, c y d. Luego definiremos las siguientes cuatro funciones:

$$F(X, Y,Z) = (X \wedge Y) \vee ((\neg X) \wedge Z)$$
$$G(X, Y,Z) = (X \wedge Z) \vee ((Y \wedge (\neg Z))$$
$$H(X, Y,Z) = X \oplus Y \oplus Z$$
$$I(X, Y,Z) = Y \oplus (X \vee (\neg Z))$$

Ahora representaremos por $\mathbf{m_j}$ el j-ésimo bloque de 32 bits del mensaje m (de 0 a 15), y definiremos otras cuatro funciones:

$FF(a, b, c, d, m_j, s, t_i)$ *representa* $a = b + ((a + F(b, c, d) + m_j + t_i) \lhd s)$
$GG(a, b, c, d, m_j, s, t_i)$ *representa* $a = b + ((a + G(b, c, d) + m_j + t_i) \lhd s)$
$HH(a, b, c, d, m_j, s, t_i)$ *representa* $a = b + ((a + H(b, c, d) + m_j + t_i) \lhd s)$
$II(a, b, c, d, m_j, s, t_i)$ *representa* $a = b + ((a + I(b, c, d) + m_j + t_i) \lhd s)$

donde la función $\mathbf{a \lhd s}$ representa desplazar circularmente el valor **a s** bits a la izquierda.
Las 64 operaciones que se realizan en total quedan agrupadas en cuatro rondas.[8]

Finalmente, los valores resultantes de **a, b, c y d** son sumados con **A, B, C y D**, se procesa el siguiente bloque de datos. El resultado final del algoritmo es la concatenación de A, B, C y D.

A modo de curiosidad, diremos que las constantes t_i empleadas en cada paso son la parte entera del resultado de la operación $\mathbf{2^{32} \cdot abs(sin(i))}$, estando **i** representado en radianes.

En los últimos tiempos el algoritmo MD5 ha mostrado ciertas debilidades, aunque sin implicaciones prácticas reales, por lo que se sigue considerando en la actualidad un algoritmo seguro, si bien su uso tiende a disminuir.

[8] Los valores de cada ronda se pueden encontrar en el siguiente link: https://es.wikipedia.org/wiki/MD5

Funcionamiento SHA-1 [9]y sus ataques

El algoritmo SHA-1 fue desarrollado por la NSA[10], para ser incluido en el estándar DSS (*Digital Signature Standard*). Al contrario que los algoritmos de cifrado propuestos por esta organización, SHA-1 se considera seguro y libre de puertas traseras, ya que favorece a los propios intereses de la NSA que el algoritmo sea totalmente seguro. Produce firmas de 160 bits, a partir de bloques de 512 bits del mensaje original.

El algoritmo es similar a MD5, y se inicializa igual que éste, añadiendo al final del mensaje un uno seguido de tantos ceros como sea necesario hasta completar 448 bits en el último bloque, para luego yuxtaponer la longitud en bytes del propio mensaje. A diferencia de MD5, SHA-1 emplea cinco registros de 32 bits en lugar de cuatro:

$$A = 67452301$$
$$B = EFCDAB89$$
$$C = 98BADCFE$$
$$D = 10325476$$
$$E = C3D2E1F0$$

Una vez que los cinco valores están inicializados, se copian en cinco variables, a, b, c, d y e. El lazo principal tiene cuatro rondas con 20 operaciones cada una:

$$F(X, Y, Z) = (X \wedge Y) \vee ((\neg X) \wedge Z)$$
$$G(X, Y,Z) = X \oplus Y \oplus Z$$
$$H(X, Y,Z) = (X \wedge Y) \vee (X \wedge Z) \vee (Y \wedge Z)$$

La operación **F** se emplea en la primera ronda (*t* comprendido entre 0 y 19), la **G** en la segunda (*t* entre 20 y 39) y en la cuarta (*t* entre 60 y 79), y la **H** en la tercera (*t* entre 40 y 59). Además, se emplean cuatro constantes, una para cada ronda:

$$K_0 = 5A827999$$
$$K_1 = 6ED9EBA1$$
$$K_2 = 8F1BBCDC$$
$$K_3 = CA62C1D6$$

El bloque de mensaje **m** se trocea en 16 partes de 32 bits m_0 a m_{15} y se convierte en 80 trozos de 32 bits w_0 a w_{79} usando el siguiente algoritmo:

[9] Dr. Pound, M. [Computerphile]. (2017, Abril 11). SHA: Secure Hashing Algorithm - Computerphile. [Archivo de Video] Youtube. Recuperado de https://www.youtube.com/watch?v= DMtFhACPnTY
[Gerald Hines]. (2011, Marzo 9). SHA-1 Hash Tutorial. [Archivo de Video] Youtube. Recuperado de https://www.youtube.com/watch?v= aLvwpJcOy6s

[10] La **Agencia de Seguridad Nacional** (en inglés: *National Security Agency*, también conocida como NSA) es una agencia de inteligencia del Gobierno de los Estados Unidos que se encarga de todo lo relacionado con la seguridad de la información.

$$w_t = m_t \textit{ para } t = 0 \ldots 15$$
$$w_t = (w_{t-3} \oplus w_{t-8} \oplus w_{t-14} \oplus w_{t-16}) \lhd 1 \textit{ para } t = 16 \ldots 79$$

Como curiosidad, diremos que la NSA introdujo el desplazamiento a la izquierda para corregir una posible debilidad del algoritmo, lo cual supuso modificar su nombre y cambiar el antiguo (SHA) por SHA-1.

El pseudo código de la función SHA-1 queda de la siguiente manera:

```
PARA t := 0 HASTA 79
        i := t div 20
        tmp := (a ◅ 5) + A(b, c, d) + e + wt + Ki
        e := d
        d := c
        c := b ◅ 30
        b := a
        a := tmp
```

Donde:

- **A** la función F
- **G** o **H** según el valor de *t*
    - **F** para $t \in [0, 19]$
    - **G** para $t \in [20, 39]$ y $[60, 79]$
    - **H** para $t \in [40, 59]$)

Después los valores de **a** a **e** son sumados a los registros **A** a **E** y el algoritmo continúa con el siguiente bloque de datos.

Funcionamiento SHA-2 y SHA-3

SHA-2

SHA-2 incluye un significante número de cambios respecto a su predecesor, SHA-1; y consiste en un conjunto de cuatro funciones hash de 224, 256, 384 o 512 bits.

En 2005, se identificaron fallas de seguridad en el SHA-1, permitiendo que existiera una debilidad matemática y evidenciando así la necesidad de una elaborar una función hash más fuerte. Aunque el SHA-2 se comporta de forma parecida al algoritmo SHA-1, estos ataques no han sido extendidos satisfactoriamente a SHA-2.

Las funciones hash SHA-2 están implementadas en una gran variedad de aplicaciones y protocolos de seguridad, como por ejemplo: TLS y SSL, PGP, SSH, S/MIME, Bitcoin, PPCoin y IPsec.

La moneda criptográfica Bitcoin depende en gran medida en un doble uso del SHA-256. El SHA-256 es usado para identificar los paquetes software de Debian GNU/Linux.

SHA-1 y SHA-2 son algoritmos hash de seguridad requeridos por ley en ciertas aplicaciones del gobierno de Estados Unidos, junto con el uso de otros algoritmos y protocolos criptográficos, para la protección de información clasificada y sensible

SHA-3[11]

SHA-3 es el último miembro de la familia de estándares *Secure Hash Algorithm*, publicado por NIST el 5 de agosto de 2015.1 El código fuente de la implementación de referencia se dedicó al dominio público. Aunque es parte de la misma serie de estándares, SHA-3 es internamente diferente de la estructura de los algoritmos MD5, SHA-1 y SHA-2.

SHA-3 es un subconjunto de la familia primitiva criptográfica más amplia Keccak. Los autores del Keccak han propuesto usos adicionales para la función, (todavía) no estandarizada por el NIST, incluyendo un cifrado de flujo, un cifrado autenticado sistema, un "árbol" hash esquema de hash más rápido en ciertas arquitecturas.

Keccak se basa en un enfoque novedoso llamado construcción de esponjas. La construcción de esponja se basa en una amplia función aleatoria o permutación aleatoria, y permite ingresar ("absorber" en la terminología de esponja) cualquier cantidad de datos, y generar (exprimir) cualquier cantidad de datos, mientras actúa como una función pseudoaleatoria con respecto a todas las entradas anteriores. Esto conduce a una gran flexibilidad. El propósito de SHA-3 es que puede ser sustituido directamente por SHA-2 en aplicaciones actuales si es necesario.

[11] [brainhub]. (2020, Septiembre 2). C implementation of SHA-3 and Keccak with Init/Update/Finalize API. [Archivo de Texto] Github. Recuperado de https://github.com/brainhub/SHA3IUF

Comparación de MD5 y SHA-1

- SHA-1 genera una salida con longitud de 160 bits mientas que la salida de MD5 es de 128 bits.
    - La dificultad de generar un mensaje que tenga un determinado hash esta en el orden de 2^{128} operaciones para MD5 y de 2^{160} operaciones para SHA-1.
    - La dificultad de generar dos mensajes aleatorios con el mismo hash usando MD5 esta en el orden de 2^{64} operaciones y en SHA-1 de 2^{80} operaciones.
- La pequeña diferencia de 16 bits hace a SHA-1 más seguro y resistente a ataques de fuerza bruta.
- Incluso cuando MD5 es más veloz que SHA-1, pero SHA-1 es aceptado como el estándar junto con SHA-2 y ahora SHA-3.
- La longitud del mensaje máximo que puede ser enviado con SHA-2 debe ser menor a 2^{64} bits, mientras que MD5 no tiene restricciones en longitud de mensajes.
- MD5 usa 64 constantes (una en cada paso), mientras que SHA-1 solo usa 4 constantes (una cada 20 pasos).

Aplicaciones de los algoritmos

Entre las aplicaciones d ellos algoritmos de resumen encontramos:

- Producir huellas digitales (firmas digitales) de tamaño fijo de mensajes o archivos de cualquier longitud.
- Producir información útil para detectar modificaciones maliciosas.
- Traducir las contraseñas a una representación de tamaño fijo y poder almacenarlas de forma segura.

Usando hash para obtener una Llave Privada (Fig. 114**, p. 74)**

Función hash en firma digital

Mensaje = **m** Función Resumen **= h(m)**

Firma Digital**: S = E_{Spriv}{h(m)}**

Un de los problemas que busca solucionar la criptografía es, ¿Cómo podemos verificar la identidad del emisor de un mensaje en un canal inseguro como es el internet? Las funciones hash en apoyo con el uso de llave pública y privada permiten solucionar este problema.

Creando un *resumen* del mensaje y luego cifrando ese resumen utilizando la llave privada del emisor se puede enviar la firma **S,** la cual será descifrada **D_{Spub}(S)** usando la llave pública del emisor. Luego se calcula el hash del mensaje **h(m'),** que fue enviado junto con la firma **m'** (se puede descifrar si es necesario). Si los dos valores hash son los mismos **h(m') = h(m)**, la firma digital es autentificada y el mensaje se encuentra íntegro.

$$D_{Spub}(S) = h(m)$$

$$¿h(m') = h(m)?$$

Longitud Adecuada para una Firma digital

Para decidir cuál debe ser la longitud apropiada de una firma digital, podemos usar la idea de la *paradoja del cumpleaños* [12], parafraseando podemos crear un ejemplo a base de ella: ¿Cuál es la cantidad de personas que hay que poner en una habitación para que la probabilidad de que el cumpleaños de una de ellas sea el mismo día que el mío y supere el 50%? Debemos calcular n tal que

$$n(1/365) > 0.5$$

luego **n > 182**. Sin embargo, ¿cuál sería la cantidad de gente necesaria para la probabilidad de que dos personas tengan el mismo cumpleaños y supere el 50%? Cada pareja tiene una probabilidad 1/365 de compartir el cumpleaños, y en un grupo de n personas hay n(n − 1)/2 parejas diferentes de personas, luego

$$n(n-1)/2 \cdot 1/365 > 0.5$$

Esto se cumple si **n > 19**, una cantidad sorprendentemente mucho menor que 182. La consecuencia de esta paradoja es que aunque resulte muy difícil dado **m** calcular un **m′** tal que **r(m) = r(m′)**, es mucho menos costoso buscar dos valores aleatorios **m** y **m′**, tales que **r(m) = r(m′)**.

[12] El problema del cumpleaños, también llamado paradoja del cumpleaños, establece que de un conjunto de 23 personas, hay una probabilidad del 50,7% de que al menos dos personas de ellas cumplan años el mismo día. Para 57 o más personas la probabilidad es mayor del 99,666%.

En el caso de una firma de 64 bits, necesitaríamos 264 mensajes dado un **m** para obtener el **m′**, pero bastaría con generar aproximadamente 232 mensajes aleatorios para que aparecieran dos con la misma firma digital (en general, si la primera cantidad es muy grande, la segunda cantidad es aproximadamente su raíz cuadrada). El primer ataque nos llevaría 600.000 años con una computadora que generara un millón de mensajes por segundo, mientras que el segundo necesitaría apenas una hora.

Hemos de añadir pues a nuestra lista de condiciones sobre las funciones resumen la siguiente:

- Debe ser difícil encontrar dos mensajes aleatorios, **m** y **m′** , tales que **r(m) = r(m′)**.

Hoy por hoy se recomienda emplear firmas digitales de al menos 128 bits, siendo 160 bits el valor más usado.

Funciones de Autenticación de Mensaje
Otro tipo de función resumen son las funciones de autenticación de mensajes, también conocidas como MAC (por sus siglas en inglés, *message authentication codes*). Los MAC se caracterizan fundamentalmente por el empleo de una clave secreta para poder calcular la integridad del mensaje. Puesto que dicha clave solo es conocida por el emisor y el receptor, el efecto conseguido es que el receptor puede, mediante el cálculo de dicha función, comprobar tanto la integridad como la procedencia del mensaje. Existen multitud de MAC diferentes, pero lo más común es cifrar el mensaje mediante un algoritmo simétrico en modo **CBC**, y emplear la salida correspondiente al cifrado del último bloque.

Código de autenticación de mensajes en clave-hash (HMAC)
Esta técnica usa un algoritmo hash y una clave simétrica para hacer el valor hash depender en tal llave. La forma más común de HMAC es: ***hash(llave, hash (llave, mensaje))***

La llave afecta el inicio y el final del proceso de hash.

Nombres de HMAC-hash

- MD5 HMAC-MD5
- SHA-1 HMAC-SHA (recomendado)

HMAC es usado en **IPSec** y **SSL**.

Colisión en Hash

La colisión en un algoritmo hash es de los ataques más comunes y de las razones por las que un algoritmo deja de ser seguro, es una situación que se produce cuando dos entradas distintas a una función hash produce una misma salida.

Es matemáticamente imposible que una función hash no presente colisiones, ya que la entrada a una función hash es infinita, mientras que las salidas de cualquier que nos pueden entregar una función hash esta en un conjunto finito. Sin embargo, las colisiones se producen más frecuentemente en los malos algoritmos. En una función en la cual se puede introducir datos de longitud arbitraria y que devuelve un hash de tamaño fijo (como MD5), siempre habrá colisiones, debido a que un hash dado puede pertenecer a un infinito número de entradas.

Resistencia fuerte y débil a colisiones

Para un *x* dado, si resulta computacionalmente fácil encontrar un $y \neq x$ tal que $H(x) = H(y)$, se habla de una resistencia débil a colisiones. Si resulta computacionalmente difícil encontrar un par (x, y) tal que $H(x) = H(y)$, se habla de una resistencia fuerte a colisiones.

Clasificación

- **Hashing Perfecto:** Existe una Función de Enumeración que asigna a cada valor del dominio una única posición de memoria. No posee colisiones.

- **Hashing Puro:** La función de Hash puede asignar a dos valores distintos el mismo valor hash. $x_1 \neq x_2$ y $h(x_1) = h(x_2)$ Estos dos valores reciben el nombre de sinónimos. Las estructuras de hashing puros poseen colisiones y en consecuencia se deberán establecer mecanismos para tratar los mismos. Podemos clasificarlos en estructuras cerradas y abiertas y dentro de las abiertas en estáticas y dinámicas:

    - **Cerradas**: No utilizan un nuevo espacio en memoria.
    - **Abiertas**: Utilizan espacio adicional.
        - **Estática**: La estructura principal no crece.
        - **Dinámica**: La estructura principal se expande a medida que aumenta la cantidad de elementos.

Al final del 2004, científicos chinos de la Universidad de Shandong presentaron un artículo donde se analiza las debilidades de las funciones hash, tales como MD5 y SHA-1 en la presencia de colisiones.

COMPONENTE	DESARROLLO
Actividad 1	Las funciones resumen tienen gran variedad de aplicaciones en el día a día y podemos encontrarlas en varios documentos oficiales para la autenticación y validación de ellos. Investiga 2 documentos (educativos, gobiernos, salud, etc.) que implementen el uso de una función resumen para la autenticación o validación del mismo. Anota qué tipo de documento y qué entidad lo produce, si es posible anota el algoritmo de función resumen se utiliza para los documentos. (Evaluado según la **Rúbrica de trabajos escrito, p. 137**)
Autoevaluación	1. Las funciones resumen permiten la entrada de mensajes de cualquier longitud y devuelven una respuesta con longitud fija. **a. Falso: Incorrecto** Recuerda la propiedad de las funciones hash de **Compresión,** donde se especifica como a partir de un mensaje de cualquier longitud se dará una respuesta con longitud fija. **b. Verdadero: ¡Correcto!** Según la propiedad de **Compresión** de las funciones resumen a partir de un mensaje de cualquier longitud, el resumen **h(m)** debe tener una longitud fija. Normalmente la longitud de **h(m)** es menor que el mensaje **m.** 2. Dado un resumen es posible conseguir el mensaje original si se cuenta con la llave con la que se realizó. **a. Falso: ¡Correcto!** Una de las propiedades más importantes de las funciones resumen es la **unidireccionalidad** la cual nos dice que es computacionalmente imposible encontrar el mensaje original a partir de un resumen. **b. Verdadero: ¡Incorrecto!** Recuerda que la propiedad de **unidireccionalidad** la cual es la más importante de una función resumen. 3. Las funciones resumen no son confiables para validar que un documento fue o no modificado. **a. Falso: ¡Correcto!** La propiedad de **Difusión** nos permite validar que si algún bit del mensaje original fue modificado el resumen no será el mismo. **b. Verdadero: Incorrecto** Recuerda que según la propiedad de **difusión** las funciones hash permiten validar si hubo cualquier modificación en el mensaje original.

4. SHA al ser desarrollado por el gobierno de E.E.U.U. se considera inseguro y con varias puertas traseras.
    a. **Falso: ¡Correcto!**
    SHA fue desarrollado para el uso de NSA y por lo tanto se considera seguro, si el gobierno lo usa no quiere que tenga puertas traseras.
    b. **Verdadero: Incorrecto**
    Recuerda que SHA fue desarrollado por la NSA y por lo tanto se considera seguro.
5. MD5 es más veloz que SHA-1 y por lo tanto es más seguro.
    a. **Falso: ¡Correcto!**
    SHA puede ser más lento, pero por la diferencia de 16 bits entre los algoritmos hace que SHA sea más seguro.
    b. **Verdadero: Incorrecto**
    Recuerda que la velocidad de un algoritmo no determina su seguridad. SHA puede ser más lento, pero por la diferencia de 16 bits entre los algoritmos hace que SHA sea más seguro.
6. Las funciones resumen se usan para almacenar contraseñas en bases de datos de manera segura.
    a. **Falso: Incorrecto**
    Una de las aplicaciones principales de las funciones hash es almacenar de forma segura las contraseñas en bases de datos.
    b. **Verdadero: ¡Correcto!**
    Las funciones resumen son usadas para almacenar de forma segura contraseñas en las bases de datos. Comparando el Hash para autenticar a un usuario en lugar de con la contraseña en texto plano.
7. Es posible tener hashes repetidos al usar una función resumen.
    a. **Falso: Incorrecto**
    Las funciones resumen buscan que no existan colisiones, pero, al no tener limitantes en las entradas y por lo tanto son entradas infinitas para salidas finitas es posible que existan hashes repetidos.
    b. **Verdadero: ¡Correcto!**
    Aunque se busca que no existan colisiones, al no tener limitantes en las entradas y por lo tanto son entradas infinitas para salidas finitas es posible que existan hashes repetidos.

5. Asignatura para la cual fue diseñado el material.

Criptografía, 9no semestre, Ingeniería en Computación.

6. Resultados esperados.

La creación de una plataforma en línea y material didáctico ayudará a que los alumnos tengan mayores recursos para el aprendizaje de la asignatura de Criptografía, con una integración más práctica de los temas vistos, donde podrán reforzar los conocimientos adquiridos, no solo siendo evaluados de forma tradicional, sino buscando diferentes enfoques de aprendizaje. De igual manera, se busca facilitar la carga de evaluación por parte de los profesores, dando lugar a un ambiente de estudio basado en preguntas concretas. Con ayuda del material creado se espera aportar los medios que facilitarán la instrucción de la asignatura y brindar apoyo a docentes con los nuevos métodos de enseñanza, creando recursos que podrán ser consultados por estudiantes. Además, se espera que el material pueda ser utilizado por usuarios de otras asignaturas afines (Seguridad informática, Redes de datos, Arquitectura cliente-servidor, etc.) así como fuera de la Facultad, logrando así expandir el conocimiento de la materia y ampliar su alcance a otros sectores profesionales.

Se espera que el material será un primer paso para las asignaturas de Criptografía, Seguridad Informática y cualquier asignatura afín al área de Redes y Seguridad dentro del plan de estudios de Ingeniería en Computación 2016 de la Facultad de Ingeniería o de otras Facultades para extender las formas de aprendizaje para los estudiantes.

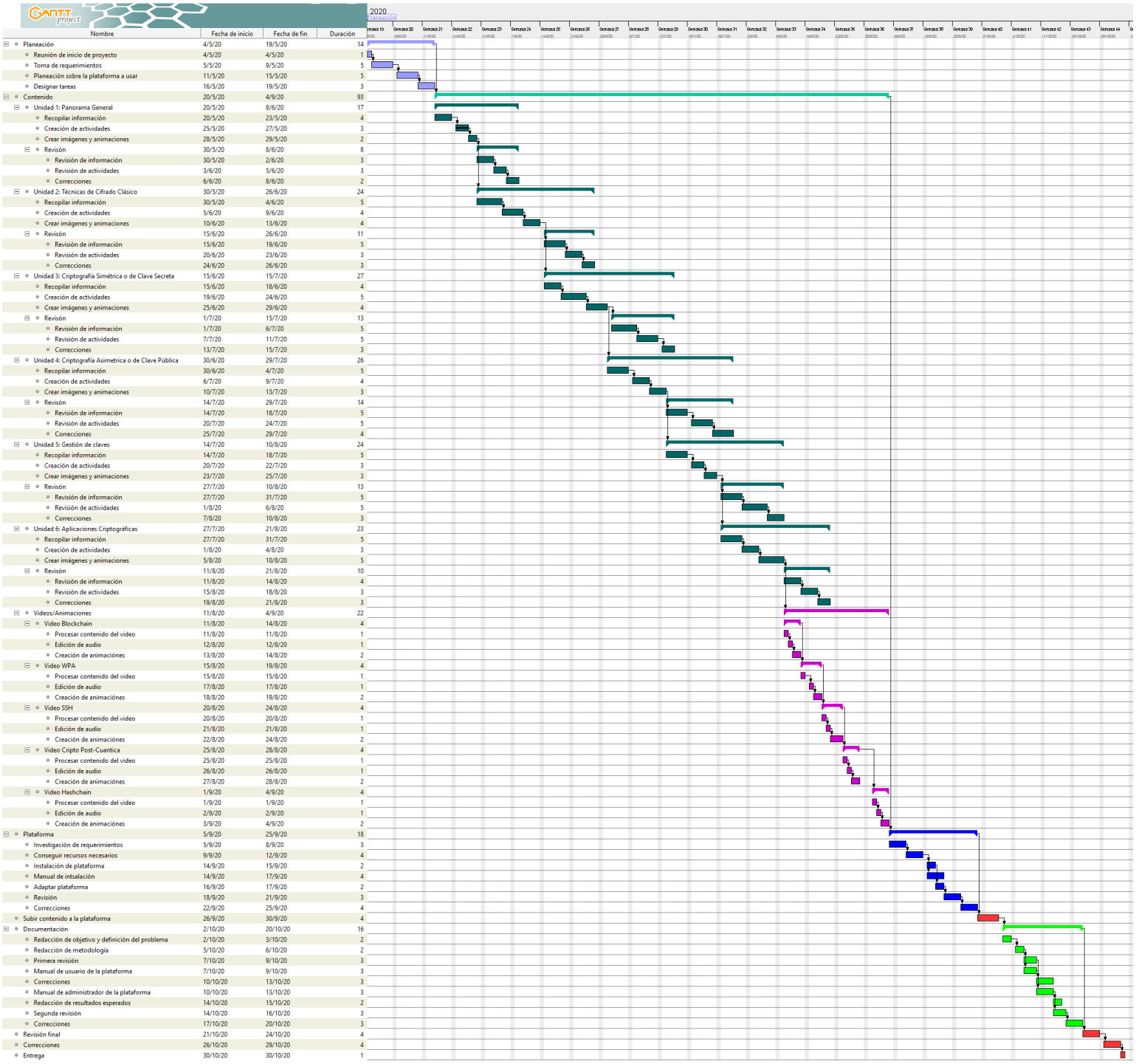
Todo el material desarrollado tiene base en el temario de la carrera de Ingeniería en Computación que imparte la Facultad de Ingeniería de la UNAM, acreditando así la selección de los temas. Al igual se vio reflejado el apoyo de expertos en el área de criptografía y fue avalado por un consultor pedagógico proporcionado por CUAIEED, los cuales dieron fe del contenido con bases teóricas y sustanciales para el uso en la enseñanza y aprendizaje.

Con la puesta en marcha de la plataforma propuesta se estará beneficiando a alrededor de 70 alumnos por semestre de la carrera de Ingeniería en Computación de la Facultad de Ingeniería, que son los que cursan la asignatura. Además, se impacta a las asignaturas relacionadas al área de Redes y Seguridad dentro y fuera de la Facultad de Ingeniería a las carreras afines.

Dar a conocer las opciones de enseñanza en línea podrá dar paso a que otras materias tengan interés en construir versiones remotas, material didáctico e información de dominio público para expandir el conocimiento y apoyar a alumnos junto con docentes en las materias impartidas. Este trabajo espera presentar un apoyo a los nuevos métodos de enseñanza, mostrar las herramientas con las que se cuentan y así facilitar la transición de educación tradicional a nuevas modalidades de enseñanza.

7. Cronograma de actividades y diagrama de Gantt.

A continuación se muestra el seguimiento de tiempos al realizar el proyecto, esto utilizando un diagrama de Gantt con fechas de inicio y final de los entregables.



Nombre	Fecha de inicio	Fecha de fin	Duración
Planeación	4/5/20	19/5/20	14
Reunión de inicio de proyecto	4/5/20	4/5/20	1
Toma de requerimientos	5/5/20	9/5/20	5
Planeación sobre la plataforma a usar	11/5/20	15/5/20	5
Designar tareas	16/5/20	19/5/20	3
Contenido	20/5/20	4/9/20	93
Unidad 1: Panorama General	20/5/20	8/6/20	17
Recopilar información	20/5/20	23/5/20	4
Creación de actividades	25/5/20	27/5/20	3
Crear imágenes y animaciones	28/5/20	29/5/20	2
Revisón	30/5/20	8/6/20	8
Revisión de información	30/5/20	2/6/20	3
Revisión de actividades	3/6/20	5/6/20	3
Correcciones	6/6/20	8/6/20	2
Unidad 2: Técnicas de Cifrado Clásico	30/5/20	26/6/20	24
Recopilar información	30/5/20	4/6/20	5
Creación de actividades	5/6/20	9/6/20	4
Crear imágenes y animaciones	10/6/20	13/6/20	4
Revisón	15/6/20	26/6/20	11
Revisión de información	15/6/20	19/6/20	5
Revisión de actividades	20/6/20	23/6/20	3
Correcciones	24/6/20	26/6/20	3
Unidad 3: Criptografía Simétrica o de Clave Secreta	15/6/20	15/7/20	27
Recopilar información	15/6/20	18/6/20	4
Creación de actividades	19/6/20	24/6/20	5
Crear imágenes y animaciones	25/6/20	29/6/20	4
Revisón	1/7/20	15/7/20	13
Revisión de información	1/7/20	6/7/20	5
Revisión de actividades	7/7/20	11/7/20	5
Correcciones	13/7/20	15/7/20	3
Unidad 4: Criptografía Asimetrica o de Clave Pública	30/6/20	29/7/20	26
Recopilar información	30/6/20	4/7/20	5
Creación de actividades	6/7/20	9/7/20	4
Crear imágenes y animaciones	10/7/20	13/7/20	3
Revisón	14/7/20	29/7/20	14
Revisión de información	14/7/20	18/7/20	5
Revisión de actividades	20/7/20	24/7/20	5
Correcciones	25/7/20	29/7/20	4
Unidad 5: Gestión de claves	14/7/20	10/8/20	24
Recopilar información	14/7/20	18/7/20	5
Creación de actividades	20/7/20	22/7/20	3
Crear imágenes y animaciones	23/7/20	25/7/20	3
Revisón	27/7/20	10/8/20	13
Revisión de información	27/7/20	31/7/20	5
Revisión de actividades	1/8/20	6/8/20	5
Correcciones	7/8/20	10/8/20	3
Unidad 6: Aplicaciones Criptográficas	27/7/20	21/8/20	23
Recopilar información	27/7/20	31/7/20	5
Creación de actividades	1/8/20	4/8/20	3
Crear imágenes y animaciones	5/8/20	10/8/20	5
Revisón	11/8/20	21/8/20	10
Revisión de información	11/8/20	14/8/20	4
Revisión de actividades	15/8/20	18/8/20	3
Correcciones	19/8/20	21/8/20	3
Videos/Animaciones	11/8/20	4/9/20	22
Video Blockchain	11/8/20	14/8/20	4
Procesar contenido del video	11/8/20	11/8/20	1
Edición de audio	12/8/20	12/8/20	1
Creación de animaciónes	13/8/20	14/8/20	2
Video WPA	15/8/20	19/8/20	4
Procesar contenido del video	15/8/20	15/8/20	1
Edición de audio	17/8/20	17/8/20	1
Creación de animaciónes	18/8/20	19/8/20	2
Video SSH	20/8/20	24/8/20	4
Procesar contenido del video	20/8/20	20/8/20	1
Edición de audio	21/8/20	21/8/20	1
Creación de animaciónes	22/8/20	24/8/20	2
Video Cripto Post-Cuantica	25/8/20	28/8/20	4
Procesar contenido del video	25/8/20	25/8/20	1
Edición de audio	26/8/20	26/8/20	1
Creación de animaciónes	27/8/20	28/8/20	2
Video Hashchain	1/9/20	4/9/20	4
Procesar contenido del video	1/9/20	1/9/20	1
Edición de audio	2/9/20	2/9/20	1
Creación de animaciónes	3/9/20	4/9/20	2
Plataforma	5/9/20	25/9/20	18
Investigación de requerimientos	5/9/20	8/9/20	3
Conseguir recursos necesarios	9/9/20	12/9/20	4
Instalación de plataforma	14/9/20	15/9/20	2
Manual de intsalación	14/9/20	17/9/20	4
Adaptar plataforma	16/9/20	17/9/20	2
Revisión	18/9/20	21/9/20	3
Correcciones	22/9/20	25/9/20	4
Subir contenido a la plataforma	26/9/20	30/9/20	4
Documentación	2/10/20	20/10/20	16
Redacción de objetivo y definición del problema	2/10/20	3/10/20	2
Redacción de metodología	5/10/20	6/10/20	2
Primera revisión	7/10/20	9/10/20	3
Manual de usuario de la plataforma	7/10/20	9/10/20	3
Correcciones	10/10/20	13/10/20	3
Manual de administrador de la plataforma	10/10/20	13/10/20	3
Redacción de resultados esperados	14/10/20	15/10/20	2
Segunda revisión	14/10/20	16/10/20	3
Correcciones	17/10/20	20/10/20	3
Revisión final	21/10/20	24/10/20	4
Correcciones	26/10/20	29/10/20	4
Entrega	30/10/20	30/10/20	1

Curso Criptografía

Tarea

Nombre	Fecha de inicio	Fecha de fin	Duración
Planeación	4/5/20	19/5/20	14
Reunión de inicio de proyecto	4/5/20	4/5/20	1
Toma de requerimientos	5/5/20	9/5/20	5
Planeación sobre la plataforma a usar	11/5/20	15/5/20	5
Designar tareas	16/5/20	19/5/20	3
Contenido	20/5/20	4/9/20	93
Unidad 1: Panorama General	20/5/20	8/6/20	17
Recopilar información	20/5/20	23/5/20	4
Creación de actividades	25/5/20	27/5/20	3
Crear imágenes y animaciones	28/5/20	29/5/20	2
Revisón	30/5/20	8/6/20	8
Revisión de información	30/5/20	2/6/20	3
Revisión de actividades	3/6/20	5/6/20	3
Correcciones	6/6/20	8/6/20	2
Unidad 2: Técnicas de Cifrado Clásico	30/5/20	26/6/20	24
Recopilar información	30/5/20	4/6/20	5
Creación de actividades	5/6/20	9/6/20	4
Crear imágenes y animaciones	10/6/20	13/6/20	4
Revisón	15/6/20	26/6/20	11
Revisión de información	15/6/20	19/6/20	5
Revisión de actividades	20/6/20	23/6/20	3
Correcciones	24/6/20	26/6/20	3
Unidad 3: Criptografía Simétrica o de Clave Secreta	15/6/20	15/7/20	27
Recopilar información	15/6/20	18/6/20	4
Creación de actividades	19/6/20	24/6/20	5
Crear imágenes y animaciones	25/6/20	29/6/20	4
Revisón	1/7/20	15/7/20	13
Revisión de información	1/7/20	6/7/20	5

Curso Criptografía

Tarea

Nombre	Fecha de inicio	Fecha de fin	Duración
Revisión de actividades	7/7/20	11/7/20	5
Correcciones	13/7/20	15/7/20	3
Unidad 4: Criptografía Asimetrica o de Clave Pública	30/6/20	29/7/20	26
Recopilar información	30/6/20	4/7/20	5
Creación de actividades	6/7/20	9/7/20	4
Crear imágenes y animaciones	10/7/20	13/7/20	3
Revisón	14/7/20	29/7/20	14
Revisión de información	14/7/20	18/7/20	5
Revisión de actividades	20/7/20	24/7/20	5
Correcciones	25/7/20	29/7/20	4
Unidad 5: Gestión de claves	14/7/20	10/8/20	24
Recopilar información	14/7/20	18/7/20	5
Creación de actividades	20/7/20	22/7/20	3
Crear imágenes y animaciones	23/7/20	25/7/20	3
Revisón	27/7/20	10/8/20	13
Revisión de información	27/7/20	31/7/20	5
Revisión de actividades	1/8/20	6/8/20	5
Correcciones	7/8/20	10/8/20	3
Unidad 6: Aplicaciones Criptográficas	27/7/20	21/8/20	23
Recopilar información	27/7/20	31/7/20	5
Creación de actividades	1/8/20	4/8/20	3
Crear imágenes y animaciones	5/8/20	10/8/20	5
Revisón	11/8/20	21/8/20	10
Revisión de información	11/8/20	14/8/20	4
Revisión de actividades	15/8/20	18/8/20	3
Correcciones	19/8/20	21/8/20	3
Videos/Animaciones	11/8/20	4/9/20	22
Video Blockchain	11/8/20	14/8/20	4
Procesar contenido del video	11/8/20	11/8/20	1

Curso Criptografía

Tarea

Nombre	Fecha de inicio	Fecha de fin	Duración
Edición de audio	12/8/20	12/8/20	1
Creación de animaciónes	13/8/20	14/8/20	2
Video WPA	15/8/20	19/8/20	4
Procesar contenido del video	15/8/20	15/8/20	1
Edición de audio	17/8/20	17/8/20	1
Creación de animaciónes	18/8/20	19/8/20	2
Video SSH	20/8/20	24/8/20	4
Procesar contenido del video	20/8/20	20/8/20	1
Edición de audio	21/8/20	21/8/20	1
Creación de animaciónes	22/8/20	24/8/20	2
Video Cripto Post-Cuantica	25/8/20	28/8/20	4
Procesar contenido del video	25/8/20	25/8/20	1
Edición de audio	26/8/20	26/8/20	1
Creación de animaciónes	27/8/20	28/8/20	2
Video Hashchain	1/9/20	4/9/20	4
Procesar contenido del video	1/9/20	1/9/20	1
Edición de audio	2/9/20	2/9/20	1
Creación de animaciónes	3/9/20	4/9/20	2
Plataforma	5/9/20	25/9/20	18
Investigación de requerimientos	5/9/20	8/9/20	3
Conseguir recursos necesarios	9/9/20	12/9/20	4
Instalación de plataforma	14/9/20	15/9/20	2
Manual de intsalación	14/9/20	17/9/20	4
Adaptar plataforma	16/9/20	17/9/20	2
Revisión	18/9/20	21/9/20	3
Correcciones	22/9/20	25/9/20	4
Subir contenido a la plataforma	26/9/20	30/9/20	4

Curso Criptografía

Tarea

Nombre	Fecha de inicio	Fecha de fin	Duración
Documentación	2/10/20	20/10/20	16
Redacción de objetivo y definición del problema	2/10/20	3/10/20	2
Redacción de metodología	5/10/20	6/10/20	2
Primera revisión	7/10/20	9/10/20	3
Manual de usuario de la plataforma	7/10/20	9/10/20	3
Correcciones	10/10/20	13/10/20	3
Manual de administrador de la plataforma	10/10/20	13/10/20	3
Redacción de resultados esperados	14/10/20	15/10/20	2
Segunda revisión	14/10/20	16/10/20	3
Correcciones	17/10/20	20/10/20	3
Revisión final	21/10/20	24/10/20	4
Correcciones	26/10/20	29/10/20	4
Entrega	30/10/20	30/10/20	1

8. Anexo

En esta sección se describe la instalación de la plataforma Moodle y la evaluación de los trabajos escritos.

8.1. Manual de instalación de Moodle

La realización del proyecto fue en un servidor basado en Ubuntu Server versión 20.04.1 LTS (GNU/Linux 5.4.0-26-generic x86_64), el manual está enfocado en dicha distribución.

La plataforma que se instalo es Moodle, la cual es una herramienta de gestión de aprendizaje (LMS por sus siglas en inglés *Learning Management System*), este de distribución libre y escrita en PHP, haciéndola de fácil acceso a diferentes dispositivos a través de un navegador web. Moodle fue creada con la idea de ayudar a docentes a crear comunidades de aprendizaje en línea, sus principales usos en la actualidad es la educación a distancia y diversos proyectos de e-learning en escuelas, universidades, oficinas y otros sectores académicos y educativos.

8.1.1. Prerrequisitos de para la instalación

En la siguiente sección se describirá todos aquellos prerrequisitos para una correcta instalación de la plataforma Moodle en el servidor.

Durante la instalación de Moodle se requiere de varios nombres de usuarios y contraseñas, los cuales serán necesarios tener a la mano, entre ellos el usuario con **acceso root** de Ubuntu y posteriormente los usuarios de MySQL y *admin* de Moodle.

Primero se verifica y realiza una actualización de software en Ubuntu

```
sudo apt-get update
```

Es necesario instalar un editor de textos dentro del Shell de Linux de Ubuntu, el cual se usará para configurar archivos. El editor usado es VIM (editor *heavyweight*), el cual se instala mediante el siguiente comando:

```
sudo apt-get install vim
```

Comandos básicos para VIM:

- Para comenzar a editar el archivo de texto presionar la tecla **i**
- Una vez se haya terminado de editar el archivo se presiona la tecla **Esc**
    - Para salvar el nuevo archivo escribir **:w** seguido de presionar la tecla **Enter**
    - Para salvar y salir de un archivo **:x** seguido de presionar la tecla **Enter**
    - Para salir del editor **:q** seguido de presionar la tecla **Enter**

- Igual es posible escribir **:wq** para escribir y salir de un archivo seguido de la tecla **Enter**
- Para forzar la salida del editor sin guardar **:q!** seguido de presionar la tecla **Enter**

8.1.2. Instalar Apache

Para poder utilizar la maquina con Ubuntu como servidor HTTP es necesario instalar un paquete de software, en este caso se usará Apache Web Server. Con Apache podemos enviar páginas web desde nuestra dirección IP a través de internet a personas que la soliciten.

Para instalar el paquete de Apache en Ubuntu se utilizará el siguiente comando:

```
sudo apt-get install apache2
```

Al verificar la instalación podemos ingresar a la dirección IP de la maquina donde se instaló a través de un navegador web, al ingresar a dicha dirección debemos ser presentados con la página web por defecto de Apache con el titulo de "*Apache2 Ubuntu Default Page*"

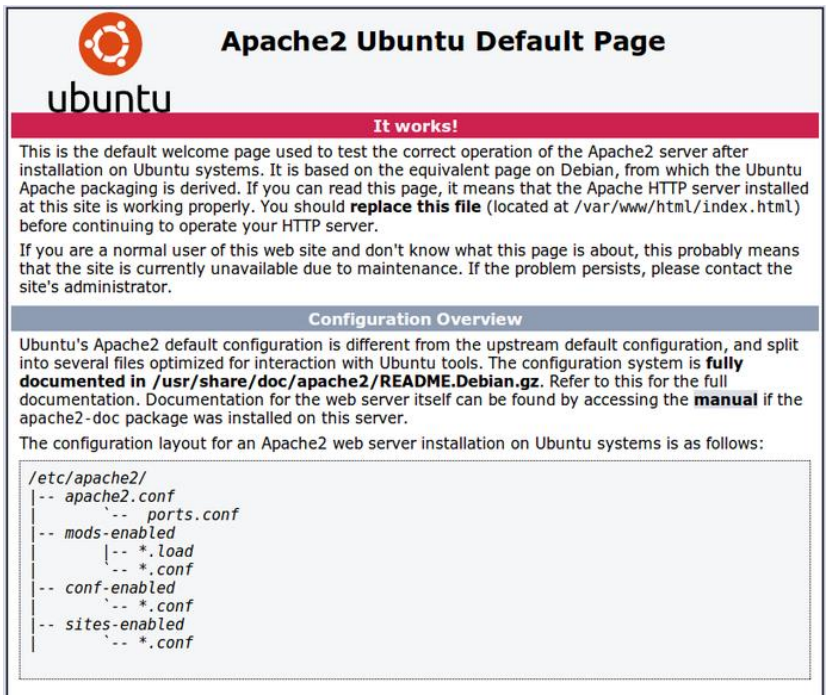


Fig. 139 Captura de pantalla de página por defecto al instalar Apache en Ubuntu

Una vez instalado apache debemos configurar la *Firewall*, en este caso usaremos la *Firewall* por defecto de Ubuntu UFW[13], la cual nos permitirá el acceso de trafico por internet mediante los puertos 80 (HTTP) y 443 (HTTPS). Mostramos las aplicaciones que nos permite configurar UFW

[13] *Uncomplicated Firewall*, una aplicación de software que bloquea el trafico de red (Usualmente por seguridad).

```
sudo ufw show app list
```

El shell nos debe responder con una lista de perfiles de aplicaciones como el siguiente ejemplo:

```
Apache

Apache Full

Apache Secure

OpenSSH
```

Usamos el siguiente comando que nos abrirá el trafico web en los puertos 80 y 443, al igual permitimos el acceso a SSH:

```
sudo ufw allow ‘Apache Full’

sudo ufw allow ‘OpenSSH’
```

Por último, verificamos el estado del *Firewall*, donde podemos comprobar que el tráfico esta permitido*:*

```
sudo ufw status
```

Es necesario tener a la mano los comandos básicos para control de Apache:

Detener Apache

```
sudo systemctl stop apache2.service
```

Iniciar Apache

```
sudo systemctl start apache2.service
```

Reiniciar Apache

```
sudo systemctl restart apache2.service
```

Recargar Apache

```
sudo systemctl reload apache2.service
```

Para la terminar la configuración de Apache instalaremos los certificados de SSL utilizando **Let's Encrypt** (https://letsencrypt.org/) una Autoridad Certificadora sin fines de lucro que provee certificados TLS a servicios web, debido a que nuestra plataforma no tendrá mayor uso al institucional esto será suficiente para mantener las conexiones seguras mediante el acceso en navegadores web (El estándar prohíbe crear certificado SSL a direcciones IP, es necesario tener un nombre de dominio).

Comenzamos con el archivo de configuración para el Servidor de Apache:

```
sudo vim /etc/apache2/sites-available/<nombre página o IP>.conf
```

Copiamos y pegamos lo siguiente:

```
<VirtualHost *:80>

     ServerAdmin admin@<nombre página o IP>

     ServerName <nombre página o IP>

     ServerAlias <nombre página o IP>

     DocumentRoot /var/www/html


     ErrorLog ${APACHE_LOG_DIR}/<nombre página o IP>-error.log

     CustomLog ${APACHE_LOG_DIR}/<nombre página o IP>-access.log combined

</VirtualHost>
```

Ahora el archivo de configuración de nuestro servidor ya se creó, corremos los siguientes comandos para activarlo:

```
sudo a2ensite <nombre página o IP>.conf

systemctl reload apache2
```

El servidor debe de estar corriendo sin ningún problema en este punto.

La creación de un certificado utilizando *Let's Encrypt* utiliza una aplicación llamada **Certbot**, el cual es una herramienta que puede conseguir y renovar de manera automática el certificado SSL de *Let's Encrypt*. Para instalarlo usamos el comando:

```
sudo apt install certbot
```

Después de instalar *Certbot*, creamos un archivo para *Let's Encrypt* para el plugin de *Webroot*, para validar nuestro dominio en el directorio $ {webroot-path}/.Well-known/acme-challenge

Para eso, creamos los directorios y le damos acceso a Apache de la siguiente manera:

```
sudo mkdir -p /var/lib/letsencrypt/.well-known

sudo chgrp www-data /var/lib/letsencrypt

sudo chmod g+s /var/lib/letsencrypt
```

A continuación, creamos el archivo *well-known* con las configuraciones que se escriben:

```
sudo vim /etc/apache2/conf-available/well-known.conf
```

En el archivo copiamos lo siguiente:

```
<Directory "/var/lib/letsencrypt/">

    AllowOverride None

    Options MultiViews Indexes SymLinksIfOwnerMatch IncludesNoExec

    Require method GET POST OPTIONS

</Directory>
```

En este punto el servidor está apuntando al dominio que le dimos al IP, pero apache lo carga en HTTP y, con *Certbot* instalado y la configuración necesaria, se procede a obtener el certificado.

Antes de pedir el certificado es necesario activar Apache SSL, los Headers, HTTPS/2 y la configuración del archivo *well-known* el cual se creó en los pasos anteriores, para eso se ejecutan los siguientes comandos:

```
sudo a2enmod ssl

sudo a2enmod headers
```

```
sudo a2enmod http2

sudo a2enconf well-known
```

Después de los comandos anterior es necesario reiniciar el servidor de Apache:

```
systemctl restart apache2
```

Con toda la configuración lista ahora es posible pedir el certificado, para eso se corre el siguiente comando:

```
sudo certbot certonly --agree-tos --email admin@example.com --webroot -w /var/lib/lets
encrypt/ -d <nombre del dominio>
```

En este punto *Let's Encrypt* deberá verificar la validez de tu dominio y el servidor, seguido de instalar el certificado del dominio, una vez que termine deberá salir un mensaje con los datos de donde se guardo los archivos de certificación (El certificado del sitio y su llave privada).

Con el certificado creado, es necesario añadirlo a las configuraciones de Apache y el dominio. Primero generamos una llave de intercambio Diffie-Hellman (DH) para poder compartir las llaves de cifrado de forma segura. Para esto se ejecuta el siguiente comando que generara un certificado de 2048 bits:

```
sudo openssl dhparam -out /etc/ssl/certs/dhparam.pem 2048
```

Abrimos el archivo que creamos de <*nombre página o IP*>.*conf* en /sites-avilable/ y cambiamos la configuración para aceptar las llamadas al puerto 443, para que se pueda acceder al servidor por medio de HTTPS e igual redirigiremos todos los llamados de HTTP a HTTPS:

```
<VirtualHost *:80>
  ServerName <nombre del dominio>
  Redirect permanent / https://<nombre del dominio>/
</VirtualHost>

<VirtualHost *:443>
  ServerName <nombre del dominio>
```

```
  DocumentRoot /var/www/html/

  Protocols h2 http:/1.1
  <If "%{HTTP_HOST} == ‘<nombre del dominio>’">
    Redirect permanent / https://<nombre del dominio>/
  </If>
  ErrorLog ${APACHE_LOG_DIR}/<nombre del dominio>-error.log
  CustomLog ${APACHE_LOG_DIR}/<nombre del dominio>-access.log combined

  SSLEngine On
  SSLCertificateFile /etc/letsencrypt/live/<nombre del dominio>/fullchain.pem
  SSLCertificateKeyFile /etc/letsencrypt/live/<nombre del dominio>/privkey.pem
  SSLOpenSSLConfCmd DHParameters "/etc/ssl/certs/dhparam.pem"

  SSLCipherSuite EECDH+AESGCM:EDH+AESGCM:AES256+EECDH:AES256+EDH
  SSLProtocol All -SSLv2 -SSLv3 -TLSv1 -TLSv1.1
  SSLCompression off
  SSLUseStapling on
</VirtualHost>
```

A continuación, se deberá configurar una caché de servidor para la información del Protocolo de Comprobación del Estado de un Certificado En línea u *Online Certificate Status* (OCSP), el cual, como su nombre lo dice, es un método para determinar el estado de vigencia de un certificado digital. El mejor lugar para esto sería el archivo de configuración de Apache SSL.

```
sudo vim /etc/apache2/mods-available/ssl.conf
```

Este archivo contiene todas las opciones que utiliza Apache para SSL. Se debe agregar una opción adicional *SSLStaplingCache* a este archivo como se muestra a continuación.

```
# Establecer la ubicación de la caché de grapado SSL OCSP
   SSLStaplingCache shmcb:/tmp/stapling_cache(128000)
```

La directiva *SSLStaplingCache* define la ubicación de la caché y un valor de tamaño para la caché OCSP. Se guardan los cambios anteriores y reinicie Apache2 para que la configuración anterior tenga efecto.

```
systemctl restart apache2
```

La certificación de *Let's Encrypt* tiene un tiempo límite de un año, por lo que cada año se debe renovar. Esto se puede hacer de manera manual o automático, para realizarlo de forma automática se siguen los siguientes pasos:

```
sudo crontab -e

0 1 * * * /usr/bin/certbot renew & > /dev/null

sudo certbot renew --dry-run
```

El servidor ya se encuentra listo con Apache, Firewall y el certificado HTTPS.

8.1.3. Instalar MySQL y PHP

Para poder continuar con la instalación de Moodle se requieren dos dependencias, MySQL que será usado para la base de datos de Moodle y PHP en su versión más actual (en este caso la 7.4), dentro de la terminal instalamos con el siguiente comando:

```
sudo apt-get install mysql-client mysql-server php7.4 libapache2-mod-php7.4
```

Continuamos con las dependencias de PHP y software adicional que requiere Moodle para funcionar:

```
sudo apt install graphviz aspell ghostscript clamav php7.4-pspell php7.4-curl php7.4-g
d php7.4-intl php7.4-mysql php7.4-xml php7.4-xmlrpc php7.4-ldap php7.4-zip php7.4-soap
php7.4-mbstring
```

Reiniciamos Apache para que los módulos se carguen de forma correcta:

```
systemctl restart apache2
```

Instalamos **Git** el cual será utilizado para descargar la aplicación principal de Moodle e igual las actualizaciones:

```
sudo apt-get install git
```

Para configurar MySQL corremos el comando de `mysql_secure_installation`, en donde nos saldrá a la configuración para la creación de un super usuario para la base de datos **GUARDAR EL NOMBRE DE USUARIO Y CONTRASEÑA** que se utilice, ya que será usado más adelante:

```
sudo systemctl start mysql

sudo systemct enable mysql

sudo mysql_secure_installation
```

Una vez creado el usuario de MySQL es necesario hacer algunos cambios a la configuración por defecto con la que se instalo, entre ellas debemos modificar el motor de almacenamiento a **innodb** y modificar las tablas a formato innodb_file_per_table:

```
sudo vim /etc/mysql/mysql.conf.d/mysqld.cnf
```

Se debe buscar la sección de [mysqld] debajo de las Preferencias Básicas (*Basic Settings*) y se agregarán las siguientes líneas después de la ultima declaración.

```
default_storage_engine = innodb

innodb_file_per_table = 1

innodb_file_format = Barracuda
```

Nota: si se usa la última versión de MariaDB en Ubuntu 20.04 estos cambios a la configuración causaran un error, si esto sucede no es necesario añadir las líneas del paso anterior, ya que la configuración de MariaDB ya está modificada por defecto con esos parámetros.

Reiniciamos MySQL para que los cambios tomen efecto:

```
sudo service mysql restart
```

Una vez configurado MySQL, debemos crear una base de datos para Moodle y un usuario que tendrá acceso a dicha base de datos. Usando la contraseña del super usuario que creamos en la instalación de MySQL procedemos a acceder a MySQL:

```
sudo mysql -u root -p
```

mysql>

```
CREATE DATABASE moodle DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;
```

En los siguientes pasos donde aparezca “moodledude" y "passwordformoodledude” se debe sustituir por un nombre de usuario y una contraseña respectivamente.

mysql>

```
create user 'moodledude'@'localhost' IDENTIFIED BY 'passwordformoodledude';
```

mysql>

```
GRANT SELECT,INSERT,UPDATE,DELETE,CREATE,CREATE TEMPORARY TABLES,DROP,INDEX,ALTER ON m
oodle.* TO 'moodledude'@'localhost';
```

mysql>

```
quit;
```

8.1.4. Instalar Moodle

Antes de poder instalar Moodle debemos descargar la versión más reciente, para lo cual vamos a realizar una descarga usando **Git** (un controlador de versiones), para la descarga utilizaremos el directorio */opt :*

```
cd /opt

sudo git clone git://git.moodle.org/moodle.git
```

Entramos al directorio de **Moodle** donde mostraremos la ramas activas del proyecto de Moodle:

```
cd moodle

sudo git branch -a
```

Nos moveremos a la rama de la última versión más estable de la aplicación (en este caso es la 3.9) y cambiando a esta rama tenemos en nuestra maquina la última versión ya descargada:

```
sudo git branch --track MOODLE_39_STABLE origin/MOODLE_39_STABLE

sudo git checkout MOODLE_39_STABLE
```

Copiamos los archivos de Moodle a la raíz de nuestro servidor Apache, /var/www/html/ y creamos un directorio donde guardaremos todos los archivos de la aplicación, este directorio debe encontrarse fuera del directorio de Apache por seguridad. Por último, se dan los permisos correspondientes a cada directorio (Moodle y Moodledata).

```
sudo cp -R /opt/moodle /var/www/html/

sudo mkdir /var/moodledata

sudo chown -R www-data /var/moodledata

sudo chmod -R 777 /var/moodledata

sudo chmod -R 0755 /var/www/html/moodle
```

Una vez creados los directorios y dados los permisos correspondientes, se pueden seguir dos alternativas, la configuración de Moodle mediante terminal o la configuración mediante la interfaz gráfica entrando al servidor, si se decide utilizar la interfaz gráfica es necesario cambiar los permisos del directorio de Moodle en `/var/www/html/moodle` a:

```
sudo chmod -R 777 /var/www/html/moodle
```

Una vez terminada la configuración por medio de la interfaz SE DEBE REGRESAR a los permisos anteriores:

```
sudo chmod -R 0755 /var/www/html/moodle
```

Para terminar la instalación de Moodle ingresamos en nuestro navegador web a la dirección de IP o dominio de nuestro servidor https://<IP o nombre de servidor>/moodle y seguimos los pasos que se muestran.

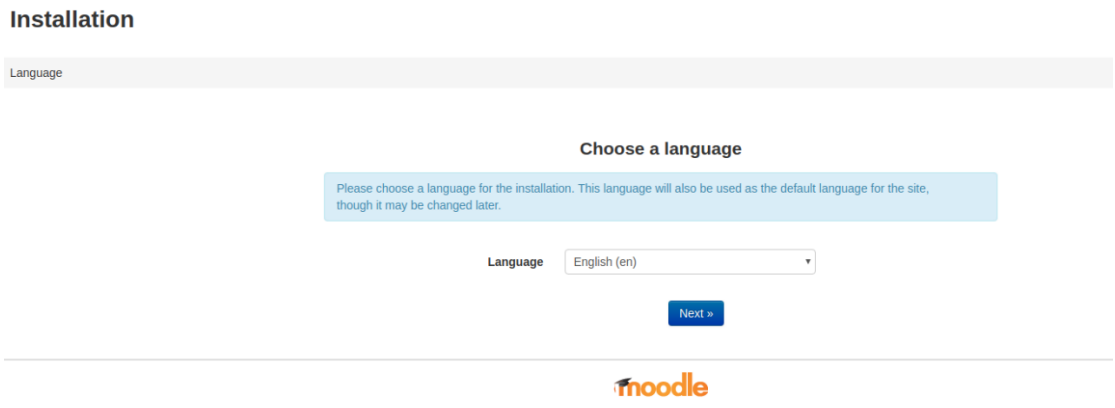


Fig. 140 Página de inicio de instalación de Moodle mediante interfaz grafica.

Cambiamos la ubicación del directorio de datos de moodle a /var/moodledata.
Seleccionamos MySQL como la base de datos.
En la configuración de la base de datos ponemos los siguientes datos:

- *Host server*: localhost
- *Database*: moodle
- *User*: moodledude (Nombre de usuario que definimos para la base de datos)
- *Password*: passwordformoodledude (Contraseña que definimos para la base de datos)
- *Table Prefix*: mdl_

La siguiente pestaña mostrará el entorno de instalación y si todos los requerimientos se cumplen para correr Moodle, si no es así, saldrán en error las dependencias que faltan instalar. Si todo esta correcto se puede continuar.

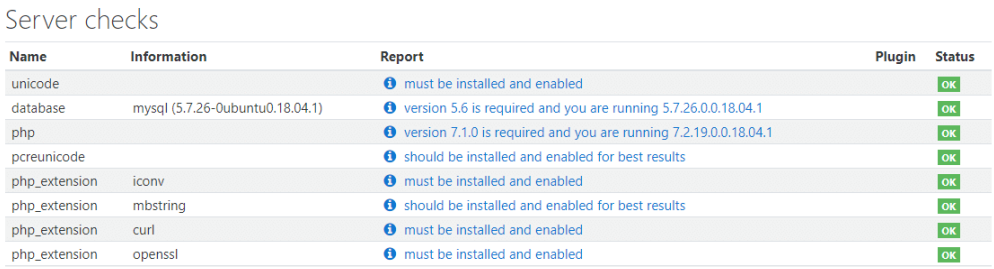


Fig. 141 Sección de requerimientos de Moodle

Por último, se necesita crear un super usuario (*admin*) para la aplicación de Moodle, este usuario controlará toda la aplicación y es necesario tener buenas prácticas al asignarle una contraseña.

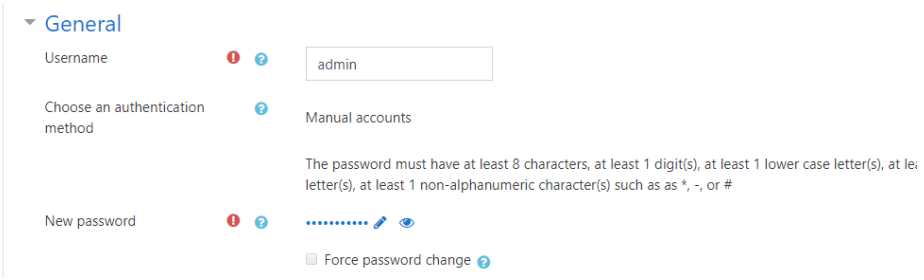


Fig. 142 Creación de Super usuario Moodle

Ahora Moodle se encuentra instalado de forma correcta en el servidor y debe ser posible entrar a la aplicación sin problema. NO OLVIDAR REGRESAR LOS PERMISOS DEL DIRECTORIO `/var/www/html/moodle.`

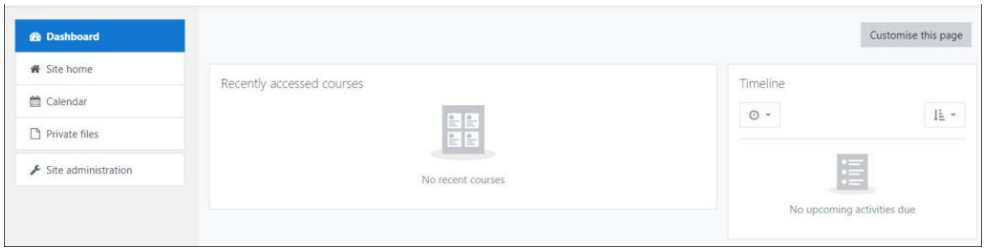


Fig. 143 Página principal de Moodle después de instalación

8.2. Código de estilo para la plataforma

El siguiente código de formato .scss tiene la función de dar una apariencia específica a la plataforma Moodle para corresponder a los colores y diseño de la Facultad de Ingeniería de la UNAM.

```
/* stylelint-disable color-hex-length */
/**
 * Default preset file.
 *
 * Use this space to customise variables, create mixins and override SCSS.
 */

// Colors
$red-fi:                    #cd171e;
$grey-fi:                   #d8d8d8;
$grey-dark-fi:              #333333;




$gray-dark:                 #373a3c !default;
$gray:                      #55595c !default;
$gray-light:                #818a91 !default;
$gray-lighter:              #eceeef !default;
$gray-lightest:             #f7f7f9 !default;

$brand-primary:             #2F7DA8 !default;
$brand-success:             #77b300 !default;
$brand-warning:             #ff8800 !default;
$brand-danger:              #ff4136 !default;
$brand-inverse:             $gray-dark !default;
$navbar-height:             50px !default;

// Spacing
$spacer:   1rem !default;
$spacer-x: $spacer !default;
$spacer-y: $spacer !default;
$border-width: 1px !default;

// Body
$body-bg:    #f4f4f4 !default;
$body-color: #3a3a3a !default;

// Typography
$font-family-sans-serif: "Open Sans", "Helvetica Neue", Arial, sans-serif !default;
$font-size-root: 14px !default;

// Tables
$table-bg:                      transparent !default;
$table-bg-accent:               #fafafa !default;
$table-bg-hover:                lighten($brand-primary, 55%) !default;
```

```
$table-bg-active:                $table-bg-hover !default;
$table-border-color:             #f4f4f4 !default;

// Dropdowns
$dropdown-border-color:           #e2e2e2 !default;
$dropdown-link-color:             rgba(0, 0, 0, .535) !default;

// Navbar
$navbar-light-color:                 rgba(0, 0, 0, .535) !default;
$navbar-light-hover-color:           $brand-primary !default;
$navbar-light-active-color:          $brand-primary !default;

// Cards
$card-border-radius:       0 !default;
$card-border-color:        rgba(238, 238, 238, 1) !default;

// Breadcrumbs
$breadcrumb-bg:                  transparent !default;
$breadcrumb-padding-x:           0 !default;
$breadcrumb-divider: "/" !default;
$breadcrumb-divider-rtl: "/" !default;

/************************** Import section  ****************************/
@import "fontawesome";
@import "bootstrap";
@import "moodle";

/**
 * Navigation bar identity.
 */
.navbar-light {
    background-color: $grey-fi !important;
    color: $red-fi;
    border-bottom: solid 2px $red-fi;
    min-height: 60px;

    .container {
        padding-left: 0;
        padding-right: 0;
    }

    .navbar-brand {
        padding-left: 1rem;
        padding-right: 1rem;
        margin-right: $spacer * 2;
        color: #fff;
        background-color: $brand-primary;
        @include hover-focus {
            color: #fff;
        }
        &.has-logo {
            color: $brand-primary;
            background-color: transparent;
```

```
            @include hover-focus {
                color: $brand-primary;
            }
        }
    }

    .navbar-nav {
        .nav-item + .nav-item {
            margin-left: $spacer * 2;
        }
        .nav-link {
            font-size: $font-size-sm;
            text-transform: uppercase;
            letter-spacing: 1px;
        }
    }
}




/**
 * Styling the dropdown menus.
 */
.dropdown-menu {
    font-size: 14px;
    border-radius: 0;
    .dropdown-item {
        padding-top: 8px;
        padding-bottom: 8px;
        border-bottom: $border-width solid $dropdown-border-color;
        &:last-child {
            border-bottom: 0;
        }
    }
    .dropdown-divider {
        display: none;
    }
}

/**
 * User picture.
 */
.userpicture {
    border-radius: 50%;
}

/**
 * For background in content areas.
 */
#page.container-fluid {
    padding: 0 (2 * $spacer);
```

```
}

#block-region-side-pre {
    padding-left: 0;
}

/**
 * Dashboard styling.
 */
#page-my-index {
    #region-main {
        background-color: transparent;
        border: 0;
        padding: 0;
    }
}







/**
 * Blocks.
 */
.block-region .card-block {
    .card-title {
        padding-bottom: ($spacer/2);
        font-size: 1.143rem;
        font-weight: 600;
        text-transform: uppercase;
    }
}

/**
 * Navigation.
 */
.block_navigation,
.block_settings {
    .block_tree .tree_item {
        margin: ($spacer/2) 0;
    }
}

/**
 * Form styles.
 */

.form-group {
    margin-top: ($spacer/2);
}

.form-inline .form-group {
```

```
    margin-top: 0;
}

/* Change footer bar characteristics */

.bg-inverse {

  background-color: #d1c8c1 !important;

  color: #fff !important;

}

/* Fix footer positioning problem - Mary */

#page-wrapper::after,
#page-wrapper,
#page-footer {

  min-height: 0;

}

#page-wrapper {
  margin-bottom: 0;
}

#page-footer {
  // text-align: center;
    background-color: $grey-dark-fi !important;
}

/* Remove the Home link in the footer */

.homelink {
    display: none;
}
```

8.3. Rúbrica de trabajo escrito

La siguiente rúbrica será usada como referencia para los profesores al momento de evaluar los trabajos de los estudiantes que serán parte de las actividades de las UAPAs.

Criterios	**Escala de calificación**					
Presentación Excelente formato que contiene los elementos visuales y organizativos descritos en las plantillas dadas.	2.5 Excelente	2.25 Muy bien	2 Bien	1.75 Satisfactorio	1.5 Necesita mejorar	0 No presentó
Contenido y estructura El contenido y la estructura siguen los descritos en la plantilla, incluida una solución clara al problema dado.	2.5 Excelente	2.25 Muy bien	2 Bien	1.75 Satisfactorio	1.5 Necesita mejorar	0 No presentó
Escritura Describe de forma clara y concisa los objetivos del trabajo.	2.5 Excelente	2.25 Muy bien	2 Bien	1.75 Satisfactorio	1.5 Necesita mejorar	0 No presentó
Claridad Discutir de forma clara y concisa las conclusiones de la etapa correspondiente del proyecto.	2.5 Excelente	2.25 Muy bien	2 Bien	1.75 Satisfactorio	1.5 Necesita mejorar	0 No presentó

9. Glosario

C

cifrados: Que está escrito con letras, símbolos o números que sólo pueden comprenderse si se dispone de la llave necesaria para descifrarlos., 61, 62, 63, 64, 67, 76, 86, 90, 91

Criptografía: La criptografía es el estudio de técnicas matemáticas relacionadas con el aspectos de la seguridad de la información. Busca mantener las comunicaciones seguras en presencia de adversarios., 4

D

didáctico: Que sirve, es adecuado o está pensado para la enseñanza., 4, 8, 11, 13, 113

H

HASH, 100

HTML: Lenguaje de marcado de hipertexto, y le permite al usuario crear y estructurar secciones, párrafos, encabezados, enlaces y elementos de cita en bloque para páginas web y aplicaciones, 14, 15

L

ludificación: El uso de técnicas, elementos y dinámicas propias de los juegos con el fin de potenciar la motivación, así como de reforzar la conducta para activar el aprendizaje, entre otras cosas., 5

M

Moodle: Herramienta de gestión de aprendizaje (LMS), de distribución libre, escrita en PHP., 9, 10, 11, 13, 14, 15, 18, 22, 23, 25, 38, 40, 119, 127, 128, 129, 130, 131, 138

O

OSI: Open Systems Interconnection, 59

P

Plug-in: Componente de software que agrega una característica específica a un programa de computadora existente., 18

S

SGA: Sistemas de Gestión de Aprendizaje, 9, 10

10. Referencias

S.A. (2020, Noviembre 6), Moodle Docs 3.10 [Web Blog] Moodle. Recuperado de https://docs.moodle.org/310/en/Main_page

Lucena-López, M. J. (2001), Criptografía y Seguridad en Computadores. pp. 121-129, 157-164, España.

Menezes, A. J. (2001), et al. Handbook of Applied Cryptography. pp. 321-385 CRC.

Dr. Pound, M. [Computerphile]. (2019, Septiembre 22). AES Explained (Advanced Encryption Standard) - Computerphile. [Archivo de Video] YouTube. Recuperado de https://www.youtube.com/watch?v=O4xNJsjtN6E

[brainhub]. (2020, Septiembre 2). C implementation of SHA-3 and Keccak with Init/Update/Finalize API. [Archivo de Texto] Github. Recuperado de https://github.com/brainhub/SHA3IUF

Scott, T. [Computerphile]. (2013, Noviembre 8). Hashing Algorithms and Security - Computerphile. [Archivo de Video] YouTube. Recuperado de https://www.youtube.com/watch?v= b4b8ktEV4Bg

Dr. Pound, M. [Computerphile]. (2017, Abril 11). SHA: Secure Hashing Algorithm - Computerphile. [Archivo de Video] YouTube. Recuperado de https://www.youtube.com/watch?v= DMtFhACPnTY

[Gerald Hines]. (2011, Marzo 9). SHA-1 Hash Tutorial. [Archivo de Video] YouTube. Recuperado de https://www.youtube.com/watch?v= aLvwpJcOy6s

Pretty Good Privacy. (25 de Agosto de 2020). En Wikipedia. https://es.wikipedia.org/w/index.php?title=Pretty_Good_Privacy&oldid=128744096

MD5. (9 de Noviembre de 2019). En Wikipedia. https://es.wikipedia.org/w/index.php?title=MD5&oldid=121189223

Paradoja del cumpleaños. (25 de Agosto de 2020). En Wikipedia. https://es.wikipedia.org/w/index.php?title=Paradoja_del_cumplea%C3%B1os&oldid=127562247