**Documento de requerimientos de software**

***[MOKTEC Campus Online]***

***Fecha: [27/10/2024]***



**Tabla de contenido**

Historial de Versiones 3

Información del Proyecto 3

Aprobaciones 3

1. Propósito 4

2. Alcance del producto / Software 4

3. Referencias 4

4. Funcionalidades del producto 5

5. Clases y características de usuarios 5

6. Entorno operativo 5

7. Requerimientos funcionales 6

9.1. (Nombre de la funcionalidad 1) 6

9.2. (Nombre de la funcionalidad 2) 7

9.3. (Nombre de la funcionalidad N) 7

8. Reglas de negocio 8

9. Requerimientos de interfaces externas 9

9.1. Interfaces de usuario 9

9.2. Interfaces de hardware 9

9.3. Interfaces de software 9

9.4. Interfaces de comunicación 9

10. Requerimientos no funcionales 10

11. Otros requerimientos 11

12. Glosario 12

**Historial de Versiones**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Fecha** | **Versión** | **Autor** | **Organización** | **Descripción** |
| 27/10/2024 | V1.0 | Ruben Hadechine | MOKTEC | Definición de propósito y alcance de producto. |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Información del Proyecto**

|  |  |
| --- | --- |
| Empresa / Organización | MOKTEC |
| Proyecto | MOKTEC Campus Online |
| Fecha de preparación | 28/10/2024 |
| Cliente | MOKTEC |
| Patrocinador principal | MOKTEC |
| Gerente / Líder de Proyecto | Ing. Cristian Rodelo Gonzalez |
| Gerente / Líder de Análisis de negocio y requerimientos | Ing. Omar David Fuentes Bustillo |

**Aprobaciones**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre y Apellido** | **Cargo** | **Departamento u Organización** | **Fecha** | **Firma** |
| Ing. Cristian Rodelo Gonzalez | Gerente | Gerencia | 28/10/2024 |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

1. **Propósito**

El sitio web de MOKTEC tiene como propósito facilitar la gestión académica y administrativa, permitiendo una experiencia fluida tanto para estudiantes como profesores. Además de servir como plataforma de promoción para atraer nuevos clientes, proporcionar un sistema eficiente para el registro, seguimiento y consulta de información académica, optimizando procesos internos de la institución.

1. **Alcance del producto / Software**

Este sistema abarcará las siguientes funcionalidades:

* Promoción y captación de estudiantes: Landing page con información sobre programas académicos, cursos disponibles y beneficios.
* Registro y gestión académica: Inscripción de estudiantes en materias, alta y baja de asignaturas.
* Consulta de información: Acceso de estudiantes y profesores mediante inicio de sesión para revisar notas, calificaciones parciales y finales.
* Gestión de profesores y materias: Registro y actualización de datos de docentes y asignación de materias.
* Cálculo automático de notas: Sistema para promediar calificaciones de los estudiantes en cada materia.

1. **Referencias**
2. **Funcionalidades del producto**

**1. Promoción Institucional:**

- Landing page con información sobre programas académicos, cursos y beneficios de MOKTEC.

**2. Registro de Usuarios:**

- Inscripción de estudiantes con datos personales y selección de materias.

- Registro de profesores y asignación de materias.

**3. Gestión Académica:**

- Alta y baja de materias para los estudiantes.

- Consulta de horarios y disponibilidad de materias.

**4. Gestión de Notas:**

- Registro y cálculo de notas parciales y finales por asignatura.

- Generación de reportes de calificaciones para estudiantes y profesores.

**5. Acceso Personalizado:**

- Inicio de sesión diferenciado por roles: estudiantes, profesores y administradores.

- Panel de control personalizado según el tipo de usuario.

**6. Seguridad de la Información:**

- Protección de datos mediante cifrado SSL y control de acceso por roles.

**7. Interfaz Amigable:**

- Diseño responsive para dispositivos móviles y escritorio.

- Navegación intuitiva para facilitar el uso del sistema.

1. **Clases y características de usuarios**

**1. Estudiantes:**

Los estudiantes son usuarios frecuentes del sistema que acceden para realizar procesos académicos y consultar información relacionada con sus materias.

* Frecuencia de uso: Alta.
* Nivel de experiencia: Básico.
* Privilegios de seguridad: Acceso limitado a su información personal y académica.

**- Funcionalidades relevantes (Sección 4):**

1. Inscripción en el sistema y selección de materias.
2. Consulta de notas parciales y finales.
3. Matrícula en asignaturas.
4. Visualización y selección de horarios disponibles.

**2. Profesores:**

Los profesores utilizan el sistema para gestionar las calificaciones de los estudiantes y organizar su carga horaria.

* Frecuencia de uso: Moderada.
* Nivel de experiencia: Básico a intermedio.
* Privilegios de seguridad: Acceso a información relacionada con las materias y los estudiantes asignados.

**- Funcionalidades relevantes (Sección 4):**

1. Registro y asignación de calificaciones a estudiantes.
2. Consulta de horarios y materias asignadas.

**3. Administrador:**

El administrador es responsable de la gestión general del sistema y su mantenimiento operativo.

* Frecuencia de uso: Moderada a alta.
* Nivel de experiencia: Avanzado.
* Privilegios de seguridad: Acceso total para gestionar usuarios, horarios y asignaturas.

**- Funcionalidades relevantes (Sección 4):**

1. Registro de estudiantes y profesores.
2. Asignación de salones y horarios a profesores.
3. Administración de las materias disponibles.

1. **Entorno operativo**

El sistema está diseñado para operar como un sitio web accesible desde cualquier navegador moderno. A continuación, se detallan los aspectos clave de su entorno operativo:

**Plataforma de hardware:**

* Dispositivos cliente: Computadoras, tablets y smartphones con acceso a internet.

**Sistema operativo:**

* Cliente: Compatible con sistemas operativos como Windows, macOS, Android e iOS.

**Navegadores soportados:**

* Google Chrome (versión 90 o superior).
* Mozilla Firefox (versión 85 o superior).
* Microsoft Edge (versión 90 o superior).
* Safari (versión 14 o superior).

**Otros componentes:**

* Base de datos relacional para la gestión de información académica y de usuarios.
* Servidor web, como Apache o Nginx, para alojar el sitio.
* Lenguajes y frameworks: HTML, CSS, JavaScript y framework Flask para el backend.

Este entorno asegura la compatibilidad, accesibilidad y rendimiento necesario para un sistema académico en línea.

1. **Requerimientos funcionales**

**Funcionalidad 1: Login de usuarios**

**Descripción:** Permitir a los usuarios (estudiantes, profesores y administradores) acceder al sistema mediante credenciales únicas.

**Prioridad:** Alta

**Acciones iniciadoras y comportamiento esperado:**

1. El usuario accede a la pantalla de inicio de sesión.
2. Ingresa su correo electrónico y contraseña.
3. El sistema verifica las credenciales y redirige al panel correspondiente según su rol.

**Requerimientos funcionales:**

* **REQ-1:** Mostrar campos obligatorios para correo electrónico y contraseña.
* **REQ-2:** Validar las credenciales ingresadas contra la base de datos.
* **REQ-3:** Redirigir al panel de control correspondiente según el rol del usuario (estudiante, profesor, administrador).
* **REQ-4:** Mostrar mensajes de error en caso de credenciales incorrectas.

**Funcionalidad 2: Emisión de formulario de inscripción**

**Descripción:** Permitir a los estudiantes completar un formulario en línea con su información básica y materias de interés.

**Prioridad:** Alta

**Acciones iniciadoras y comportamiento esperado:**

1. El estudiante accede al formulario desde la landing page.
2. Completa los datos requeridos y selecciona las materias de interés.
3. Envía el formulario, y el sistema lo guarda para revisión administrativa.

**Requerimientos funcionales:**

* **REQ-1:** Validar campos obligatorios antes de permitir el envío.
* **REQ-2:** Guardar los datos del formulario en la base de datos para su revisión.
* **REQ-3**: Mostrar mensaje de confirmación tras el envío exitoso.

**Funcionalidad 3: Registro de estudiantes**

**Descripción:** Permitir a los administradores registrar estudiantes con base en formularios enviados y asignarles materias.

**Prioridad:** Alta

**Acciones iniciadoras y comportamiento esperado:**

1. El administrador revisa los formularios enviados por los estudiantes.
2. Completa o corrige la información y asigna materias.
3. Confirma el registro.

**Requerimientos funcionales:**

* + **REQ-1:** Mostrar una lista de formularios enviados para revisión.
  + **REQ-2:** Validar campos obligatorios antes de confirmar el registro.
  + **REQ-3:** Guardar la información en la base de datos y notificar al estudiante por correo.

**Funcionalidad 4: Inscripción en materias**

**Descripción:** Permitir a los estudiantes seleccionados inscribirse en materias adicionales desde su panel de usuario.

**Prioridad:** Alta

**Acciones iniciadoras y comportamiento esperado:**

1. El estudiante accede al módulo de inscripción.
2. Selecciona materias disponibles y confirma la inscripción.

**Requerimientos funcionales:**

* + **REQ-1:** Mostrar una lista de materias disponibles con horarios.
  + **REQ-2:** Validar cupos disponibles antes de la inscripción.
  + **REQ-3:** Restringir inscripción a materias no cursadas previamente.

**Funcionalidad 5: Consulta de horarios y notas**

**Descripción:** Permitir a los estudiantes consultar horarios y calificaciones, incluyendo histórico de notas.

**Prioridad:** Media

**Acciones iniciadoras y comportamiento esperado:**

1. El estudiante selecciona el período académico o consulta su historial completo.
2. El sistema muestra las notas y horarios correspondientes.

**Requerimientos funcionales:**

* + **REQ-1:** Permitir la selección de períodos académicos o historial completo.
  + **REQ-2:** Mostrar notas y horarios de manera clara y organizada.
  + **REQ-3:** Validar que el usuario tenga acceso solo a su información.

**Funcionalidad 6: Registro de profesores y asignación de materias**

**Descripción:** Permitir a los administradores registrar nuevos profesores y asignarles materias y horarios.

**Prioridad:** Alta

**Acciones iniciadoras y comportamiento esperado:**

1. El administrador registra al profesor y asigna materias y horarios desde un formulario.
2. Confirma la operación y el sistema guarda la información.

**Requerimientos funcionales:**

* + **REQ-1:** Validar los campos obligatorios del formulario.
  + **REQ-2:** Asignar horarios mediante un calendario interactivo.
  + **REQ-3:** Guardar la información en la base de datos.

**Funcionalidad 7: Asignación de calificaciones**

**Descripción:** Permitir a los profesores registrar calificaciones de los estudiantes.

**Prioridad:** Alta

**Acciones iniciadoras y comportamiento esperado:**

1. El profesor selecciona la materia y al estudiante.
2. Ingresa las calificaciones parciales o finales y las guarda.

**Requerimientos funcionales:**

* + **REQ-1:** Mostrar lista de estudiantes inscritos por materia.
  + **REQ-2:** Validar que las calificaciones estén dentro del rango permitido.
  + **REQ-3:** Guardar las calificaciones en la base de datos.

**Funcionalidad 8: Registro de fallas y cálculo de totales**

**Descripción:** Permitir a los profesores registrar fallas por estudiante y calcular el total acumulado.

**Prioridad:** Media

**Acciones iniciadoras y comportamiento esperado:**

1. El profesor accede al módulo de "Registro de Fallas".
2. Selecciona una materia y al estudiante correspondiente.
3. Registra las fallas de una sesión específica.
4. El sistema calcula automáticamente el total acumulado de fallas por materia.

**Requerimientos funcionales:**

* + **REQ-1:** Permitir al profesor registrar fallas por sesión.
  + **REQ-2:** Mostrar el total acumulado de fallas por estudiante.
  + **REQ-3:** Validar que el total acumulado no exceda el límite permitido antes de alertar al profesor.

**Funcionalidad 9: Cálculo de calificación definitiva**

**Descripción:** Calcular automáticamente la calificación definitiva de los estudiantes en cada materia con base en las calificaciones parciales ingresadas.

**Prioridad:** Media

**Acciones iniciadoras y comportamiento esperado:**

1. El sistema calcula la calificación final al registrar todas las notas parciales de un estudiante en una materia.
2. Muestra la calificación definitiva al profesor y estudiante.

**Requerimientos funcionales:**

* **REQ-1:** Calcular la definitiva como promedio ponderado de las notas parciales.
* **REQ-2:** Mostrar la calificación definitiva una vez completado el registro.

**Funcionalidad 10: Modificación de información personal desde el perfil de usuario**

**Descripción:** Permitir a los usuarios (estudiantes, profesores y administradores) modificar su información personal desde su perfil de usuario.

**Prioridad:** Media

**Acciones iniciadoras y comportamiento esperado:**

1. El usuario accede a su perfil desde el menú de usuario.
2. Selecciona la opción para editar su información personal.
3. Modifica los campos que desee (nombre, correo, teléfono, etc.).
4. Confirma los cambios y el sistema actualiza la información en la base de datos.

**Requerimientos funcionales:**

* **REQ-1:** Mostrar un formulario con los datos personales actuales del usuario.
* **REQ-2:** Validar que los campos modificados sean correctos y completos (ej. correo electrónico válido).
* **REQ-3:** Permitir solo la modificación de información personal asociada al usuario autenticado.
* **REQ-4:** Actualizar la base de datos con los cambios realizados.
* **REQ-5:** Mostrar un mensaje de éxito tras la actualización de los datos.

1. **Reglas de negocio**

**1. Acceso restringido por rol**

* Solo los **administradores** pueden realizar registros de nuevos estudiantes y profesores, así como asignar materias y horarios.
* Los **profesores** solo pueden ver y modificar las calificaciones de los estudiantes inscritos en sus materias.
* Los **estudiantes** solo pueden ver y modificar su propia información personal, así como consultar sus calificaciones y horarios.

**2. Validación de inscripción en materias**

* Un estudiante no puede inscribirse en una materia en la que ya esté registrado.
* Un estudiante no puede inscribirse en una materia si no hay cupos disponibles.

**3. Modificación de información personal**

* Los usuarios pueden modificar su información personal solo desde su perfil de usuario.
* Un usuario no podrá cambiar su rol (estudiante, profesor, administrador) desde su perfil.

**4. Calificación definitiva**

* La calificación definitiva de un estudiante debe ser calculada como el promedio ponderado de las calificaciones parciales, de acuerdo con las políticas de la institución.
* La calificación definitiva no podrá ser modificada una vez que haya sido calculada y almacenada.

**5. Registro de fallas**

* Los profesores solo pueden registrar fallas para los estudiantes inscritos en sus materias.
* El sistema debe calcular automáticamente el total de fallas acumuladas de cada estudiante, y alertar si el número de faltas excede el límite permitido para aprobar la materia.

**6. Autenticación y autorización**

* Los usuarios deben ser autenticados (con correo y contraseña) antes de acceder al sistema.
* Los usuarios solo pueden acceder a las funcionalidades que correspondan a su rol (ej. los administradores tienen acceso completo, los estudiantes solo a su información personal y calificaciones, etc.).

**7. Historial de notas**

* El historial de notas de los estudiantes debe mantenerse disponible en todo momento para su consulta.
* Los estudiantes solo pueden ver su propio historial de notas, no el de otros estudiantes.

**8. Restricción de materias según semestre**

* + Los estudiantes solo pueden inscribirse en las materias correspondientes a su semestre académico.
  + Los administradores son responsables de asignar las materias y horarios adecuados a cada semestre.

1. **Requerimientos de interfaces externas**

**9.1 Interfaces de usuario**

**Características generales de la interfaz:**

**- Diseño de interfaz gráfica (GUI):** El sistema utilizará un diseño limpio y moderno, con una interfaz intuitiva que facilita la navegación. Las páginas estarán organizadas en bloques bien definidos para facilitar la visualización de la información. Los colores deben ser consistentes y alineados con la identidad visual de MOKTEC.

**- Pantallas principales del sistema:**

**- Pantalla de inicio de sesión:** Campos para ingresar correo electrónico y contraseña. Botón de inicio de sesión y enlace para recuperar contraseña.

**- Panel de usuario:** Dependiendo del rol del usuario (estudiante, profesor, administrador), el panel de usuario mostrará acceso directo a las funcionalidades correspondientes (por ejemplo, inscripción de materias, consulta de calificaciones, registro de fallas, etc.).

**- Formulario de inscripción de estudiantes:** Se presentarán campos para los datos personales y selección de materias, con validaciones visuales para los campos obligatorios.

**- Panel administrativo:** Para gestionar usuarios (estudiantes, profesores) y asignación de materias y horarios.

**Estándares de interfaz gráfica (GUI):**

**- Botones:** Los botones deben ser de forma rectangular con esquinas redondeadas y color destacado para acciones primarias (Ej. verde para "Enviar", rojo para "Eliminar"). Los botones secundarios deben ser más neutrales.

**- Fuente:** Se utilizará una fuente sans-serif como Arial o Helvetica para mejorar la legibilidad.

**- Consistencia:** Los elementos clave deben mantenerse consistentes a lo largo de todas las pantallas. Por ejemplo, los botones de "Guardar" y "Cancelar" siempre estarán en la parte inferior de los formularios.

**- Accesibilidad:** Debe cumplirse con las pautas de accesibilidad, como texto alternativo para imágenes, navegación fácil con teclado y buen contraste de colores para personas con dificultades visuales.

**9.2 Interfaces de hardware**

**Dispositivos soportados:**

**- Computadores de escritorio y portátiles:** El sistema será accesible en navegadores web en dispositivos de escritorio y portátiles con sistemas operativos Windows, macOS y Linux.

**- Dispositivos móviles (tabletas y smartphones):** La interfaz será responsive, adaptándose a pantallas de diferentes tamaños (con especial énfasis en smartphones Android e iOS).

**- Impresoras:** El sistema podrá generar informes de calificaciones, registros de inscripciones y otros documentos académicos en formato PDF, que podrán ser impresos en cualquier impresora compatible con el sistema operativo del dispositivo.

**Protocolos de comunicación soportados:**

**- Protocolo HTTP/HTTPS:** El sistema se comunicará a través de estos protocolos para acceder a las funcionalidades del sitio web y garantizar que toda la información esté encriptada mediante HTTPS.

**- Protocolo IMAP/SMTP:** Utilizado para enviar y recibir correos electrónicos de notificación a usuarios.

**9.3 Interfaces de software**

**Interacciones con otros componentes:**

**- Base de datos:** El sistema interactuará con una base de datos SQL (por ejemplo, MySQL o PostgreSQL) para almacenar y recuperar información sobre usuarios, calificaciones, materias, horarios, etc.

**- \*\*Requerimientos de base de datos:** El sistema debe poder realizar operaciones CRUD (Crear, Leer, Actualizar, Eliminar) de manera eficiente.

**- Sistema operativo:** El sistema será accesible en servidores que operan bajo Linux o Windows Server. El backend deberá estar diseñado en un lenguaje como PHP, Node.js o Python, con un framework adecuado como Laravel, Express o Django.

**- Herramientas y librerías:**

**- Frontend:** Utilizará HTML5, CSS3, JavaScript (con React o Vue.js) para las interfaces de usuario interactivas.

**- Backend:** El backend podrá ser implementado con Node.js, Express.js y bases de datos SQL.

**9.4 Interfaces de comunicación**

**Requerimientos de comunicación:**

**- Correo electrónico:** El sistema utilizará servicios de correo electrónico para notificar a los estudiantes y profesores sobre inscripciones, cambios en las materias, calificaciones, etc.

**- Formato de correos electrónicos:** Los correos enviados desde el sistema deben contener un formato estructurado y profesional, incluyendo la información relevante de forma clara y organizada.

**- Protocolos de comunicación de red:**

**- HTTP/HTTPS:** Todos los datos entre el cliente y el servidor se comunicarán mediante los protocolos HTTP y HTTPS para garantizar la seguridad de la información transmitida.

**- Requerimientos de encriptación y seguridad en las comunicaciones:**

**- SSL/TLS para HTTPS:** Todo el tráfico web estará cifrado con SSL/TLS para proteger los datos sensibles, como credenciales de inicio de sesión, información personal, y calificaciones.

**- Encriptación de contraseñas:** Las contraseñas de los usuarios se almacenarán encriptadas en la base de datos mediante algoritmos de hashing seguros, como bcrypt.

1. **Requerimientos no funcionales**

**1. Rendimiento del sistema**

- El sistema debe soportar hasta 200 usuarios concurrentes sin afectar significativamente los tiempos de respuesta.

- Las operaciones críticas (ej. consulta de notas, inscripciones) deben ejecutarse en menos de 2 segundos bajo carga máxima.

**2. Disponibilidad**

- El sistema debe estar disponible 24/7, con un tiempo de inactividad máximo de 2 horas al mes para mantenimiento.

- Implementar monitoreo para garantizar la disponibilidad y alertar sobre fallos.

**3. Escalabilidad**

- El sistema debe ser escalable para manejar un crecimiento en el número de usuarios (hasta 1000 usuarios concurrentes) y datos sin requerir reestructuración significativa.

- El backend debe permitir la integración con servicios de nube en el futuro.

**4. Seguridad**

- Las contraseñas deben almacenarse utilizando un algoritmo de hashing seguro (ej. bcrypt).

- El sistema debe implementar autenticación de dos factores (2FA) para roles administrativos.

- Todo el tráfico debe estar cifrado mediante HTTPS.

- El sistema debe prevenir ataques comunes como inyecciones SQL, XSS y CSRF.

**5. Compatibilidad**

- Compatible con los navegadores más utilizados: Google Chrome, Firefox, Safari y Microsoft Edge (últimas dos versiones).

- Soporte para pantallas de diferentes tamaños, desde dispositivos móviles hasta escritorios.

**6. Mantenimiento y actualizaciones**

- Las actualizaciones de software deben implementarse sin interrumpir el uso del sistema, excepto en ventanas de mantenimiento programadas.

- El código fuente debe estar documentado para facilitar el mantenimiento y futuras expansiones.

**7. Usabilidad**

- El sistema debe ser intuitivo, permitiendo que usuarios con conocimientos básicos en tecnología puedan navegar sin necesidad de un manual extenso.

- Tiempo máximo para completar tareas críticas (ej. inscripción a materias): 3 minutos.

**8. Portabilidad**

- El sistema debe ser portable para ejecutarse en servidores Linux y Windows sin cambios significativos en la configuración.

**9. Capacidad de almacenamiento**

- La base de datos debe poder almacenar al menos \*\*10 años\*\* de datos académicos sin pérdida de rendimiento.

- El sistema debe admitir exportación de datos en formatos estándar como CSV y PDF.

**10. Auditoría y registro**

- El sistema debe registrar todas las acciones críticas (ej. inscripciones, cambios de calificaciones) con detalles de usuario, fecha y hora para fines de auditoría.

- Los logs de auditoría deben ser accesibles solo por administradores y conservarse durante al menos 5 años.

**11. Tolerancia a fallos**

- En caso de fallas del servidor, el sistema debe realizar un respaldo automático de la información crítica y permitir la recuperación en menos de 15 minutos.

**12. Internacionalización e idiomas**

- La interfaz debe estar inicialmente en español, con soporte para agregar otros idiomas en el futuro.

**13. Conformidad con regulaciones**

- El sistema debe cumplir con regulaciones de protección de datos, como la Ley de Protección de Datos Personales de Colombia.

1. **Otros requerimientos**

1. **Glosario**

**Administrador:** Usuario con privilegios avanzados para gestionar usuarios, datos y configuraciones del sistema.

**Algoritmo:** Conjunto finito de pasos o instrucciones para realizar una tarea o resolver un problema específico.

**API (Interfaz de Programación de Aplicaciones):** Conjunto de definiciones y protocolos para que diferentes sistemas se comuniquen entre sí.

**Autenticación:** Proceso para verificar la identidad de un usuario o sistema antes de permitirle acceso.

**Base de datos (BD):** Sistema organizado para almacenar, gestionar y recuperar datos de forma eficiente.

**Baja de usuarios:** Proceso por el cual un usuario, como un estudiante o profesor, es eliminado del sistema.

**Back-end:** Parte del sistema que incluye la lógica del servidor, procesamiento de datos y comunicación con la base de datos.

**Bug:** Error en el código que produce un comportamiento inesperado en el sistema.

**Cache:** Almacenamiento temporal de datos para mejorar la velocidad de acceso a estos.

**Ciclo de vida del desarrollo de software (SDLC):** Conjunto de fases para desarrollar un sistema desde su concepción hasta su implementación y mantenimiento.

**Cliente:** Componente o dispositivo que interactúa con el servidor para acceder a un sistema o servicio.

**Concurrencia:** Capacidad del sistema para manejar múltiples operaciones o usuarios al mismo tiempo.

**Configuración:** Ajustes o parámetros que determinan el funcionamiento del sistema.

**Consulta:** Proceso de recuperación de información específica de la base de datos.

**CRUD:** Acrónimo de Create, Read, Update, Delete; operaciones básicas sobre los datos en un sistema.

**Dashboard:** Pantalla que muestra información relevante de manera resumida y visualmente atractiva.

**Datos sensibles:** Información personal o confidencial que requiere protección especial.

**Defecto:** Diferencia entre el comportamiento esperado y el observado en el sistema.

**Encriptación:** Proceso para convertir datos en un formato ilegible para protegerlos contra accesos no autorizados.

**Escalabilidad:** Capacidad del sistema para manejar un aumento en la carga de trabajo o usuarios.

**Especificación:** Documento detallado que define los requisitos y características de un sistema o componente.

**Error:** Problema detectado en la fase de ejecución de un programa.

**Evaluación:** Proceso de medir la efectividad y calidad de un sistema o componente.

**Fallos:** Eventos en los que el sistema no cumple con su función esperada.

**Front-end:** Parte del sistema que interactúa directamente con el usuario, como la interfaz gráfica.

**Funcionalidad:** Característica o capacidad específica que un sistema debe ofrecer.

**Gestión de usuarios:** Proceso de creación, modificación y eliminación de cuentas de usuario en el sistema.

**GUI (Interfaz Gráfica de Usuario):** Interfaz que permite a los usuarios interactuar con el sistema mediante elementos visuales.

**Hardware:** Componentes físicos de un sistema informático, como servidores, computadoras y dispositivos móviles.

**Historial:** Registro de acciones o eventos en el sistema.

**HTTP/HTTPS:** Protocolo de comunicación para transferir datos en la web, con HTTPS asegurando la transmisión cifrada.

**Inscripción:** Proceso mediante el cual un estudiante se registra en materias o actividades específicas.

**Interfaz:** Punto de interacción entre diferentes sistemas o entre el usuario y el sistema.

**Iteración:** Repetición de un proceso de diseño o desarrollo para mejorar el sistema.

**Latencia:** Tiempo que tarda el sistema en responder a una solicitud del usuario.

**Login:** Proceso de autenticación que permite a un usuario acceder al sistema mediante credenciales.

**Logs:** Registros de eventos o actividades realizados en el sistema.

**Mantenimiento:** Actividades realizadas para garantizar el correcto funcionamiento del sistema a lo largo del tiempo.

**Middleware:** Software que actúa como intermediario entre diferentes aplicaciones o sistemas.

**Mockup:** Representación gráfica de cómo se verá el sistema, utilizada para diseño y planificación.

**Panel de control:** Sección del sistema donde los usuarios pueden gestionar información o configuraciones.

**Parámetro:** Dato o variable que se pasa a una función o sistema para personalizar su comportamiento.

**Perfil de usuario:** Información relacionada con un usuario, incluyendo detalles personales y preferencias.

**Prioridad:** Nivel de importancia asignado a un requisito o tarea dentro del sistema.

**Privilegios:** Nivel de acceso o permisos asignados a un usuario o rol en el sistema.

**Prototipo:** Modelo preliminar del sistema, utilizado para pruebas y validaciones.

**Requerimiento:** Necesidad o condición que un sistema debe cumplir.

**Requerimiento funcional:** Describe lo que el sistema debe hacer para cumplir con las necesidades del usuario.

**Requerimiento no funcional:** Describe cómo debe operar el sistema, incluyendo rendimiento, seguridad y usabilidad.

**Respaldo:** Copia de seguridad de los datos para prevenir pérdidas en caso de fallos.

**Responsiveness:** Capacidad de una interfaz para adaptarse a diferentes dispositivos y tamaños de pantalla.

**Rol:** Conjunto de permisos asignados a un usuario según sus responsabilidades dentro del sistema.

**Seguridad:** Medidas y prácticas para proteger los datos y recursos del sistema contra accesos no autorizados.

**Servidor:** Computadora o sistema que proporciona servicios y datos a otros dispositivos conectados.

**Test:** Proceso de evaluación del sistema para asegurar que cumple con los requisitos especificados.

**Token:** Elemento utilizado para autenticar y autorizar usuarios en un sistema.

**Usuario final:** Persona que interactúa directamente con el sistema para cumplir sus necesidades.

**Validación:** Proceso de comprobar que el sistema cumple con las necesidades y requisitos definidos.

**VPN (Red Privada Virtual):** Tecnología que proporciona una conexión segura entre dispositivos y redes.

**Workflow (Flujo de trabajo):** Secuencia de tareas o pasos que se realizan dentro de un sistema.