**DINOSAPIENS**

**PROGRAMACIÓN DE SOFTWARE**

**Integrantes**

**Andrés Moreno, Daniela Cotes, Jaime Rosero y Sebastian Niño**

**SENA – CENTRO DE MATERIALES Y ENSAYOS.**

**PROGRAMACION DE SOFTWARE.**

**FICHA 2875259.**

**2024**

1. **Introducción**

El proyecto pretende ayudar más en el aprendizaje de nuestros niños, a los que se les dificulta aprender y desarrollar habilidades en diferentes materias. Se crea Dinosapiens, un aplicativo que cumple con el desafío de brindar una forma didáctica y divertida de aprendizaje, sin que la necesidad de aprender sea tan monótona y en ciertos casos aburrida.

* 1. **Propósito del sistema.**

Implementar una aplicación como Dinosapiens busca darle una nueva cara a la educación actual, en la que el preadolescente, como el adolescente pueda aprender de manera didáctica, divertida, sin dejar a un lado los principios fundamentales de la educación, esto con diversos ejercicios que dan un toque diferencial en la forma de estudiar y por ende aprender de manera que el  aprendizaje no sea monótono, al contrario que quiera aprender más porque le gusta la metodología que tiene para ellos Dinosapiens.

**1.2. Objetivos del diseño**

- Crear módulos educativos que cubran los principios fundamentales de las materias escolares, adaptados a las edades de preadolescentes y adolescentes.

- Diseñar actividades y ejercicios interactivos que se basen en gamificación para hacer el aprendizaje más atractivo y motivador.

- Implementar desafíos y juegos educativos que recompensen el progreso del usuario, estimulando la participación continua.

- Incluir elementos sociales, como la posibilidad de que los usuarios compartan sus logros y compitan amistosamente con amigos en actividades educativas.

- Desarrollar un sistema de adaptación que ajuste el nivel de dificultad y los contenidos basados en el progreso y las necesidades individuales de cada usuario.

- Ofrecer opciones de personalización para que los usuarios puedan elegir temas y estilos de aprendizaje que les resulten más interesantes y adecuados.

- Implementar sistemas de retroalimentación inmediata en los ejercicios para que los usuarios puedan corregir errores y mejorar continuamente.

- Proporcionar informes de progreso detallados para que los usuarios y sus padres o tutores puedan seguir el avance e identificar áreas de mejora.

- Utilizar tecnología de realidad aumentada para crear experiencias de aprendizaje inmersivas que hagan que los conceptos educativos cobren vida de manera visual y dinámica.

- Diseñar una interfaz de usuario amigable y atractiva que facilite la navegación y el acceso a los contenidos educativos.

- Asegurar que la aplicación sea accesible en dispositivos móviles y tabletas para llegar a una mayor audiencia.

- Incorporar funciones que permitan a los usuarios trabajar en proyectos y actividades en grupo, fomentando habilidades de colaboración y trabajo en equipo.

- Crear espacios dentro de la aplicación donde los usuarios puedan intercambiar ideas y resolver problemas en conjunto.

- Implementar herramientas de evaluación y pruebas para medir la comprensión y retención de los conceptos aprendidos.

- Realizar encuestas y recopilar comentarios de los usuarios para mejorar continuamente la calidad y relevancia del contenido educativo.

**1.3. Definiciones, acrónimos y abreviaturas**

**Adaptación**: Sistema que ajusta el nivel de dificultad y contenido basado en el progreso y necesidades del usuario.

**Aplicación**: Software que se utiliza en dispositivos electrónicos, en este caso, Dinosapiens.

**Dinosapiens**: Nombre del aplicativo que busca enseñar de manera didáctica y divertida a preadolescentes y adolescentes.

**Gamificación**: Uso de mecánicas de juego en entornos no lúdicos para hacer más atractivo y motivador el aprendizaje.

**Interfaz de usuario**: Parte visible de la aplicación con la que los usuarios interactúan, diseñada para ser amigable y atractiva.

**Módulos educativos**: Secciones que cubren los principios fundamentales de las materias escolares.

**Preadolescentes**: Niños entre 9 y 12 años aproximadamente, el grupo objetivo de Dinosapiens.

**Realidad aumentada (RA)**: Tecnología que combina el mundo real con elementos virtuales para crear experiencias inmersivas.

**Retroalimentación inmediata**: Proceso por el cual el sistema informa de forma instantánea sobre el desempeño de los usuarios en los ejercicios.

**Sistema de evaluación**: Herramientas para medir la comprensión y retención de conceptos mediante pruebas.

**Tabletas**: Dispositivos móviles en los que se puede acceder a la aplicación.

**Usuarios**: Las personas que utilizan la aplicación, en este caso, preadolescentes, adolescentes y padres/tutores.

**1.4. Referencias**

**Referencias a otros documentos e información de trazabilidad**

**Documento de análisis de requisitos relacionados:**

**Análisis de necesidades educativas para preadolescentes y adolescentes**: Este documento describe los desafíos actuales en el aprendizaje de materias escolares, específicamente en matemáticas, y cómo el aplicativo **Dinosapiens** pretende resolver estas dificultades a través de módulos educativos y actividades interactivas.

**Investigación sobre el impacto de la gamificación en el aprendizaje**: Un estudio sobre cómo el uso de la **gamificación** puede motivar a los estudiantes, mejorar la retención de información y hacer el aprendizaje más atractivo y efectivo, que fundamenta la implementación de **desafíos y juegos educativos** en Dinosapiens.

**Sistemas existentes relacionados:**

**Plataformas educativas interactivas**: Se ha tomado en cuenta el diseño de otras plataformas educativas interactivas como referencia para desarrollar un sistema que proporcione **retroalimentación inmediata**, **personalización de contenidos** y una **interfaz amigable**.

**Aplicaciones de realidad aumentada (RA)**: Se consideraron aplicaciones que usan **tecnología de RA** para crear experiencias inmersivas en el proceso de aprendizaje, para evaluar la integración de esta tecnología en el diseño de **Dinosapiens**.

**Restricciones que afectan a la arquitectura del software:**

**Compatibilidad con dispositivos móviles y tabletas:** La aplicación debe acceder a muchos dispositivos móviles, lo que limita el uso de recursos y el diseño de la interfaz para asegurar que funcione de manera óptima en distintos sistemas operativos.

**Adaptación a diferentes niveles educativos**: El sistema debe ajustarse automáticamente según el progreso de cada usuario, lo que requiere una arquitectura flexible que pueda manejar dinámicamente los cambios en el contenido y la dificultad.

**Interacción social**: La aplicación debe permitir a los usuarios compartir logros y competir amistosamente, lo que implica el uso de servidores en la nube y la necesidad de gestionar la seguridad y privacidad de los datos de los usuarios.

**2. Representación de la arquitectura.**

**2.1 Metas y restricciones de la Arquitectura**

**Metas de la Arquitectura**

**Rendimiento**:

Hay que asegurar que la aplicación tenga tiempos de carga rápidos y responda de manera eficiente a las interacciones de los usuarios.

Optimizar el rendimiento en dispositivos móviles y de escritorio.

**Escalabilidad**:

Permitir que la aplicación soporte un aumento en el número de usuarios sin degradar la experiencia de uso.

Facilitar la adición de nuevas funcionalidades y contenido educativo en el futuro.

**Mantenibilidad**:

Diseñar el código y la estructura de la aplicación de manera que sea fácil de actualizar y corregir.

Documentar el sistema para que otros desarrolladores puedan entender y trabajar en él fácilmente.

**Seguridad**:

Implementar medidas para proteger la información personal y los datos de los usuarios.

Hay que asegurar que la aplicación esté protegida contra accesos no autorizados y vulnerabilidades.

**Usabilidad**:

Crear una interfaz intuitiva y amigable para los preadolescentes y adolescentes.

Incorporar elementos visuales y de interacción que hagan el aprendizaje divertido y atractivo.

**Costo**:

Optimizar los recursos para ajustarse al presupuesto disponible, tanto en desarrollo como en operación.

Considerar soluciones de código abierto y servicios en la nube que puedan reducir costos.

**Flexibilidad**:

Permitir que la aplicación se adapte a cambios en el contenido educativo y en las necesidades del usuario.

Facilitar la integración con nuevas tecnologías y plataformas educativas.

**Restricciones de la Arquitectura**

**Recursos**:

Limitar el uso de recursos del servidor y del dispositivo para asegurar un rendimiento óptimo sin sobrecargar el sistema.

**Tecnología**:

Usar tecnologías que sean compatibles con los dispositivos y sistemas operativos más comunes entre los usuarios objetivo.

Asegurarse de que el stack tecnológico seleccionado sea adecuado para las necesidades de la aplicación.

**Tiempo**:

Cumplir con los plazos de desarrollo establecidos para el lanzamiento inicial de la aplicación.

Planificar el tiempo necesario para pruebas y corrección de errores antes del lanzamiento.

**Regulaciones**:

Cumplir con las normativas relacionadas con la protección de datos de menores y otras regulaciones educativas.

Asegurarse de que la aplicación cumpla con las políticas de privacidad y seguridad relevantes.

**Interoperabilidad**:

Garantizar que la aplicación pueda integrarse con otras plataformas educativas o sistemas de gestión de aprendizaje si es necesario.

**Compatibilidad**:

Asegurarse de que la aplicación funcione correctamente en diferentes dispositivos, navegadores y sistemas operativos.

**Costo**:

Ajustar el diseño y las decisiones tecnológicas para mantenerse dentro del presupuesto asignado para el desarrollo y el mantenimiento de la aplicación.

**2.2 Reutilización**

**Componentes Reutilizables**:

**Modularidad**: Diseña la aplicación en módulos independientes, como un módulo para la gestión de usuarios, otro para el contenido educativo, y otro para la evaluación y seguimiento del progreso. Esto permitirá reutilizar estos módulos en otras partes de la aplicación o incluso en futuros proyectos.

**Bibliotecas y Frameworks**: Utiliza bibliotecas y frameworks que ya resuelvan problemas comunes (como manejo de formularios, gráficos o autenticación de usuarios). Esto evita reinventar la rueda y acelera el desarrollo.

**Plantillas y Componentes de Interfaz de Usuario (UI)**:

**Diseño de UI**: Crea componentes de interfaz de usuario reutilizables, como botones, formularios y cuadros de diálogo, que puedan ser utilizados en diferentes partes de la aplicación. Esto no solo mejora la consistencia, sino que también facilita el mantenimiento.

**Plantillas de Contenido**: Diseña plantillas de contenido educativo que puedan ser fácilmente adaptadas para diferentes temas o lecciones.

**Servicios y API**:

**Servicios Backend**: Desarrolla servicios backend que puedan ser reutilizados para diferentes funcionalidades. Por ejemplo, un servicio para manejar el progreso del usuario puede ser reutilizado tanto para mostrar estadísticas como para generar informes.

**API:** Define API para integraciones con otras aplicaciones o plataformas educativas, permitiendo una fácil extensión y conexión con otros sistemas.

**Código y Lógica de Negocio**:

**Funciones y Métodos Reutilizables**: Escribe funciones y métodos que encapsulen la lógica común y puedan ser reutilizados en diferentes partes de la aplicación. Por ejemplo, funciones para calcular puntuaciones o validar respuestas pueden ser reutilizadas en diferentes módulos.

**Patrones de Diseño**: Utiliza patrones de diseño (como el patrón de diseño de fábrica, singleton o estrategia) que promuevan la reutilización del código y faciliten su adaptación a nuevas necesidades.

**Educación y Recursos**:

**Contenido Educativo**: Desarrolla recursos educativos (como ejercicios y cuestionarios) que puedan ser reutilizados en diferentes lecciones o temas. Considera la posibilidad de crear un repositorio de contenido que se pueda actualizar y expandir fácilmente.

**Pruebas y Documentación**:

**Casos de Prueba**: Crea casos de prueba reutilizables para validar diferentes partes de la aplicación. Esto facilitará la automatización de pruebas y asegurará que los cambios no rompan funcionalidades existentes.

**Documentación**: Documenta componentes y servicios de manera que sea fácil para otros desarrolladores reutilizarlos y entender cómo integrarlos en diferentes partes del sistema.

1. **Vista lógica**

**Modelos de Datos**:

**Usuarios**: Define las entidades relacionadas con los usuarios, como perfiles de estudiantes y administradores, historial de progresos, y configuraciones de usuario.

**Contenido Educativo**: Estructura la información educativa, incluyendo temas, lecciones, ejercicios, y evaluaciones. Incluye cómo se organizan y se relacionan entre sí.

**Progreso y Evaluaciones**: Modela cómo se rastrea el progreso de los estudiantes y cómo se gestionan las evaluaciones y retroalimentaciones.

**Componentes de Negocio**:

**Gestión de Usuarios**: Procesos para registrar, autenticar, y gestionar usuarios, así como para asignar roles y permisos.

**Gestión del Contenido**: Módulos que permiten la creación, edición, y organización del contenido educativo, así como la personalización del aprendizaje.

**Evaluación y Retroalimentación**: Procesos para evaluar el desempeño del estudiante, generar informes y proporcionar retroalimentación.

**Servicios y APIs**:

**Servicios de Autenticación**: Encargados de gestionar el inicio de sesión, registro y autenticación de usuarios.

**Servicios de Contenido**: Encargados de servir el contenido educativo, gestionar el acceso y entregar material relevante basado en el progreso del usuario.

**Servicios de Evaluación**: Manejan la generación de evaluaciones, la recopilación de respuestas, y el cálculo de puntuaciones.

**Interacciones y Flujo de Datos**:

**Interacción Usuario-Contenido**: Cómo los usuarios interactúan con el contenido educativo, incluyendo la navegación por lecciones, la realización de ejercicios y la revisión de resultados.

**Interacción Usuario-Administración**: Cómo los administradores gestionan el contenido, supervisan el progreso de los estudiantes y configuran parámetros del sistema.

**Flujo de Datos**: Cómo los datos fluyen entre los componentes del sistema, como el flujo de información desde el usuario hacia la base de datos y viceversa.

**Reglas de Negocio**:

**Reglas de Progreso**: Cómo se calculan y actualizan los niveles de progreso de los estudiantes.

**Reglas de Evaluación**: Cómo se generan y califican las evaluaciones, y cómo se proporciona retroalimentación basada en los resultados.

**Reglas de Personalización**: Cómo se personaliza la experiencia de aprendizaje para cada estudiante basado en su progreso y desempeño.

**Integraciones**:

**Servicios Externos**: Define cómo se integrará la aplicación con servicios externos, como plataformas de pago, sistemas de autenticación de terceros, o APIs educativas adicionales.

**Interoperabilidad**: Asegura que los componentes del sistema puedan comunicarse y trabajar juntos de manera efectiva.

* 1. **Identificación de Subsistemas**

**Subsistema de Gestión de Contenidos**:

**Descripción**: Este subsistema se encarga de gestionar el contenido educativo, como los ejercicios de matemáticas, español, inglés y videos, y recursos didácticos.

**Componentes**: Base de datos de contenidos, editor de contenidos, sistema de categorización y etiquetado.

**Subsistema de Usuario**:

**Descripción**: Maneja la gestión de usuarios, incluyendo el registro, inicio de sesión, y perfiles de usuario.

**Componentes**: Registro y autenticación de usuarios, perfiles de usuario, gestión de roles y permisos.

**Subsistema de Ejercicios y Evaluaciones**:

**Descripción**: Se encarga de crear, gestionar y evaluar ejercicios matemáticos y pruebas.

**Componentes**: Generador de ejercicios, evaluador automático, sistema de retroalimentación.

**Subsistema de Interacción y Gamificación**:

**Descripción**: Implementa elementos de gamificación para hacer el aprendizaje más atractivo, como puntos, medallas, y niveles.

**Componentes**: Sistema de recompensas, desafíos y misiones, tablas de clasificación.

**Subsistema de Informes y Estadísticas**:

**Descripción**: Recopila y presenta datos sobre el progreso del usuario, rendimiento en ejercicios y estadísticas generales.

**Componentes**: Generador de informes, panel de estadísticas, gráficos de progreso.

**Subsistema de Soporte y Feedback**:

**Descripción**: Proporciona asistencia al usuario y recolecta retroalimentación para mejorar la aplicación.

**Componentes**: Sistema de soporte en línea, encuestas de satisfacción, sistema de reportes de problemas.

**Subsistema de Configuración y Administración**:

**Descripción:** Permite a los administradores de la aplicación configurar y gestionar el sistema.

**Componentes**: Panel de administración, configuración del sistema, gestión de permisos y accesos.

**Estructuración por capas.**

**Capa de Presentación**:

**Descripción**: Esta capa es la interfaz de usuario y la experiencia del usuario. Se encarga de la interacción con el usuario final.

**Componentes**:

Interfaces gráficas de usuario (GUI), como pantallas de inicio, menús, y formularios.

Elementos de interacción, como botones, sliders y campos de texto.

Componentes de visualización, como gráficos de progreso y tablas de clasificación.

**Capa de Aplicación**:

**Descripción**: Esta capa contiene la lógica de negocio y las reglas específicas de la aplicación. Aquí se procesa la información y se coordinan las actividades entre la interfaz de usuario y la capa de datos.

**Componentes**:

Controladores que gestionan las solicitudes del usuario y coordinan la comunicación entre la capa de presentación y la capa de datos.

Servicios que implementan la lógica de negocio, como la gestión de ejercicios, la evaluación automática y la gamificación.

Procesadores de datos que transforman la información antes de enviarla a la capa de presentación o almacenar en la capa de datos.

**Capa de Datos**:

**Descripción**: Se encarga del almacenamiento y recuperación de datos. Esta capa interactúa con bases de datos y otros sistemas de almacenamiento.

**Componentes**:

**Base de Datos**: Almacena la información de usuarios, ejercicios, evaluaciones, y estadísticas.

**Repositorios**: Interfaces que proporcionan métodos para acceder a la base de datos y manipular los datos.

**Servicios de Datos**: Implementan la lógica para acceder a los datos y realizar operaciones CRUD (crear, leer, actualizar, eliminar).

**Capa de Integración**:

**Descripción**: Facilita la comunicación con servicios externos y otras aplicaciones. Maneja la integración con APIs y servicios de terceros.

**Componentes**:

**APIs Externas**: Para integrar servicios adicionales como herramientas de análisis, plataformas de pago, o servicios de autenticación.

**Adaptadores**: Componentes que adaptan la comunicación entre tu aplicación y servicios externos.

**Capa de Seguridad**:

**Descripción**: Asegura que la aplicación sea segura y protege los datos del usuario.

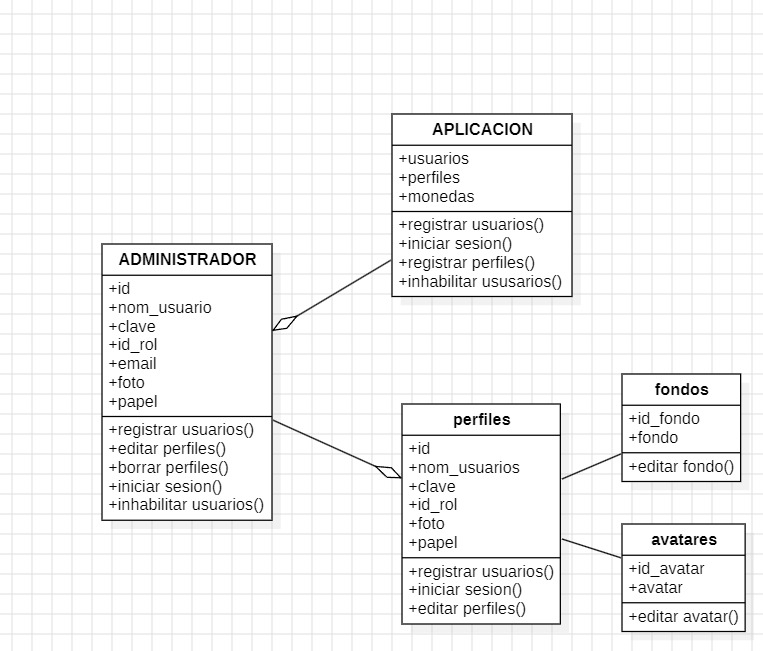
**Componentes**:

**Autenticación y Autorización**: Gestión de acceso de usuarios y permisos.

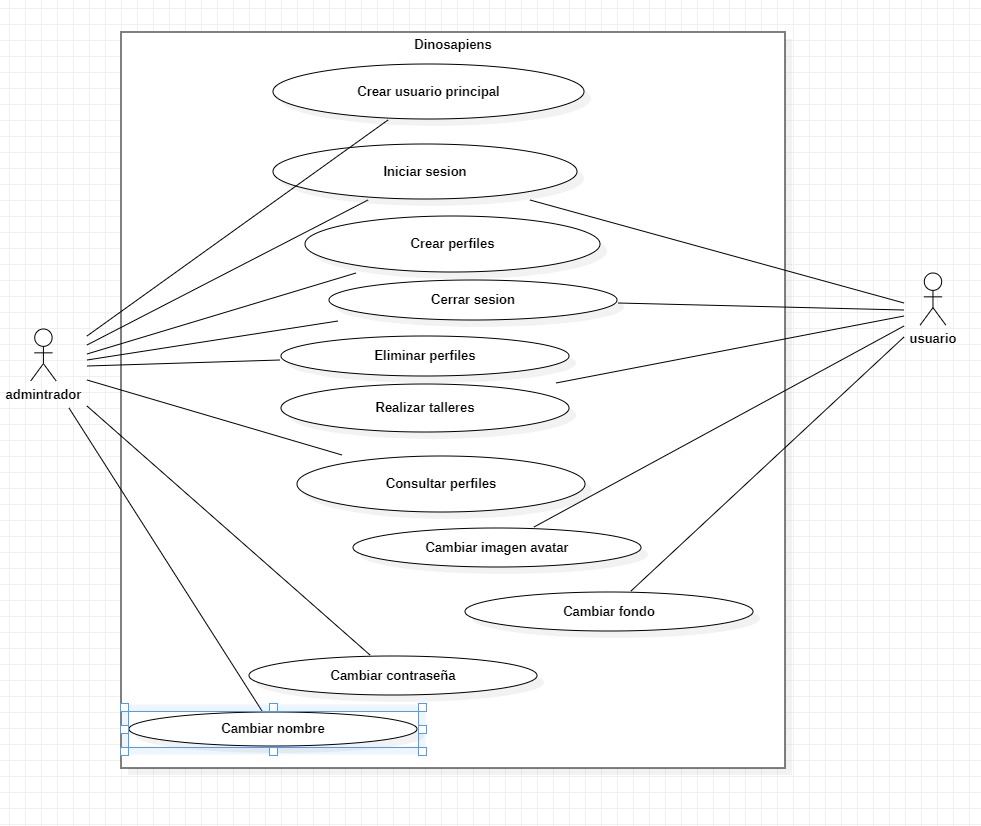
**Cifrado**: Protección de datos sensibles tanto en tránsito como en reposo.

**Protección contra ataques**: Implementación de medidas de seguridad para proteger contra vulnerabilidades comunes.

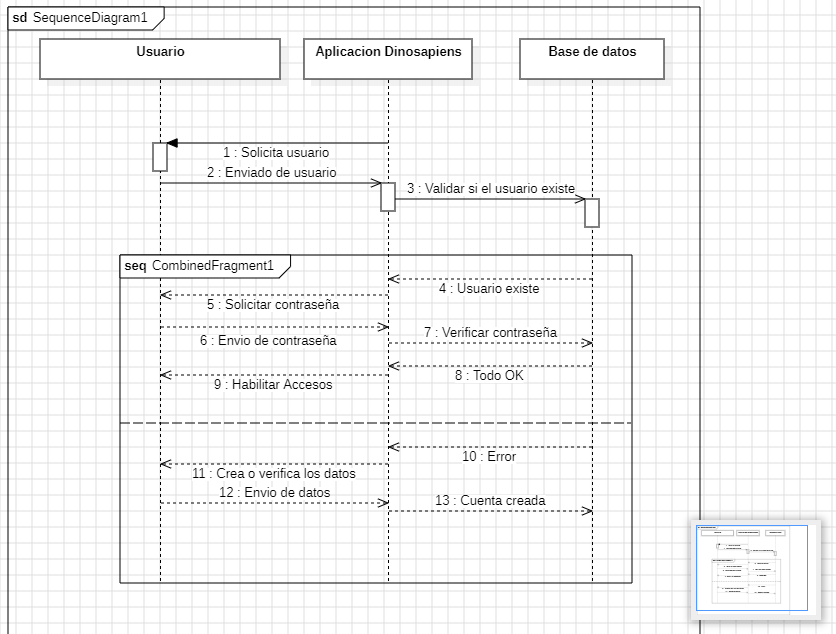
* 1. **Diagrama de Clases del diseño**



1. **Vista del proceso**



* 1. **Diagramas de secuencia**



* 1. **Modelo de datos**

**1. Tabla: administrador**

Esta tabla almacena información sobre los administradores de la plataforma.

| **Campo** | **Tipo** | **Descripción** |
| --- | --- | --- |
| id | INT (PK) | Identificador único del administrador. |
| nomusuario | VARCHAR(15) | Nombre de usuario del administrador (mín. 4 caracteres, máx. 15, solo letras minúsculas). |
| clave | VARCHAR(11) | Contraseña del administrador (mín. 5 caracteres, máx. 11). |
| idrol | INT | Identificador del rol del administrador (1: Administrador). |
| foto | VARCHAR | Ruta de la imagen de perfil del administrador. |

**2. Tabla: perfiles**

Esta tabla almacena los perfiles de los exploradores (usuarios no administradores).

| **Campo** | **Tipo** | **Descripción** |
| --- | --- | --- |
| id | INT (PK) | Identificador único del perfil. |
| nomusuario | VARCHAR(15) | Nombre de usuario del explorador (mín. 4 caracteres, máx. 15, solo letras minúsculas). |

|  |  |  |
| --- | --- | --- |
| clave | VARCHAR(4) | Contraseña del explorador (4 caracteres, patrón exacto). |
| idrol | INT | Identificador del rol del explorador. |
| foto | VARCHAR | Ruta de la imagen de perfil del explorador. |

Tabla de avatares

| **Campo** | **Tipo** | **Descripción** |
| --- | --- | --- |
| Id\_avatar | INT (PK) | Identificador único del avatar |
| avatar | VARCHAR(15) | Ruta de la imagen de perfil del avatar. |

Tabla fondos

| **Campo** | **Tipo** | **Descripción** |
| --- | --- | --- |
| Id\_fondo | INT (PK) | Identificador único del fondo |
| fondo | VARCHAR(15) | Ruta de la imagen de perfil del fondo. |

Tabla roles

| **Campo** | **Tipo** | **Descripción** |
| --- | --- | --- |
| Id | INT (PK) | Identificador único del rol |
| rol | VARCHAR(15) | Indica que rol es el usuario que ingresa |

* 1. **Diccionario de datos**

**Diccionario de Datos**

* **Descripción de Funciones Clave:**
* **Inicio de Sesión:**
  + Se verifica la sesión con $\_SESSION['rol']. Solo los usuarios con rol de administrador (rol = 1) pueden acceder al panel de administración.
* **Visualización de Datos:**
  + Los administradores y exploradores se listan en tablas con sus respectivos nombres de usuario y avatares (imágenes de perfil).
* **Actualización de Datos:**
  + Los administradores y exploradores pueden actualizar sus nombres de usuario y contraseñas. El formulario envía los datos actualizados a la base de datos mediante la consulta UPDATE.
* **Eliminación de Perfiles:**
  + Los administradores pueden eliminar perfiles de exploradores mediante la opción "BORRAR" en la tabla de usuarios.
* **Tabla: avatares**
  + **Descripción**: Esta tabla contiene los diferentes avatares disponibles para los usuarios.
  + **Campos**:
    - id\_avatar (INT, PK, AUTO\_INCREMENT): Identificador único de cada avatar.
    - avatar (VARCHAR): Nombre o ruta de la imagen del avatar.
* **Tabla: perfiles**
  + **Descripción**: Almacena la información de los usuarios y su configuración personalizada, como la imagen de avatar seleccionada.
  + **Campos**:
    - id (INT, PK, AUTO\_INCREMENT): Identificador único del perfil de usuario.
    - foto (VARCHAR): Almacena el nombre o la ruta de la imagen del avatar seleccionada por el usuario.
* **Sesión de usuario ($\_SESSION)**
  + **$\_SESSION['rol']**: Almacena el rol del usuario, que se utiliza para validar el acceso a la página (en este caso, rol 2).
  + **$\_SESSION['id']**: Almacena el ID del usuario logueado, que se utiliza para actualizar la información del perfil.
  + **Resumen de Funcionalidades:**
* **Consulta de avatares**:
  + Se realiza una consulta (SELECT \* FROM avatares) para obtener todos los avatares disponibles y mostrarlos en la página.
* **Cambio de avatar**:
  + El usuario selecciona un avatar, y se ejecuta una actualización en la tabla perfiles, modificando el campo foto con la nueva imagen seleccionada.
  + **Relación entre tablas:**
* **avatares**: Contiene la lista de avatares disponibles.
* **perfiles**: Contiene los datos del perfil del usuario, incluyendo el avatar seleccionado.

**Conexión a la Base de Datos**

* **$usuario**: Nombre de usuario para la conexión a la base de datos (en este caso, es "root").
* **$password**: Contraseña para la conexión a la base de datos (en este caso, está vacía "").
* **$servidor**: Dirección del servidor de base de datos (en este caso, "localhost").
* **$basededatos**: Nombre de la base de datos a la que se conecta (en este caso, "crud").
* **$conexion**: Variable que contiene el objeto de conexión a la base de datos, creado mediante mysqli\_connect.

**Consulta y Resultados**

* **$consulta**: Cadena de texto con la consulta SQL para obtener datos de la base de datos. Ejemplos:
  + "SELECT \* FROM usuarios": Obtiene todas las columnas de la tabla usuarios.
  + "SELECT id, nomusuario, clave, idrol, email FROM usuarios": Obtiene columnas específicas de la tabla usuarios.
  + "SELECT id, nomusuario, clave, idrol, email FROM usuarios LIMIT $inicio, $registros\_por\_pagina": Obtiene columnas específicas de la tabla usuarios con paginación.
* **$resultado**: Variable que almacena el resultado de la ejecución de la consulta SQL mediante mysqli\_query.
* **$fila**: Variable que almacena una fila de datos recuperada de la base de datos, obtenida mediante mysqli\_fetch\_array o mysqli\_fetch\_assoc.

**Columnas de la Tabla usuarios**

* **id**: Identificador único del usuario (entero).
* **nomusuario**: Nombre de usuario (cadena de texto).
* **clave**: Contraseña del usuario (cadena de texto).
* **idrol**: Identificador del rol del usuario (entero).
* **email**: Dirección de correo electrónico del usuario (cadena de texto).

**Paginación (en la parte de paginación del código)**

* **$registros\_por\_pagina**: Número de registros que se muestran por página (en este caso, 10).
* **$pagina\_actual**: Número de la página actual para la paginación (por defecto, 1).
* **$inicio**: Índice del primer registro a mostrar en la consulta actual, calculado en base a la página actual y el número de registros por página.
* **$total\_registros**: Número total de registros en la tabla usuarios.
* **$total\_paginas**: Número total de páginas basado en el número total de registros y el número de registros por página.

**Manejo de Errores**

* **mysqli\_connect\_error()**: Función utilizada para obtener el mensaje de error en caso de fallo en la conexión a la base de datos.
* **mysqli\_error($conexion)**: Función utilizada para obtener el mensaje de error en caso de fallo en la ejecución de la consulta SQL.
* **Exception**: Excepciones capturadas y mostradas en caso de errores durante la ejecución de consultas.

**Seguridad**

* **htmlspecialchars()**: Función utilizada para evitar ataques de Cross-Site Scripting (XSS) al escapar caracteres especiales en los datos mostrados.

**Estructura HTML**

* **<table>**: Elemento HTML utilizado para mostrar los datos en formato tabular.
* **<tr>**: Elemento HTML que define una fila en la tabla.
* **<th>**: Elemento HTML que define un encabezado de columna en la tabla.
* **<td>**: Elemento HTML que define una celda de datos en la tabla.
* **<a href="?pagina=1">1</a>**: Enlaces de paginación para navegar entre páginas de resultados.

1. **Vista de Interacción**

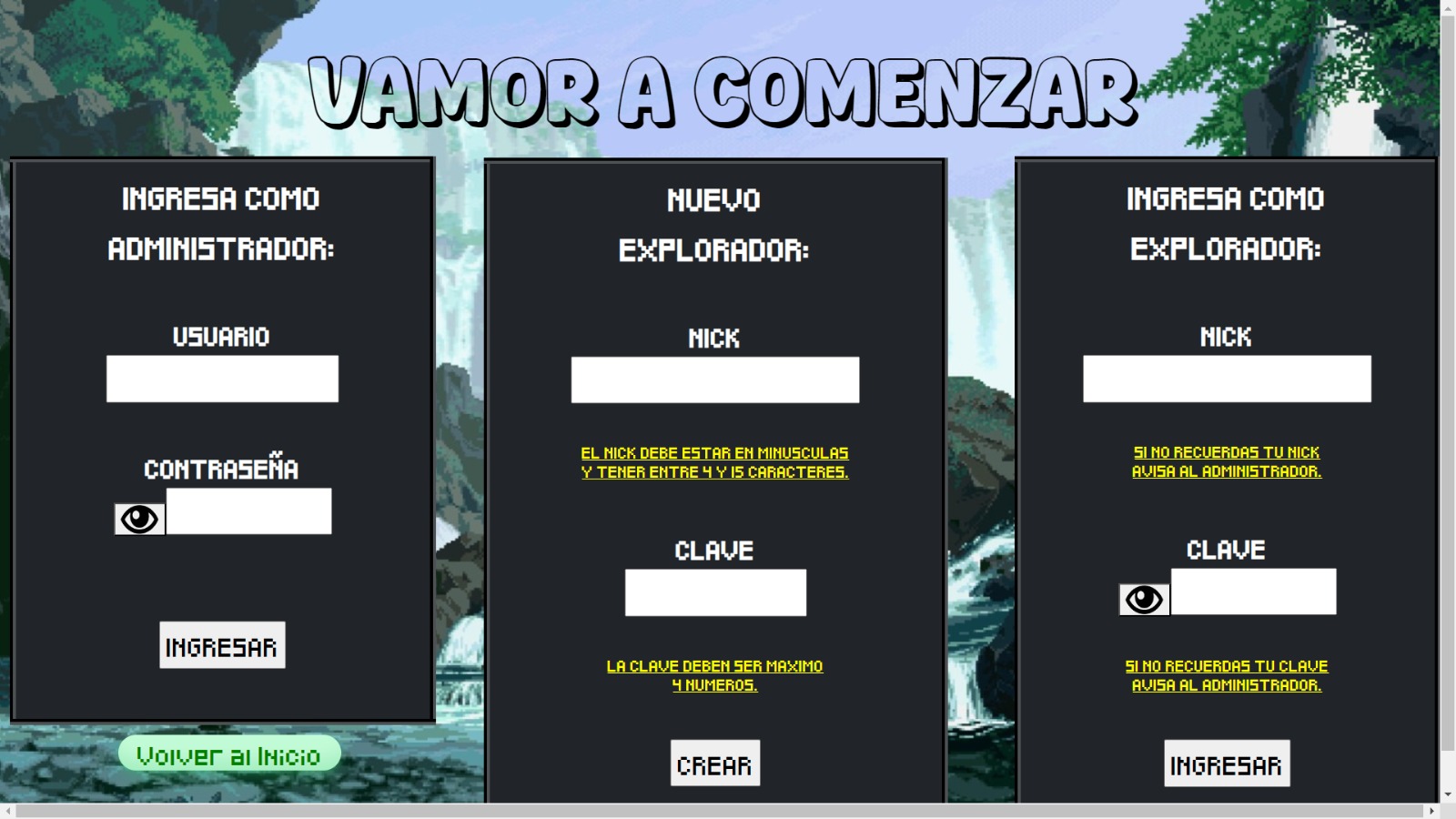
Esta es la pantalla con la que se inicia el aplicativo en la cual iniciaremos la aventura dandole en el boton “COMENZAR”. Al darle click en el botón de comenzar también creara la base de datos.



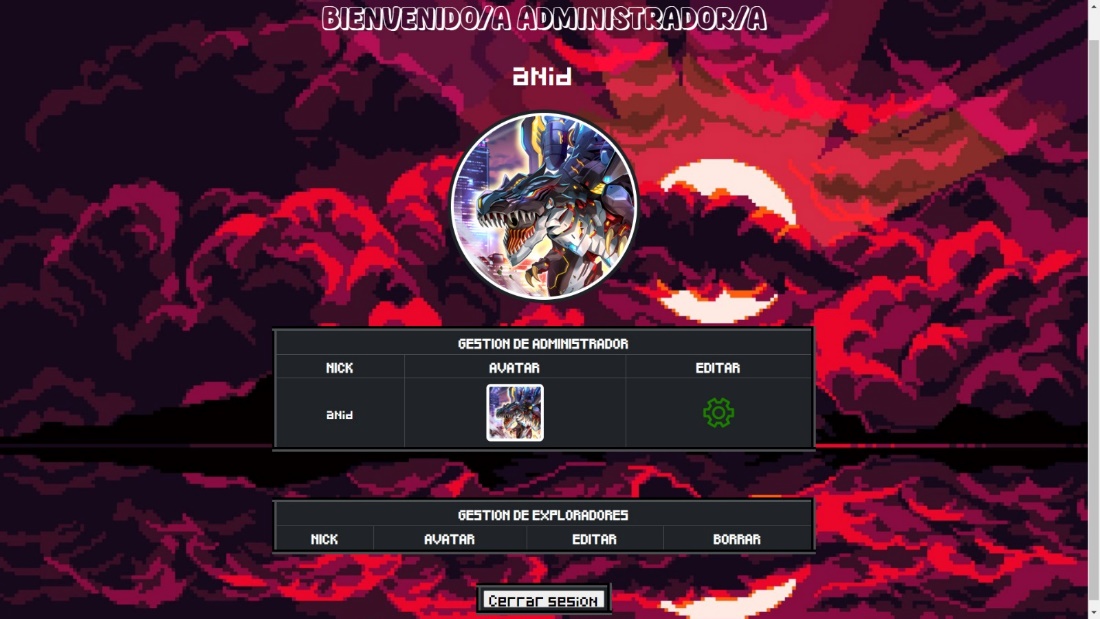
Esta es la pantalla donde crearemos el usuario principal con rol de admistrador el cual tendra contro sobre tood el sisitema



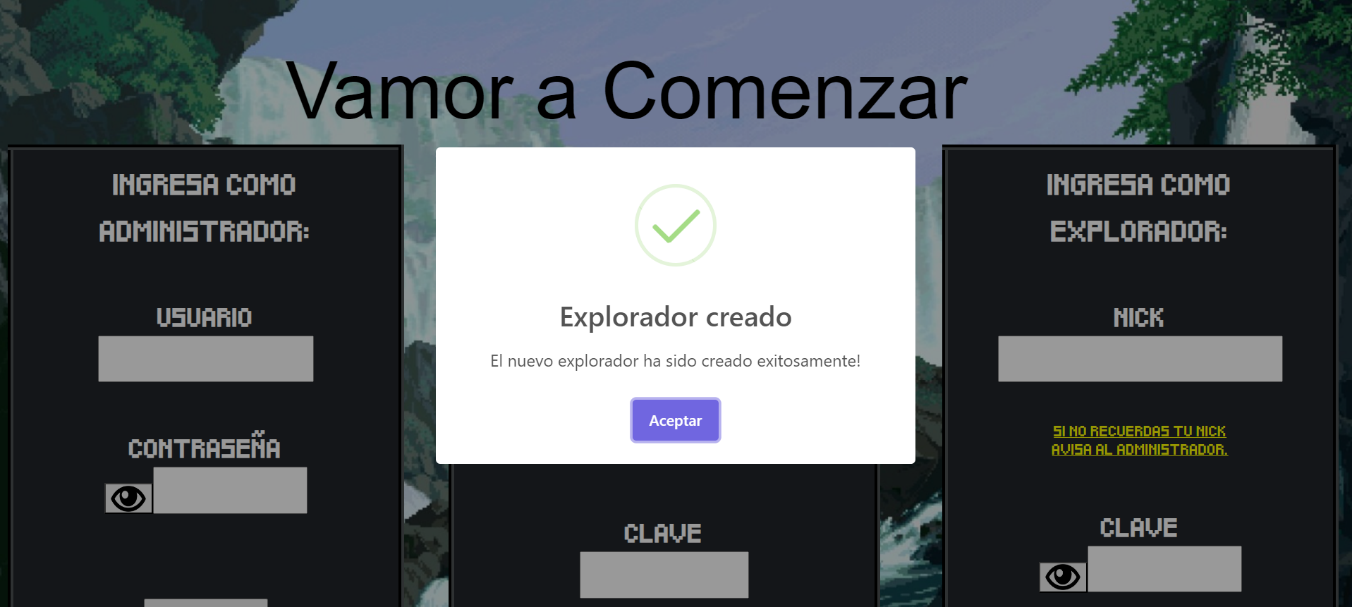
Esta en la pantalla donde ya creado el perfil de administrador, nos direcciona a la creación de los usuarios que son los que van a interactuar con el sistema



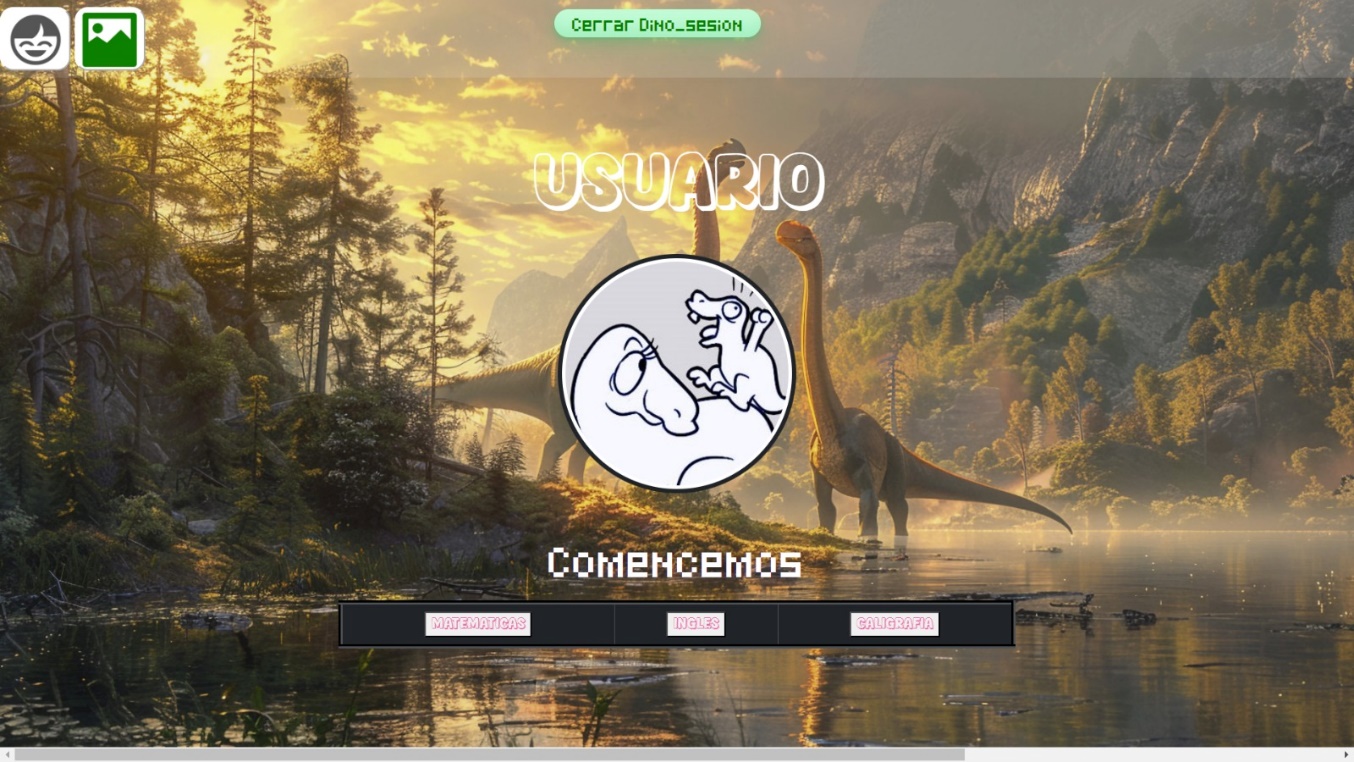
Esta es la pantalla donde el perfil de administrador tiene control sobre los perfiles



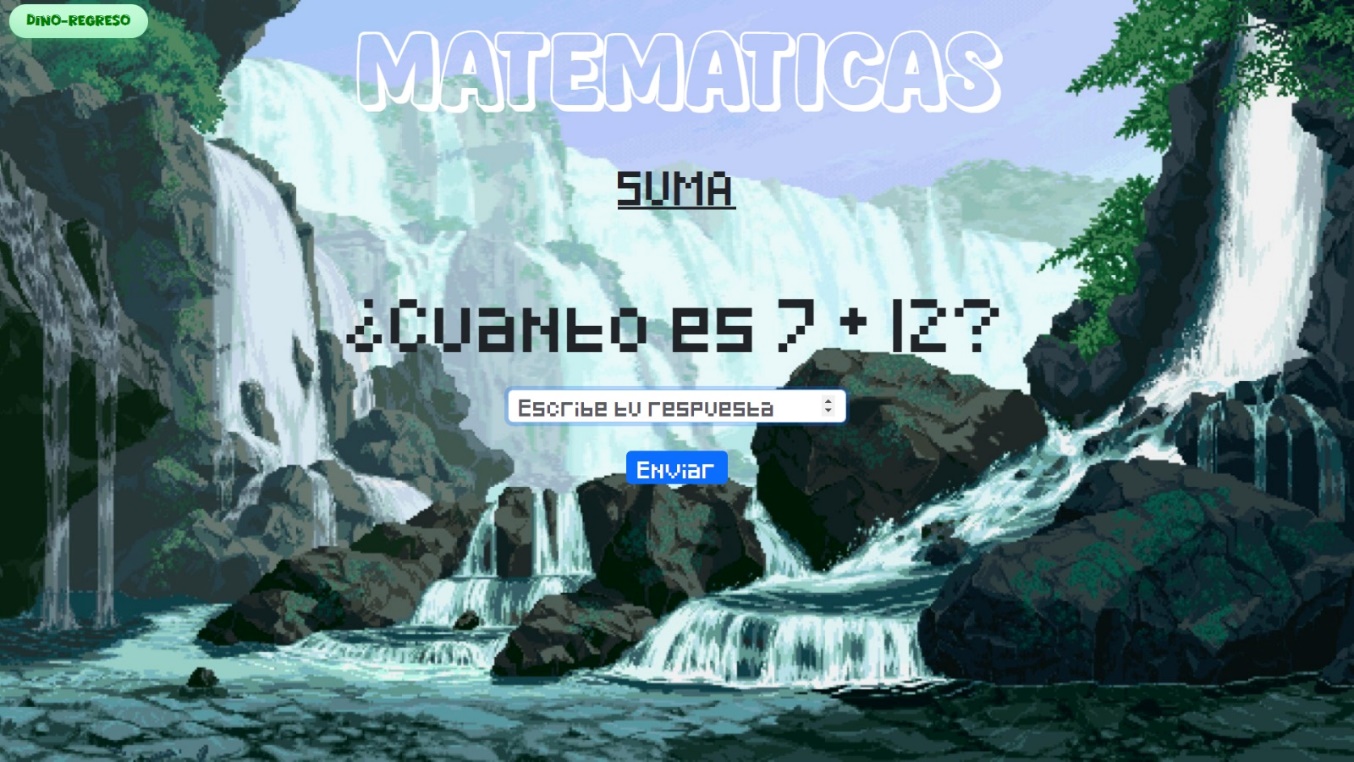
Esta es la pantalla donde creamos el usuario ”explorador” guardado con exito



Esta es la pantalla que se la mostrara a los usuarios , donde podrán practicar las actividades que deseen , dondo clic a cualquier de las materias



Estas son las pantallas donde se mostraran los talleres que los usuarios podran utilizar para poner en practica sus conocimientos



Aquí el explorador podra realizar el ejercicio de ingles



* 1. **Interfaces de Usuario.**

|  |  |
| --- | --- |
| **DSCU-001** | **Iniciar Sesión** |
| **Descripción** | El sistema deberá permitir al administrador y usuario, en cualquier momento, iniciar sesión según se describe en el siguiente caso de uso: |
|  | 1+ El sistema mostrara la pantalla de inicio de sesión    2+El sistema solicitara datos usuario y contraseña    3+El administrador, ingresara sus credenciales de usuario y contraseña de ingreso en el espacio correspondiente.    4+ El sistema lee los datos ingresados y verifica si son correctos.    5+ Si los datos son correctos abrirá la pantalla principal. |
| **Excepciones** | 2' Los datos ingresados son incorrectos.  +Se mostrará mensaje de “Datos incorrectos”  +Vuelve a la pantalla de inicio de sesión y solicita nuevamente las credenciales de ingreso. |

|  |  |
| --- | --- |
| **DSCU-009** | **Eliminar Perfil** |
| **Descripción** | El sistema deberá permitir al administrador en cualquier momento borrar los perfiles según se describe en el siguiente caso de uso: |
| **Secuencia Normal** | 1+ El administrador ingresa al perfil deseado desde la pantalla de perfiles.    2+ El sistema abre el perfil |
| 3+ El administrador selecciona la opción “eliminar perfil”.    4+ El sistema lee la opción.    5+ El sistema envía una ventana de confirmación. |
| 6+ Si el administrador elije “si”.    7+ El sistema limitará el perfil, su acceso, sus funciones y datos. Dejándolo oculto a la vista. |
| **Excepciones** | 6' Si el administrador elige “No”.  + El sistema regresará a la pantalla de perfiles. |

|  |  |  |
| --- | --- | --- |
| **DSCU-006** | **Consulta Perfiles** |  |
| **Descripción** | El sistema le permite al administrador en cada momento visualizar las los resultados y tiempo de uso de los perfiles según se describe en el siguiente caso de uso: |  |
| **Secuencia Normal** | 1+El administrador selecciona el icono de lupa del perfil correspondiente en la pantalla de perfiles.    2+ El sistema desplegara una cinta de opciones |  |
| 3+ El administrador selecciona la opción de ‘’mostrar calificaciones’’.    4+ El sistema le entregara el informe de las calificaciones de las notas de los perfiles según la asignatura de las actividades o ejercicios realizados. |  |
| 5+ El administrador selecciona la opción “tiempo de uso”.    6+ El sistema mostrara la página con el tiempo de uso en horas y minutos del perfil en cuestión. |  |
| **Excepciones** | 3' Si el usuario no elige ninguna opción, el sistema le cerrara el menú de opciones pasados 7 segundos de seleccionar la opción ‘’mostrar calificaciones’’. |  |

1. **Vista de seguridad**

**Sistema de Acceso**

**Nivel de Seguridad de Acceso:** El acceso al sistema Dinosapiens se controla mediante un sistema de autenticación robusto que protege las áreas sensibles de la aplicación y los datos de los usuarios. El nivel de seguridad incluye:

* **Autenticación:** Los usuarios deben ingresar un nombre de usuario y una contraseña para acceder al sistema. Las contraseñas son almacenadas de manera segura en la base de datos.
* **Autorización:** Dependiendo del perfil del usuario, se determinan los permisos para acceder a distintas secciones y funcionalidades del sistema.

**Empleo de las Claves de Acceso:**

* **Contraseñas Seguras:** Las contraseñas deben cumplir con criterios de seguridad, como longitud mínima y complejidad (mezcla de letras, números y caracteres especiales).
* **Hashing de Contraseñas:** Las contraseñas son hashadas utilizando un algoritmo seguro (como bcrypt) antes de ser almacenadas en la base de datos. Esto asegura que incluso si la base de datos es comprometida, las contraseñas permanecen protegidas.

**Segmentación de Procesos, Perfiles y Roles:**

* **Perfiles de Usuario:**
  + **Estudiantes:** Acceso a ejercicios y contenido educativo, con funcionalidades limitadas a la visualización y resolución de problemas.
  + **Educadores:** Acceso a funcionalidades de gestión de contenido y monitoreo del progreso de los estudiantes.
  + **Administradores:** Acceso completo al sistema, incluyendo gestión de usuarios, contenido y configuración del sistema.
* **Roles:**
  + **Estudiante:** Puede ver y trabajar con los ejercicios, acceder a resultados de sus actividades.
  + **Docente:** Puede crear y modificar ejercicios, revisar y analizar el rendimiento de los estudiantes.
  + **Administrador:** Puede gestionar todos los aspectos del sistema, incluyendo usuarios, contenido, y configuraciones del sistema.

**Mecanismos de Autenticación:**

* **Autenticación de Dos Factores (2FA):** Opcional para mejorar la seguridad. Requiere que los usuarios proporcionen un segundo factor de autenticación además de su contraseña.
* **Token de Sesión:** Se utilizan tokens de sesión para mantener la sesión activa y validada durante el uso de la aplicación, minimizando el riesgo de secuestro de sesión.

**2. Cifrado de Datos**

**Cifrado de Datos en la Base de Datos:**

* **Datos Sensibles:** Ciertos datos sensibles como contraseñas, datos personales y de contacto pueden requerir cifrado adicional.
* **Algoritmos de Cifrado:**
  + **Contraseñas:** Hashing con bcrypt, que es un algoritmo de hashing seguro y adaptativo diseñado específicamente para almacenar contraseñas de forma segura.
  + **Datos Personales:** Se puede utilizar cifrado simétrico (como AES-256) para proteger datos sensibles en la base de datos. Este algoritmo proporciona un alto nivel de seguridad mediante el uso de una clave de cifrado para cifrar y descifrar los datos.

**Implementación de Cifrado:**

* **Encriptación en Reposo:** Los datos cifrados se almacenan en la base de datos en formato cifrado para proteger la información en caso de acceso no autorizado a la base de datos.
* **Encriptación en Tránsito:** Utilización de HTTPS para cifrar la comunicación entre el servidor y los clientes, asegurando que los datos transmitidos no puedan ser interceptados o modificados.

**Respaldo y Recuperación:**

* **Cifrado de Respaldo:** Los backups de la base de datos también deben estar cifrados para proteger los datos almacenados en copias de seguridad.

1. **Vista de Implementación**

* **Dinosapiens** se basa en una arquitectura modular y escalable, diseñada para permitir una implementación flexible y un mantenimiento eficiente. La estructura general del modelo de implementación incluye los siguientes componentes:
* **Frontend (Interfaz de Usuario):**
  + **Tecnologías:** HTML, CSS, JavaScript, y frameworks como Bootstrap para el diseño responsivo.
  + **Funcionalidades:** Interfaz de usuario interactiva para estudiantes, docentes y administradores. Permite la navegación a través de ejercicios, visualización de resultados, y administración de contenidos.
  + **Componentes Clave:** Páginas de inicio, panel de ejercicios, perfiles de usuario, y paneles de administración.
* **Backend (Lógica del Servidor):**
  + **Tecnologías:** PHP para la lógica de servidor, gestionando solicitudes de los usuarios y la interacción con la base de datos.
  + **Funcionalidades:** Procesamiento de datos, gestión de usuarios, control de acceso, y lógica de aplicación.
  + **Componentes Clave:** Controladores, modelos de datos, y servicios para manejar la lógica del negocio.
* **Base de Datos:**
  + **Tecnologías:** SQL para la gestión de bases de datos relacionales.
  + **Funcionalidades:** Almacenamiento de datos de usuarios, ejercicios, resultados y configuración del sistema.
  + **Componentes Clave:** Tablas para usuarios, ejercicios, resultados y configuraciones.
* **Servicios y API:**
  + **Servicios Internos:** Módulos para funcionalidades específicas como autenticación, generación de reportes, y procesamiento de ejercicios.
  + **API:** (si aplica) Interfaces para integrar con otros sistemas o servicios externos.
* **Seguridad y Autenticación:**
  + **Tecnologías:** Implementación de autenticación y autorización para controlar el acceso y proteger los datos.
  + **Componentes Clave:** Sistemas de autenticación, cifrado de datos y gestión de permisos.
* **Infraestructura:**
  + **Servidor:** Configuración del servidor para hospedar la aplicación, que puede incluir servidores web (como Apache o Nginx) y servidores de base de datos.
  + **Hosting:** Puede estar en servidores locales o en la nube, dependiendo de los requisitos y presupuesto.
* **Descomposición del Sistema**
* **Dinosapiens** se descompone en varios módulos y componentes que interactúan entre sí. La descomposición del sistema se presenta de la siguiente manera:
* **Módulo de Autenticación y Gestión de Usuarios:**
  + **Registro y Login:** Maneja la creación de cuentas y el inicio de sesión.
  + **Gestión de Perfiles:** Permite a los usuarios actualizar su información y preferencias.
  + **Control de Acceso:** Implementa la segmentación de roles y permisos.
* **Módulo de Ejercicios y Contenidos:**
  + **Creación y Edición:** Permite a los docentes crear y modificar ejercicios.
  + **Visualización y Resolución:** Facilita a los estudiantes acceder a y trabajar en los ejercicios.
  + **Evaluación:** Calcula y muestra los resultados y retroalimentación a los estudiantes.
* **Módulo de Reportes y Análisis:**
  + **Generación de Reportes:** Ofrece a los docentes informes sobre el desempeño de los estudiantes.
  + **Análisis de Datos:** Proporciona herramientas para analizar el progreso y los resultados de los ejercicios.
* **Módulo de Configuración del Sistema:**
  + **Ajustes Generales:** Permite a los administradores configurar parámetros del sistema y gestionar usuarios.
  + **Gestión de Contenidos:** Administra la configuración de ejercicios y contenido educativo.
* **Módulo de Seguridad:**
  + **Cifrado de Datos:** Asegura que los datos sensibles estén protegidos.
  + **Auditoría y Monitoreo:** Registra y analiza eventos y accesos para identificar posibles problemas de seguridad.
* **Módulo de Integración:**
  + **API Externas:** Si se requieren integraciones con otros sistemas o servicios, este módulo se encarga de la comunicación y el intercambio de datos.
* **Diagrama de Arquitectura**
* Un diagrama de arquitectura puede ayudar a visualizar cómo estos componentes interactúan:
* **Frontend:** Se comunica con el **Backend** a través de solicitudes HTTP.
* **Backend:** Interactúa con la **Base de Datos** para recuperar y almacenar datos.
* **Servicios y API:** Facilitan la integración con sistemas externos si es necesario.
* **Seguridad:** Protege la comunicación entre el **Frontend** y el **Backend**, y asegura la **Base de Datos**.
  1. **Herramientas de Desarrollo e implementación**

**PHP**

* PHP es un lenguaje de programación del lado del servidor que se utiliza para desarrollar aplicaciones web dinámicas. Es ideal para manejar la lógica de la aplicación y la interacción con la base de datos, permitiendo generar contenido dinámico en las páginas web.

**C++**

* C++ es un lenguaje de programación de alto rendimiento, utilizado en Dinosapiens para manejar funciones o módulos que requieren mayor eficiencia o control sobre los recursos del sistema. Es conocido por ser versátil y eficiente en aplicaciones que demandan procesamiento intensivo.

**Bootstrap**

* Bootstrap es un framework de diseño web que facilita la creación de interfaces responsivas y modernas. Se utiliza en el frontend de Dinosapiens para garantizar que la aplicación se vea bien, sin necesidad de escribir mucho código CSS.

**SQL**

* SQL es el lenguaje estándar para interactuar con bases de datos relacionales. En Dinosapiens, SQL se utilizó para gestionar la base de datos, permitiendo almacenar y consultar datos, como las lecciones o el progreso de los estudiantes, de manera estructurada y eficiente.

**SweetAlert2**

* Una librería de JavaScript para mostrar alertas y diálogos de confirmación con un diseño bonito.

**Sass**

* Sass es un preprocesador CSS, es una herramienta que nos permite generar, de manera automática, hojas de estilo, añadiéndoles características que no tiene CSS, y que son propias de los lenguajes de programación, como pueden ser variables, funciones, selectores anidados, herencia, etcétera.

**Css**

* Las hojas de estilo en cascada permiten crear páginas web atractivas.
  1. **Paquetes/Componentes**
* **Frontend:**
* **Interacción:** Envía solicitudes HTTP al Backend para operaciones como la visualización de contenido o el envío de respuestas.
* **Backend:**
* **Componentes:**
* **API de Integración:** Maneja las solicitudes externas y la comunicación con otros sistemas.
* **Lógica de Negocio:** Implementa la lógica central del sistema, incluyendo la gestión de usuarios y el procesamiento de ejercicios.
* **Servicios Internos:** Proporciona funcionalidades adicionales como la generación de reportes y el procesamiento de datos.
* **Autenticación:** Controla la autenticación y autorización de usuarios.
* **Base de Datos:**
* **Interacción:** El Backend realiza consultas SQL a la Base de Datos para almacenar y recuperar datos relacionados con usuarios, ejercicios y resultados.
* Despliegue
* **Descripción General**
* Dinosapiens será desplegado en una infraestructura que consta de varios nodos físicos, cada uno con funciones específicas. La configuración debe garantizar el rendimiento, la disponibilidad y la escalabilidad del sistema. La plataforma se divide en nodos que incluyen servidores de aplicaciones, servidores de bases de datos y estaciones de trabajo para usuarios finales.
* **Nodos Físicos y sus Interconexiones**
* **Servidor de Aplicaciones**
* **Hardware:**
* Procesador: Intel Xeon o equivalente (mínimo 4 núcleos)
* Memoria: 16 GB RAM
* Almacenamiento: 500 GB SSD
* **Conexiones:**
* Conectado a la red LAN local (topología estrella)
* Red de comunicaciones: Ethernet Gigabit
* **Servidor de Base de Datos**
* **Hardware:**
* Procesador: Intel Xeon o equivalente (mínimo 8 núcleos)
* Memoria: 32 GB RAM
* Almacenamiento: 1 TB SSD en RAID 1 para redundancia
* **Conexiones:**
* Conectado a la red LAN local (topología estrella)
* Red de comunicaciones: Ethernet Gigabit
* **Estaciones de Trabajo (Clientes)**
* **Hardware:**
* Procesador: Intel Core i5 o equivalente
* Memoria: 8 GB RAM
* Almacenamiento: 256 GB SSD
* **Conexiones:**
* Conectadas a la red LAN local
* Red de comunicaciones: Ethernet Gigabit o Wi-Fi (según la configuración)
* **Vista de Despliegue**
* Para visualizar la configuración y las interconexiones, se puede usar un diagrama de despliegue en UML. Aquí te proporciono una descripción general de cómo se puede estructurar:
* **Servidor de Aplicaciones**
* Representado como un nodo en el diagrama de despliegue.
* Conectado al Servidor de Base de Datos y a las Estaciones de Trabajo.
* **Servidor de Base de Datos**
* Representado como un nodo separado.
* Conectado al Servidor de Aplicaciones.
* **Estaciones de Trabajo (Clientes)**
* Representadas como nodos que se conectan al Servidor de Aplicaciones a través de la red LAN.
* **Interconexiones:**
* **Servidor de Aplicaciones** se conecta al **Servidor de Base de Datos** a través de una conexión de red LAN.
* **Estaciones de Trabajo** se conectan al **Servidor de Aplicaciones** a través de la red LAN.
* **Mapeo entre Procesos y Nodos Físicos**
* **Procesos en el Servidor de Aplicaciones:**
* Procesos de manejo de solicitudes de usuarios.
* Ejecución de lógica de negocio y aplicaciones web.
* **Procesos en el Servidor de Base de Datos:**
* Procesos de gestión y consulta de la base de datos.
* Ejecución de transacciones y almacenamiento de datos.
  1. Vista de Administración

La administración del software Dinosapiens se organiza en torno a dos roles principales: Administrador y Explorador (estudiantes). Cada rol tiene distintas capacidades y opciones de administración que facilitan la gestión del sistema y el uso por parte de los estudiantes.

**1. Rol de Administrador**

**Opciones de Administración:**

* **Iniciar Sesión:** El Administrador puede iniciar sesión en el sistema utilizando sus credenciales (usuario y contraseña). Este proceso autentica al usuario y le otorga acceso a las funciones administrativas.
* **Cerrar Sesión:** Después de finalizar las tareas administrativas, el Administrador puede cerrar sesión para asegurar que la sesión de administración se termine correctamente y que no se deje abierta por descuido.
* **Eliminar Perfiles:** El Administrador tiene la capacidad de eliminar perfiles de usuario del sistema. Esta opción es útil para gestionar la base de datos de usuarios, eliminando perfiles inactivos o incorrectos.
* **Gestión de Usuarios:** Además de eliminar perfiles, el Administrador puede tener opciones para agregar nuevos usuarios, modificar datos de perfiles existentes y asignar roles o permisos adicionales según sea necesario.
* **Visualización de Estadísticas y Reportes:** El Administrador puede acceder a reportes y estadísticas del sistema, como el número de usuarios activos, desempeño de los ejercicios prácticos, y otros datos relevantes para la gestión del sistema.
* **Configuración del Sistema:** El Administrador puede ajustar configuraciones del sistema, como parámetros de seguridad, opciones de personalización de la interfaz, y ajustes relacionados con la operatividad general del software.

**2. Rol de Explorador (Estudiantes)**

**Opciones de Administración:**

* **Iniciar Sesión:** Los Exploradores (estudiantes) inician sesión para acceder a los ejercicios prácticos y funcionalidades educativas del sistema. La autenticación asegura que cada estudiante pueda acceder solo a su perfil y datos asociados.
* **Cerrar Sesión:** Los Exploradores cierran sesión al finalizar sus sesiones de práctica para proteger su información personal y garantizar que no permanezcan conectados innecesariamente.
* **Acceso a Ejercicios Prácticos:** Los Exploradores tienen acceso a una variedad de ejercicios y materiales educativos diseñados para reforzar los conocimientos adquiridos en el colegio.
* **Visualización de Progreso:** Los Exploradores pueden ver su progreso en los ejercicios, incluyendo las calificaciones obtenidas y las áreas donde necesitan mejorar.
* **Actualización de Perfil (Opcional):** Dependiendo de la configuración del sistema, los Exploradores pueden tener opciones limitadas para actualizar su información personal, como el correo electrónico o la contraseña.

Describe las distintas opciones de la administración del software.