**CARRERA DE INGENIERÍA EN SOFTWARE**

Un conjunto de letras blancas en un fondo blanco

Descripción generada automáticamente con confianza media

**APLICACIONES WEB**

**Autores:**

Muñiz Rivas Leopoldo Miquel

**Tema:**

**Control de asistencia para docentes de la uleam**

**Curso:**

4 - A

**Periodo 2025(1)**

INDICE

[1. Resumen 1](#_Toc199271175)

[1.1. Finalidad del sistema y características principales 1](#_Toc199271176)

[1.2. Función básica 1](#_Toc199271177)

[1.3. Resumen de la arquitectura del sistema 1](#_Toc199271178)

[1.4. Patrón de Arquitectura de Componente 2](#_Toc199271179)

[1.5. Usuarios y casos de uso de destino 3](#_Toc199271180)

[1.6. Casos de uso primario 3](#_Toc199271181)

[1.7. Apiladora de la tecnología 3](#_Toc199271182)

[1.8. Principios clave del diseño 4](#_Toc199271183)

[2. Arquitectura de sistemas 4](#_Toc199271184)

[2.1. Finalidad y alcance 4](#_Toc199271185)

[2.2. Arquitectura general del sistema 5](#_Toc199271186)

[2.3. Diagrama de capa del sistema 5](#_Toc199271187)

[2.1.3 Patrón de Arquitectura de Componente 5](#_Toc199271188)

[2.2.3 Patrón de Estructura de Componentes 5](#_Toc199271189)

[2.3.3 Ejemplos de ejecución de componentes 6](#_Toc199271190)

[2.4. Arquitectura de flujo de datos 6](#_Toc199271191)

[2.1.4 Patrón de acceso a los datos 6](#_Toc199271192)

[2.5. Interacciones básicas del sistema 7](#_Toc199271193)

[2.1.5 Asistencia Gestión de asistencia Flujo 7](#_Toc199271194)

[2.6. Flujo del sistema de navegación 8](#_Toc199271195)

[2.7. Principios arquitectónicos clave 8](#_Toc199271196)

[2.1.7 Componente Independencia 8](#_Toc199271197)

[2.2.7 Centralización de la gestión de datos 8](#_Toc199271198)

[2.8. Evento-Driven Arquitectura 9](#_Toc199271199)

[3. Gestión de asistencia 10](#_Toc199271200)

[3.1. Finalidad y alcance 10](#_Toc199271201)

[3.2. Resumen del sistema 10](#_Toc199271202)

[3.3. Añadir registros de asistencia 11](#_Toc199271203)

[3.1.3 Añadir el componente de Asistencia 11](#_Toc199271204)

[3.2.3 Asistencia Grabación Lógica Flujo 11](#_Toc199271205)

[3.1.3 Funciones clave y operaciones de datos 12](#_Toc199271206)

[3.4. Mostrando registros de asistencia 12](#_Toc199271207)

[3.1.4 Componente de visualización 12](#_Toc199271208)

[3.1.4 Estructura de plantilla y vinculación de datos 13](#_Toc199271209)

[3.1.4 Generación y actualización de servicios públicos 13](#_Toc199271210)

[3.5. Flujo e integración de datos 14](#_Toc199271211)

[3.1.5 Integración con Capa de Datos 14](#_Toc199271212)

[3.1.5 Coordinación de Refresca de IU 15](#_Toc199271213)

[3.6. Componentes de IU y Estilo 16](#_Toc199271214)

[3.1.6 Pantalla con tarjeta 16](#_Toc199271215)

[3.2.6 Añadir botón interactivo 16](#_Toc199271216)

[4. Gestión de los Horarios 16](#_Toc199271217)

[4.1. Finalidad del sistema y alcance 17](#_Toc199271218)

[4.2. Arquitectura de componentes 17](#_Toc199271219)

[4.3. Estructura de datos y plantilla de refuerzo 18](#_Toc199271220)

[4.1.3 Cuadro de datos de calendario 18](#_Toc199271221)

[4.2.3 Implementación de la función de plantilla 18](#_Toc199271222)

[4.4. Generación y gestión de calendarios 18](#_Toc199271223)

[4.1.4 Funciones básicas 18](#_Toc199271224)

[4.5. Diseño visual y diseño 19](#_Toc199271225)

[4.1.5 Estructura de la tarjeta de calendario 19](#_Toc199271226)

[4.6. Especificaciones de Estilo CSS 20](#_Toc199271227)

[4.7. Conducta sensible 20](#_Toc199271228)

[4.8. Integración con Capa de Datos 21](#_Toc199271229)

[4.1.8 Patrón de acceso a los datos 21](#_Toc199271230)

[4.9. DOM Integración y Gestión de contenedores 21](#_Toc199271231)

[4.1.9 Elementos de contenedores 21](#_Toc199271232)

[4.2.9 Cargando del módulo 22](#_Toc199271233)

[4.10. Relación con el sistema de asistencia 22](#_Toc199271234)

[5. Sistema de autenticación 22](#_Toc199271235)

[5.1. Finalidad y alcance 22](#_Toc199271236)

[5.2. Autación Flujo 23](#_Toc199271237)

[5.3. Login de arquitectura de componentes 23](#_Toc199271238)

[5.4. Forma la generación de campo 24](#_Toc199271239)

[5.5. Proceso de autenticación 24](#_Toc199271240)

[5.6. Arquitectura de componentes de sesión 25](#_Toc199271241)

[5.7. Gestión estatal de Button 26](#_Toc199271242)

[5.8. Integración con Capa de Datos 26](#_Toc199271243)

[5.9. Integración de lanzamiento 27](#_Toc199271244)

[6. Perfil del profesor 27](#_Toc199271245)

[6.1. Finalidad y alcance 27](#_Toc199271246)

[6.2. Reseña del componente 27](#_Toc199271247)

[6.3. Estructura de componentes 28](#_Toc199271248)

[6.4. Flujo e integración de datos 28](#_Toc199271249)

[6.1.4 Proceso de recuperación de datos 29](#_Toc199271250)

# Resumen

## Finalidad del sistema y características principales

El sistema de gestión de la asistencia al profesorado es una aplicación web del lado del cliente que permite a las instituciones educativas rastrear digitalmente la asistencia al profesorado y gestionar los horarios de enseñanza. El sistema proporciona una interfaz en español para registrar las entradas de asistencia, ver datos históricos de asistencia y gestionar los horarios de enseñanza.

## Función básica

|  |  |  |
| --- | --- | --- |
| Área de características | Descripción | Componentes primarios |
| Registro de asistencia | Los profesores pueden entrar y salir, con seguimiento automático de los horarios de entrada y salida | add-asistencia, asistencia |
| Gestión de los cuadros | Pantalla y gestión de los horarios docentes con integración al seguimiento de asistencia | horarios |
| Autenticación del usuario | Distribuye funcionalidad de inicio de sesión y logout para profesores | form-login, logout-btn |
| Gestión del perfil | Mostración de perfil docente y gestión de la información | Docente |
| Navegación | Enrutamiento de aplicaciones y navegación entre diferentes áreas funcionales | nav, sistema de enrutamiento |

## Resumen de la arquitectura del sistema

El siguiente diagrama ilustra la arquitectura de alto nivel y cómo las entidades de código mapean las áreas funcionales:

**Arquitectura de sistemas de alto nivel**

A black screen with white text

AI-generated content may be incorrect.

## Patrón de Arquitectura de Componente

El sistema sigue una arquitectura consistente basada en componentes donde cada área funcional se implementa como un módulo autónomo con archivos dedicados HTML, CSS y JavaScript.

**Estructura de componentes y flujo de datos**



## Usuarios y casos de uso de destino

El sistema está diseñado para la ULEAM, con la que se dirige específicamente a:

* **Maestros:** Usuarios de primaria que registran su asistencia diaria y ven sus horarios

## Casos de uso primario

1. **Grabación diaria de asistencia:** Maestros usan el add-asistenciacomponente para atracar al inicio de su jornada laboral y desgastado al salir
2. **Historia de asistencia Revisión** : Maestros y administradores pueden ver registros de asistencia histórica a través de la asistenciacomponentes
3. **Gestión del horario:** Los usuarios pueden ver los horarios de enseñanza y los próximos compromisos a través de la horarioscomponentes
4. **Gestión de sesiones:** La funcionalidad segura de inicio de sesión y logout garantiza un control de acceso adecuado

## Apiladora de la tecnología

El sistema se implementa como una aplicación web del lado del cliente utilizando:

* **Frontend** : HTML5, CSS3, y vainilla JavaScript con módulos ES6
* **Arquitectura** : Arquitectura basada en componentes con generación de IU impulsada por plantillas
* **Gestión** de **datos:** Almacenamiento y gestión de datos del lado del cliente a través de un Dataclase
* **Ruta** : Sistema de enrutamiento personalizado del lado del cliente (useRoutes)
* **Plantillas** : Elementos de plantilla HTML para la generación de contenido dinámico (trabajación, tarjeta de tarea)

## Principios clave del diseño

|  |  |  |
| --- | --- | --- |
| Principio | Aplicación | Ejemplo |
| Aislamiento del componente | Cada área de características tiene archivos HTML, CSS y JS dedicados | [componentes/add-asistencia/add-asistencia.html1](https://github.com/lesquel/proyecto-asistencia-docentes/blob/63ad56c3/components/add-asistencia/add-asistencia.html#L1-L1) [componentes/asistencia/asistencia.html1](https://github.com/lesquel/proyecto-asistencia-docentes/blob/63ad56c3/components/asistencia/asistencia.html#L1-L1) |
| IDT de plantilla-diverso | Generación de contenido dinámico con plantillas HTML | [componentes/asistencia/asistencia.html5 a 16](https://github.com/lesquel/proyecto-asistencia-docentes/blob/63ad56c3/components/asistencia/asistencia.html#L5-L16) [componentes/horarios/horarios.html2 a 12](https://github.com/lesquel/proyecto-asistencia-docentes/blob/63ad56c3/components/horarios/horarios.html#L2-L12) |
| Gestión centralizada de los datos | Unico Dataclase maneja todas las operaciones de datos | Referenciada en funciones de utilidad en los componentes |
| JavaScript modular | Módulos ES6 con clara separación de preocupaciones | [componentes/asistencia/asistencia.html21-22](https://github.com/lesquel/proyecto-asistencia-docentes/blob/63ad56c3/components/asistencia/asistencia.html#L21-L22) [componentes/horarios/horarios.html17 a 18](https://github.com/lesquel/proyecto-asistencia-docentes/blob/63ad56c3/components/horarios/horarios.html#L17-L18) |

# Arquitectura de sistemas

## Finalidad y alcance

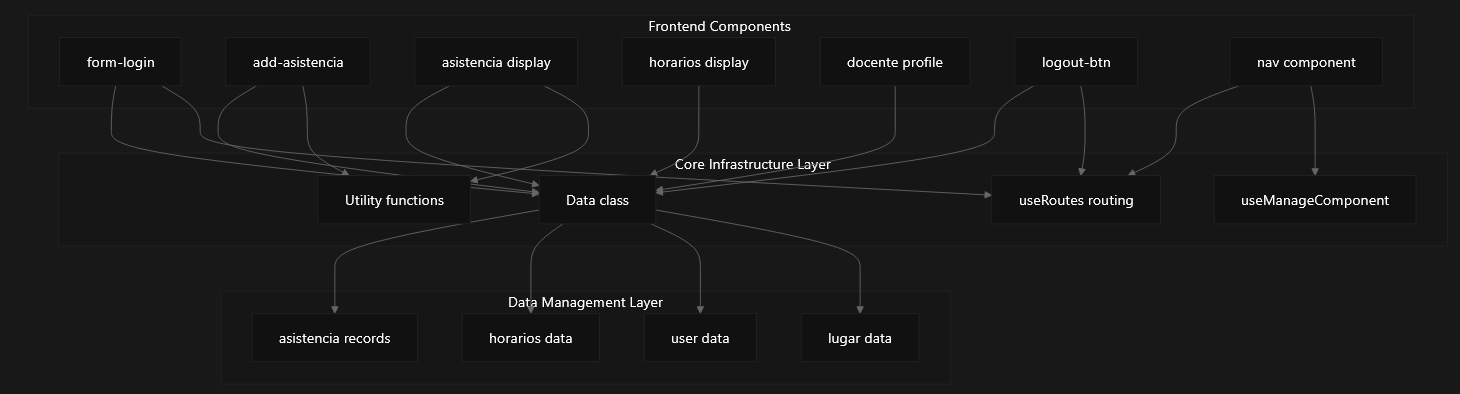
La arquitectura general del sistema del sistema de gestión de la asistencia al profesorado, incluyendo organización de componentes, patrones de flujo de datos e interacciones del sistema. Abarca el diseño estructural de alto nivel y los patrones arquitectónicos utilizados en toda la base de código.

Para obtener información detallada sobre los componentes funcionales específicos, consulte Componentes Core. Para las especificaciones del componente de IU, consulte Componentes de UI. Para la configuración del medio ambiente de desarrollo, vea Development Setup.

## Arquitectura general del sistema

El sistema sigue una arquitectura basada en componentes con tres capas primarias: Componentes Frontend, Infraestructura Básica y Gestión de Datos.

## Diagrama de capa del sistema

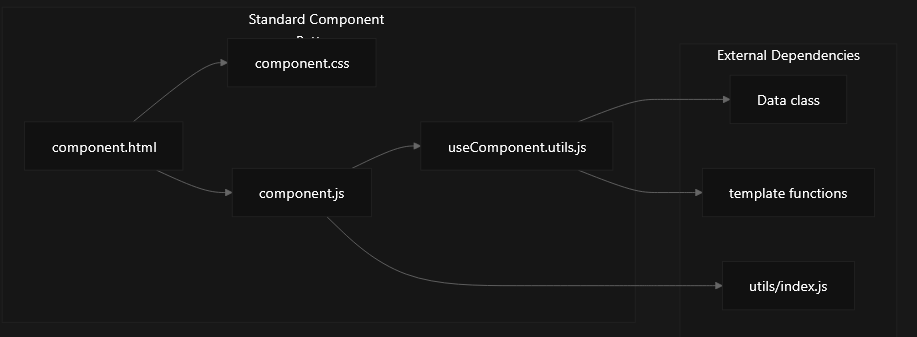


Fuentes: [componentes/add-asistencia/add-asistencia.js1 a 8](https://github.com/lesquel/proyecto-asistencia-docentes/blob/63ad56c3/components/add-asistencia/add-asistencia.js#L1-L8) [componentes/asistencia/useAsistencia.utils.js1 a 4](https://github.com/lesquel/proyecto-asistencia-docentes/blob/63ad56c3/components/asistencia/useAsistencia.utils.js#L1-L4) [componentes/horarios/useHorarios.utils.js1 a 4](https://github.com/lesquel/proyecto-asistencia-docentes/blob/63ad56c3/components/horarios/useHorarios.utils.js#L1-L4) [componentes/nav/nav.js1 a 3](https://github.com/lesquel/proyecto-asistencia-docentes/blob/63ad56c3/components/nav/nav.js#L1-L3)

### Patrón de Arquitectura de Componente

Cada componente del sistema sigue un patrón estructural consistente con la separación de preocupaciones entre la presentación, el comportamiento y la lógica empresarial.

### Patrón de Estructura de Componentes



### Ejemplos de ejecución de componentes

A screenshot of a computer

AI-generated content may be incorrect.

Fuentes: [componentes/add-asistencia/add-asistencia.js10 a 16](https://github.com/lesquel/proyecto-asistencia-docentes/blob/63ad56c3/components/add-asistencia/add-asistencia.js#L10-L16) [componentes/asistencia/useAsistencia.utils.js3](https://github.com/lesquel/proyecto-asistencia-docentes/blob/63ad56c3/components/asistencia/useAsistencia.utils.js#L3-L3) [componentes/horarios/useHorarios.utils.js3](https://github.com/lesquel/proyecto-asistencia-docentes/blob/63ad56c3/components/horarios/useHorarios.utils.js#L3-L3)

## Arquitectura de flujo de datos

El sistema aplica un patrón centralizado de gestión de datos en el que todos los componentes interactúan a través de la Dataclase.

### Patrón de acceso a los datos

A black screen with white text

AI-generated content may be incorrect.

Fuentes: [componentes/add-asistencia/add-asistencia.js17 a 46](https://github.com/lesquel/proyecto-asistencia-docentes/blob/63ad56c3/components/add-asistencia/add-asistencia.js#L17-L46) [componentes/asistencia/useAsistencia.utils.js7 a 26](https://github.com/lesquel/proyecto-asistencia-docentes/blob/63ad56c3/components/asistencia/useAsistencia.utils.js#L7-L26) [componentes/horarios/useHorarios.utils.js5 a 11](https://github.com/lesquel/proyecto-asistencia-docentes/blob/63ad56c3/components/horarios/useHorarios.utils.js#L5-L11)

## Interacciones básicas del sistema

La funcionalidad central del sistema gira en torno a la gestión de asistencia con patrones de interacción específicos entre componentes.

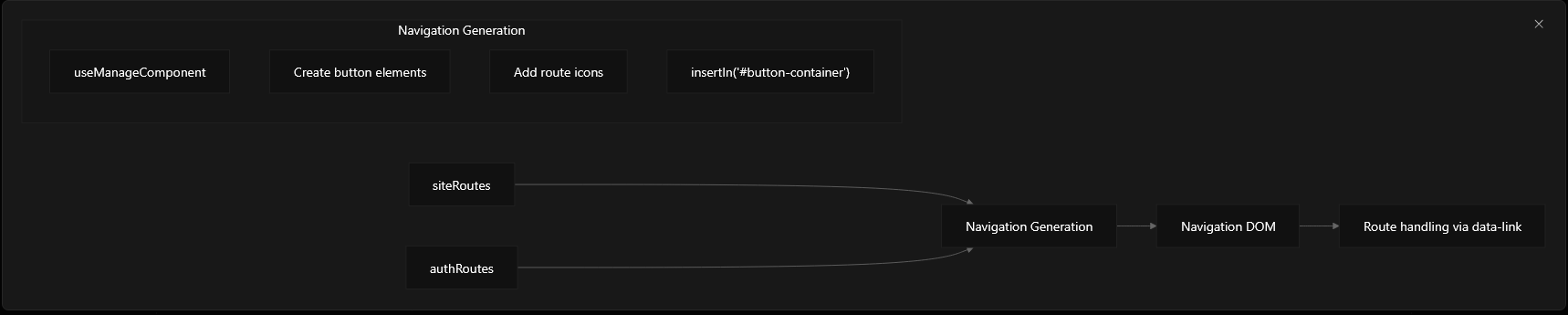
### Asistencia Gestión de asistencia Flujo

A screenshot of a computer

AI-generated content may be incorrect.

Fuentes: [componentes/add-asistencia/add-asistencia.js19 a 52](https://github.com/lesquel/proyecto-asistencia-docentes/blob/63ad56c3/components/add-asistencia/add-asistencia.js#L19-L52)

## Flujo del sistema de navegación



Fuentes: [componentes/nav/nav.js5 - 66](https://github.com/lesquel/proyecto-asistencia-docentes/blob/63ad56c3/components/nav/nav.js#L5-L66)

## Principios arquitectónicos clave

### Componente Independencia

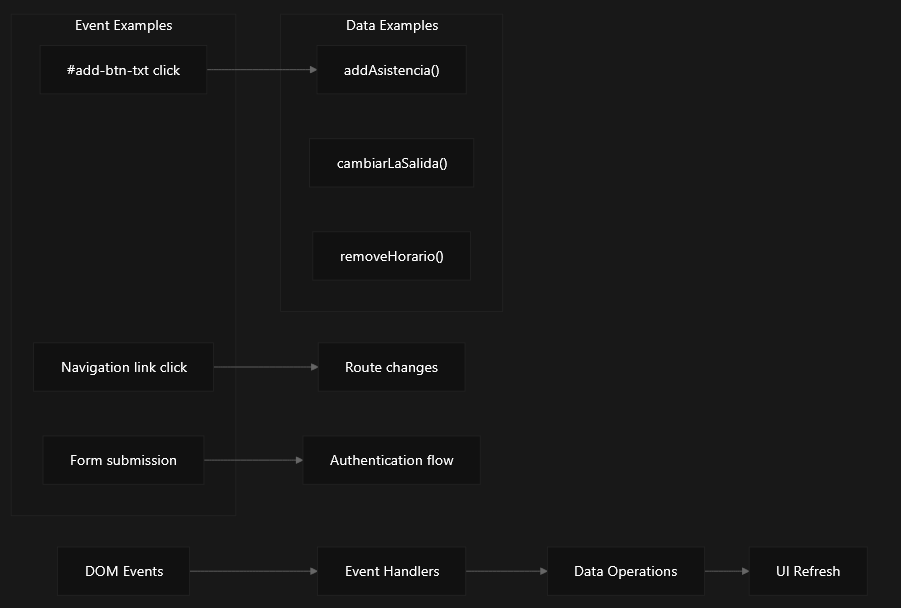
|  |  |  |
| --- | --- | --- |
| Principio | Aplicación | Ejemplo |
| **Separación de las preocupaciones** | Cada componente tiene archivos HTML, CSS y JS distintos | add-asistencia.html, add-asistencia.css, add-asistencia.js |
| **Separación de utilidad** | Lógica de negocio dividida en useComponent.utils.jsArchivos | useAsistencia.utils.js, useHorarios.utils.js |
| **Renderización basada en la planilla** | Generación HTML a través de funciones de plantilla | templateAsistencia(), templateHorarios() |

### Centralización de la gestión de datos

|  |  |  |
| --- | --- | --- |
| Patrón | Aplicación | Referencia del código |
| **Fuente de datos única** | Todos los componentes utilizan una sola Datainstancia de clase | const data = new Data() |
| **Acceso de datos consistente** | Nombre de método estandarizado para operaciones de la CRUD | findAsistencias(), addAsistencia(), removeHorario() |
| **Actualizaciones de IU reactivas** | Regeneración de IU tras cambios de datos | removeAsitencia()seguidos por generateAsitencia() |

## Evento-Driven Arquitectura

El sistema utiliza manejadores de eventos DOM para desencadenar la lógica del negocio, manteniendo una clara separación entre las interacciones de los usuarios y las operaciones de datos.



Fuentes: [componentes/add-asistencia/add-asistencia.js19](https://github.com/lesquel/proyecto-asistencia-docentes/blob/63ad56c3/components/add-asistencia/add-asistencia.js#L19-L19)

[componentes/nav/nav.jsA 23](https://github.com/lesquel/proyecto-asistencia-docentes/blob/63ad56c3/components/nav/nav.js#L22-L23)

Esta arquitectura proporciona una separación sostenible de preocupaciones, al tiempo que permite un flujo de datos eficiente y un manejo de la interacción de los usuarios a lo largo del sistema de gestión de la asistencia al profesorado.

# Gestión de asistencia

## Finalidad y alcance

La funcionalidad de grabación y visualización de asistencia dentro del sistema de gestión de asistencia al profesorado. El subsistema de gestión de asistencia se encarga de la creación de nuevas entradas de asistencia, actualizando los tiempos de salida para las entradas existentes y mostrando registros de asistencia a los usuarios a través de una interfaz basada en tarjetas.

Para obtener información sobre la gestión de horarios y su relación con la asistencia, consulte Gestión de Listas. Para más detalles sobre la autenticación del usuario y la gestión de sesiones, consulte Sistema de autenticación.

## Resumen del sistema

El subsistema de gestión de asistencia consta de dos componentes principales que trabajan juntos para proporcionar funcionalidad completa de seguimiento de asistencia:

A screenshot of a computer

AI-generated content may be incorrect.

Fuentes: [componentes/add-asistencia/add-asistencia.js1-52](https://github.com/lesquel/proyecto-asistencia-docentes/blob/63ad56c3/components/add-asistencia/add-asistencia.js#L1-L52) [componentes/asistencia/useAsistencia.utils.js1 a 31](https://github.com/lesquel/proyecto-asistencia-docentes/blob/63ad56c3/components/asistencia/useAsistencia.utils.js#L1-L31) [componentes/asistencia/asistencia.js1-50](https://github.com/lesquel/proyecto-asistencia-docentes/blob/63ad56c3/components/asistencia/asistencia.js#L1-L50)

## Añadir registros de asistencia

### Añadir el componente de Asistencia

El add-asistenciacomponente proporciona la interfaz principal para registrar entradas y salidas de asistencia. El componente cuenta con un botón circular que se expande en flota para mostrar el texto de acción.

|  |  |
| --- | --- |
| Archivo de componentes | Objeto |
| add-asistencia.html | Estructura de botones y diseño |
| add-asistencia.css | Estilo de botón animado con efectos flotantes |
| add-asistencia.js | Lógica de grabación de asistencia básica |

### Asistencia Grabación Lógica Flujo

A screenshot of a computer

AI-generated content may be incorrect.

Fuentes: [componentes/add-asistencia/add-asistencia.js19 a 52](https://github.com/lesquel/proyecto-asistencia-docentes/blob/63ad56c3/components/add-asistencia/add-asistencia.js#L19-L52)

### Funciones clave y operaciones de datos

El proceso de registro de asistencia utiliza varias funciones de utilidad y operaciones de datos:

|  |  |  |
| --- | --- | --- |
| Función | Objeto | Ubicación |
| obtenerLaprimeraAsistencia() | Obtiene la primera entrada de asistencia incompleta | Método de la clase de datos |
| cambiarLaSalida(id, salidaData) | Actualiza el tiempo de salida para la entrada existente | Método de la clase de datos |
| addAsistencia(attendanceData) | Crea nuevo récord de asistencia | Método de la clase de datos |
| removeHorario(index) | Elimina la franja horaria tras la creación de asistencia | Método de la clase de datos |
| obtenerRadomTipoEntrada() | Genera tipo de entrada aleatorio | Función de utilidad |
| obtenerRadomTipoSalida() | Genera tipo de salida aleatorio | Función de utilidad |
| obtenerHoraActual() | Consigue el tiempo actual | Función de utilidad |

Fuentes: [componentes/add-asistencia/add-asistencia.js1 a 8](https://github.com/lesquel/proyecto-asistencia-docentes/blob/63ad56c3/components/add-asistencia/add-asistencia.js#L1-L8) [componentes/add-asistencia/add-asistencia.js20 a 1945](https://github.com/lesquel/proyecto-asistencia-docentes/blob/63ad56c3/components/add-asistencia/add-asistencia.js#L20-L45)

## Mostrando registros de asistencia

### Componente de visualización

El asistenciaEl componente realiza registros de asistencia utilizando un diseño basado en tarjetas con un sistema de plantillas. Cada récord de asistencia se muestra como una tarjeta de estilo que muestra información de entrada y salida.

A black screen with white text

AI-generated content may be incorrect.

Fuentes: [componentes/asistencia/asistencia.html5 a 16](https://github.com/lesquel/proyecto-asistencia-docentes/blob/63ad56c3/components/asistencia/asistencia.html#L5-L16) [componentes/asistencia/asistencia.js4 a 49](https://github.com/lesquel/proyecto-asistencia-docentes/blob/63ad56c3/components/asistencia/asistencia.js#L4-L49)

### Estructura de plantilla y vinculación de datos

El templateAsistenciafunción crea tarjetas de asistencia mediante la clonación de la plantilla HTML y poblarla con datos:

|  |  |  |
| --- | --- | --- |
| Elemento de la plantilla | Fuente de datos | Descripción |
| .workout-title | lugar.nombre + direccion | Nombre de la ubicación y dirección |
| .workout-time | asistencia.fecha | Fecha de asistencia |
| .workout-list | Elementos de entrada y salida | Contenedor para detalles de asistencia |
| .workout-item(entrada) | entrada.tipoEntrada + horaEntrada | Tipo de entrada y tiempo |
| .workout-item(salida) | salida.tipoSalida + horaSalida | Tipo y hora de salida |

Fuentes: [componentes/asistencia/asistencia.js9 a 1945](https://github.com/lesquel/proyecto-asistencia-docentes/blob/63ad56c3/components/asistencia/asistencia.js#L9-L45)

### Generación y actualización de servicios públicos

El useAsistencia.utils.jsfile proporciona funciones para la gestión del ciclo de vida de visualización de asistencia:



Fuentes: [componentes/asistencia/useAsistencia.utils.js7 a 30](https://github.com/lesquel/proyecto-asistencia-docentes/blob/63ad56c3/components/asistencia/useAsistencia.utils.js#L7-L30)

## Flujo e integración de datos

### Integración con Capa de Datos

El sistema de gestión de la asistencia se integra estrechamente con el Dataclase para todas las operaciones de persistencia:

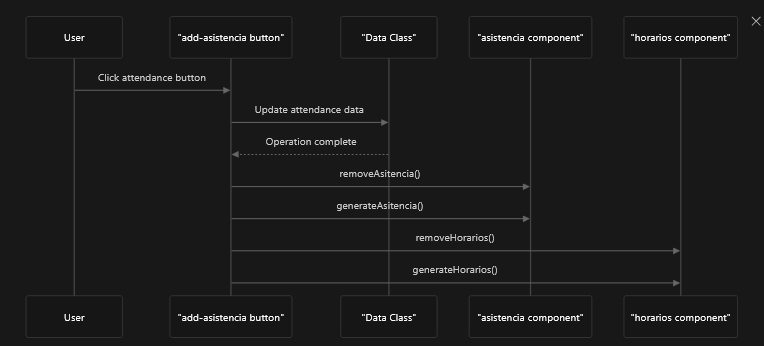
|  |  |  |
| --- | --- | --- |
| Método de datos | Usage in Asistencia | Parámetros |
| getUser() | Obtenga el usuario actual para los registros de asistencia | Ninguno |
| findAsistencias(user) | Recupera los registros de asistencia del usuario | Objeto de usuario |
| findLugarById(id) | Obtenga detalles de ubicación para la visualización | Identificación de ubicación |
| addAsistencia(data) | Crear nueva entrada de asistencia | Objeto de asistencia |
| cambiarLaSalida(id, salida) | Actualizar la información de salida | ID, objeto de datos de salida |
| obtenerLaprimeraAsistencia() | Comprobar las entradas incompletas | Ninguno |
| removeHorario(index) | Eliminar programar ranura después de la entrada | Indice de la lista |

Fuentes:

[componentes/add-asistencia/add-asistencia.js17 a 46](https://github.com/lesquel/proyecto-asistencia-docentes/blob/63ad56c3/components/add-asistencia/add-asistencia.js#L17-L46) [componentes/asistencia/useAsistencia.utils.js4 a 25](https://github.com/lesquel/proyecto-asistencia-docentes/blob/63ad56c3/components/asistencia/useAsistencia.utils.js#L4-L25)

### Coordinación de Refresca de IU

Las operaciones de asistencia desencadenan actualizaciones coordinadas de la interfaz de usuario en múltiples componentes:



Fuentes: [componentes/add-asistencia/add-asistencia.js48 - 51](https://github.com/lesquel/proyecto-asistencia-docentes/blob/63ad56c3/components/add-asistencia/add-asistencia.js#L48-L51)

## Componentes de IU y Estilo

### Pantalla con tarjeta

La pantalla de asistencia utiliza un diseño basado en tarjetas con las siguientes clases de CSS:

|  |  |
| --- | --- |
| Clase CSS | Objeto |
| .workout-card | Contensión de tarjeta principal con tamaño sensible |
| .workout-bg | Semi-transparente superposición de fondo |
| .workout-title | Mostrar nombre de ubicación |
| .workout-time | Fecha de visualización posicionada de arriba-derecha |
| .workout-icon-wrapper | Contenedor de icono con pantalla de emoji |
| .workout-list | Contenedor para artículos de entrada/salida |
| .workout-item | Registro individual de entrada o salida |
| .workout-dot | Indicador circular pequeño |
| .workout-text | Contenido de texto para los detalles de entrada/salida |

Fuentes:

[componentes/asistencia/asistencia.css1-144](https://github.com/lesquel/proyecto-asistencia-docentes/blob/63ad56c3/components/asistencia/asistencia.css#L1-L144)

### Añadir botón interactivo

El botón de asistencia de añadir características de efectos de flotante animados:

|  |  |
| --- | --- |
| Clase CSS | Efecto |
| .icon-btn | Base de estilo de botones circulares |
| .add-btn:hover | Amplía ancho para mostrar texto |
| .btn-txt | Texto oculto que aparece en flotante |
| .add-icon::before, .add-icon::after | Crea icono de "o" con CSS |

Fuentes:

[componentes/add-asistencia/add-asistencia.css14 - 111](https://github.com/lesquel/proyecto-asistencia-docentes/blob/63ad56c3/components/add-asistencia/add-asistencia.css#L14-L111)

# Gestión de los Horarios

Este documento cubre la visualización de horarios y la funcionalidad de gestión dentro del sistema de asistencia al profesorado. El componente de gestión del horario es responsable de renderizar las tarjetas de horario del maestro y manejar la eliminación dinámica de las franjas horarias de horarios cuando se registra la asistencia.

Para obtener información sobre cómo interactúan los horarios con la grabación de asistencia, consulte Gestión de asistencia. Para más detalles sobre la navegación general del sistema, consulte Sistema de navegación.

## Finalidad del sistema y alcance

El sistema de gestión de horarios muestra los horarios de trabajo del profesorado como tarjetas visuales que muestran la fecha, la hora de entrada y la información del tiempo de salida. El sistema se integra con el seguimiento de asistencia eliminando automáticamente las franjas horarias del horario cuando los profesores completan su asistencia para ese período de tiempo.

## Arquitectura de componentes

El sistema de gestión de horarios sigue un patrón modular de componentes con archivos separados para estructura, estilo y comportamiento:

A black background with white text

AI-generated content may be incorrect.

**Fuentes:** [componentes/horarios/horarios.html1 a 19](https://github.com/lesquel/proyecto-asistencia-docentes/blob/63ad56c3/components/horarios/horarios.html#L1-L19) [componentes/horarios/horarios.js1 a 10](https://github.com/lesquel/proyecto-asistencia-docentes/blob/63ad56c3/components/horarios/horarios.js#L1-L10) [componentes/horarios/useHorarios.utils.js1 a 16](https://github.com/lesquel/proyecto-asistencia-docentes/blob/63ad56c3/components/horarios/useHorarios.utils.js#L1-L16)

## Estructura de datos y plantilla de refuerzo

### Cuadro de datos de calendario

Cada lista de entrada (horario) contiene las siguientes propiedades:

|  |  |  |
| --- | --- | --- |
| Bienes | Tipo | Descripción |
| fecha | Cadena | Fecha de la sesión prevista |
| entrada | Cadena | Tiempo de entrada/inicio |
| salida | Cadena | Tiempo de salida/fin |

### Implementación de la función de plantilla

El templateHorariosfunción crea tarjetas de horario a partir de datos de plantilla:



**Fuentes:** [componentes/horarios/horarios.js1-9](https://github.com/lesquel/proyecto-asistencia-docentes/blob/63ad56c3/components/horarios/horarios.js#L1-L9) [componentes/horarios/horarios.html2 a 12](https://github.com/lesquel/proyecto-asistencia-docentes/blob/63ad56c3/components/horarios/horarios.html#L2-L12)

## Generación y gestión de calendarios

### Funciones básicas

El sistema de gestión de horarios se desempeña dos funciones principales:

#### generarHorarios ()

Renderiza todas las entradas de horarios de la capa de datos en el contenedor DOM:

A screenshot of a computer

AI-generated content may be incorrect.

**removeHorarios()**

Borra todas las tarjetas de horario del contenedor de visualización mediante la configuración innerHTMLa la cuerda vacía.

**Fuentes:**

[componentes/horarios/useHorarios.utils.js5 a 15](https://github.com/lesquel/proyecto-asistencia-docentes/blob/63ad56c3/components/horarios/useHorarios.utils.js#L5-L15)

## Diseño visual y diseño

### Estructura de la tarjeta de calendario

Cada tarjeta de horario sigue un diseño visual consistente:

A black background with white text

AI-generated content may be incorrect.

## Especificaciones de Estilo CSS

|  |  |  |  |
| --- | --- | --- | --- |
| Elemento | Dimensiones | Antecedentes | Propiedades clave |
| .task-card | 465px . 75px | F8d57e | borde-radio: 20px, diseño de flexbox |
| .task-icon | 35px 35px | "ffffffff" | Sector fronterizo: 10px, emoji centrado |
| .task-title | flexible | transparente | Fuente de Gilroy-Bold, color n.b2442 |
| .task-time | Alineado por la derecha | transparente | Gilroy-Medium font, tamaño 14px |

## Conducta sensible

El sistema incluye un estilo móvil que responde al móvil para pantallas de ancho de 480px:

* Ancho de la tarjeta se ajusta al 100%
* Reducido el acolchado (15px vs 31px)
* márgenes de icono más pequeños (10px vs 17px)
* Tamaños de fuente reducidos para el título (14px) y tiempo (12px)

**Fuentes:**

[componentes/horarios/horarios.css77-144](https://github.com/lesquel/proyecto-asistencia-docentes/blob/63ad56c3/components/horarios/horarios.css#L77-L144)

## Integración con Capa de Datos

### Patrón de acceso a los datos

El sistema de gestión de horarios se integra con la capa de datos centralizada:

A screenshot of a black screen

AI-generated content may be incorrect.

**Fuentes:** [componentes/horarios/useHorarios.utils.js1 a 4](https://github.com/lesquel/proyecto-asistencia-docentes/blob/63ad56c3/components/horarios/useHorarios.utils.js#L1-L4)

## DOM Integración y Gestión de contenedores

### Elementos de contenedores

Tarjetas de horario se entregan en el #horarios\_conteelemento contenedor, que utiliza:

* Clases: flex-container-grid
* Estilo: margin: 1em 0
* Finalidad: contenedor de diseño de cuadrícula para tarjetas de horario

### Cargando del módulo

El componente de horario utiliza las importaciones de módulos ES6 y se inicializa a través de la plantilla HTML:



**Fuentes:** [componentes/horarios/horarios.html14 a 19](https://github.com/lesquel/proyecto-asistencia-docentes/blob/63ad56c3/components/horarios/horarios.html#L14-L19)

## Relación con el sistema de asistencia

El sistema de gestión de los horarios está estrechamente integrado con el seguimiento de la asistencia. Cuando los profesores registran asistencia, se eliminan las franjas horarias de horario completadas de la pantalla para evitar entradas duplicadas y mantener la consistencia de los datos. Esta integración se gestiona a través de la capa de datos compartida y funciones de utilidad coordinadas.

# Sistema de autenticación

## Finalidad y alcance

Este documento cubre el sistema de autenticación que gestiona la funcionalidad de inicio de sesión de usuario y logout en la aplicación de gestión de asistencia al profesor. El sistema proporciona un control seguro del acceso mediante autenticación basada en la forma y la gestión de sesiones. Para obtener información sobre la navegación entre zonas autenticadas y no autenticadas, consulte [Sistema de navegación](https://deepwiki.com/lesquel/proyecto-asistencia-docentes/3.5-navigation-system).

El sistema de autenticación consta de dos componentes primarios: el formulario de inicio de sesión para la autenticación del usuario y el botón de inicio de sesión para la terminación de sesión. Ambos componentes se integran con la capa central de datos y el sistema de enrutamiento para proporcionar experiencia de usuario sin problemas.

## Autación Flujo

El sistema de autenticación sigue un flujo de aplicación web estándar en el que los usuarios deben autenticarse antes de acceder a las principales características de aplicación.



**Fuentes:** [componentes/form-login/form-login.js41 a 56](https://github.com/lesquel/proyecto-asistencia-docentes/blob/63ad56c3/components/form-login/form-login.js#L41-L56) [componentes/logout-btn/logout-btn.js10 a 21](https://github.com/lesquel/proyecto-asistencia-docentes/blob/63ad56c3/components/logout-btn/logout-btn.js#L10-L21)

## Login de arquitectura de componentes

El componente de forma de inicio de sesión implementa un enfoque modular con preocupaciones separadas para la presentación, validación y lógica de negocio.

A screenshot of a computer

AI-generated content may be incorrect.

**Fuentes:** [componentes/form-login/form-login.html1 a 16](https://github.com/lesquel/proyecto-asistencia-docentes/blob/63ad56c3/components/form-login/form-login.html#L1-L16) [componentes/form-login/form-login.js1-57](https://github.com/lesquel/proyecto-asistencia-docentes/blob/63ad56c3/components/form-login/form-login.js#L1-L57)

## Forma la generación de campo

La forma de inicio de sesión genera dinámicamente campos de entrada utilizando el generateInputfunción de utilidad con patrones de validación específicos:

|  |  |  |  |
| --- | --- | --- | --- |
| Campo | Tipo | Patrón de validación | Aferradero de la Coloc |
| Correo electrónico | email | patternList.email | "Correo" |
| Contraseña | password | patternList.numberAndLetters | "Contraseña" |

Ambos campos están configurados con la misma clase CSS input-fieldy se insertan en el #form-logincontenedor. El formulario implementa validación del lado del cliente mediante la coincidencia de patrones antes de la presentación.

**Fuentes:** [componentes/form-login/form-login.js10 a 38](https://github.com/lesquel/proyecto-asistencia-docentes/blob/63ad56c3/components/form-login/form-login.js#L10-L38)

## Proceso de autenticación

El proceso de inicio de sesión sigue estos pasos:

1. **Formulario de envío Manejo** : El evento de envío se impide el comportamiento predeterminado del navegador
2. **Validación** : Cada campo de entrada llama showError()para mostrar los mensajes de validación
3. **Auténtico Intento** : El Data.login()método se llama con valores de correo electrónico y contraseña
4. **Redireccion en Success:** Si la autenticación tiene éxito, el usuario es redirigido a la primera ruta del sitio

**Fuentes:** [componentes/form-login/form-login.js41 a 56](https://github.com/lesquel/proyecto-asistencia-docentes/blob/63ad56c3/components/form-login/form-login.js#L41-L56)

## Arquitectura de componentes de sesión

El componente de logout proporciona un botón dinámico que cambia el comportamiento basado en el estado de autenticación del usuario.

A black and white diagram

AI-generated content may be incorrect.

**Fuentes:** [componentes/logout-btn/logout-btn.html1 a 6](https://github.com/lesquel/proyecto-asistencia-docentes/blob/63ad56c3/components/logout-btn/logout-btn.html#L1-L6) [componentes/logout-btn/logout-btn.js1 a 22](https://github.com/lesquel/proyecto-asistencia-docentes/blob/63ad56c3/components/logout-btn/logout-btn.js#L1-L22)

## Gestión estatal de Button

El botón de inicio de sesión implementa la gestión de estado inteligente:

* **Estado autentnticoted:** Muestra texto "Logout" y realiza la operación de logout cuando se hace clic
* **Estado noautizado** : Muestra texto "Entrar" texto y redirecciones a la ruta de autenticación cuando se hace clic

El Estado se determina llamando data.getUser()que devuelve la información actual de la sesión de usuario.

**Fuentes:** [componentes/logout-btn/logout-btn.js10 a 21](https://github.com/lesquel/proyecto-asistencia-docentes/blob/63ad56c3/components/logout-btn/logout-btn.js#L10-L21)

## Integración con Capa de Datos

Ambos componentes de autenticación se basan en la Dataclase para operaciones de autenticación básica:

|  |  |  |
| --- | --- | --- |
| Método | Objeto | Se utiliza por |
| login({email, password}) | Authenticar credenciales de usuario | componente de la forma-login |
| logout() | Terminar sesión de usuario | componente de logout-btn |
| getUser() | Comprobar el estado actual de autenticación | componente de logout-btn |

El Dataclase sirve como la única fuente de la verdad para el estado de autenticación y las operaciones a lo largo de la aplicación.

**Fuentes:** [componentes/form-login/form-login.js48 - 51](https://github.com/lesquel/proyecto-asistencia-docentes/blob/63ad56c3/components/form-login/form-login.js#L48-L51) [componentes/logout-btn/logout-btn.js10 a 14](https://github.com/lesquel/proyecto-asistencia-docentes/blob/63ad56c3/components/logout-btn/logout-btn.js#L10-L14)

## Integración de lanzamiento

El sistema de autenticación se integra con el sistema de enrutamiento de la aplicación a través de dos configuraciones de ruta:

* **authRoutes** : Rutas para usuarios no autentéticos (página de blogs)
* **sitioRutas** : Rutas para usuarios autenticados (aplicación principal)

La navegación entre las zonas autenticadas y no autentictadas es manejada por la useRoutesutilidad, que proporciona la redirect()método de navegación programática.

**Fuentes:** [componentes/form-login/form-login.js54](https://github.com/lesquel/proyecto-asistencia-docentes/blob/63ad56c3/components/form-login/form-login.js#L54-L54) [componentes/logout-btn/logout-btn.js14 a 19](https://github.com/lesquel/proyecto-asistencia-docentes/blob/63ad56c3/components/logout-btn/logout-btn.js#L14-L19)

# Perfil del profesor

## Finalidad y alcance

El componente de Perfil Docente proporciona una visual visualización de información del profesorado autenticada dentro del sistema de gestión de asistencia. Este componente proporciona los detalles básicos del profesor, incluyendo nombre, correo electrónico e imagen de perfil en un diseño basado en tarjetas.

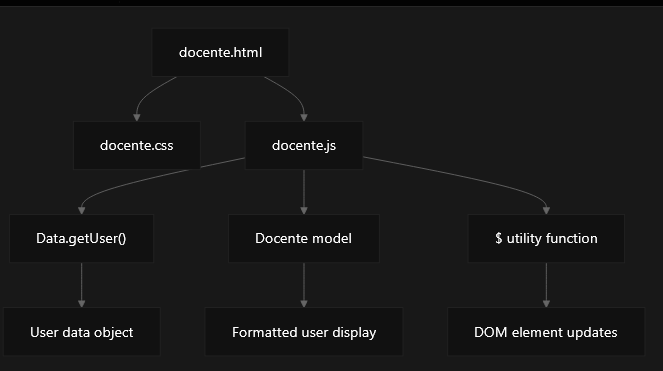
Para la autenticación y la gestión de las sesiones de los usuarios, consulte Sistema de autenticación. Para la navegación global del sistema, consulte Sistema de navegación.

## Reseña del componente

El docentecomponente es un elemento de interfaz de usuario autónomo que muestra información de perfil del profesor. Sigue el patrón de arquitectura de componentes estándar con archivos separados HTML, CSS y JavaScript.

|  |  |
| --- | --- |
| Archivo de componentes | Objeto |
| docente.html | Define la estructura y el diseño HTML |
| docente.css | Proporciona estilo y apariencia visual |
| docente.js | Maneja la recuperación de datos y la población de DOM |

## Estructura de componentes



Fuentes: [componentes/docente/docente.html1-9](https://github.com/lesquel/proyecto-asistencia-docentes/blob/63ad56c3/components/docente/docente.html#L1-L9) [componentes/docente/docente.css1-49](https://github.com/lesquel/proyecto-asistencia-docentes/blob/63ad56c3/components/docente/docente.css#L1-L49) [componentes/docente/docente.js1 a 10](https://github.com/lesquel/proyecto-asistencia-docentes/blob/63ad56c3/components/docente/docente.js#L1-L10)

## Flujo e integración de datos

El componente se integra con el sistema más amplio a través de la Dataclase y Docentemodelo para recuperar y en formato de la información del usuario.

### **Proceso de recuperación de datos**

A black and white diagram

AI-generated content may be incorrect.

El componente utiliza el siguiente patrón de flujo de datos:

1. Crea un Data instance para acceder a la información del usuario [componentes/docente/docente.js5](https://github.com/lesquel/proyecto-asistencia-docentes/blob/63ad56c3/components/docente/docente.js#L5-L5)
2. Recupera los datos actuales de los usuarios a través de data.getUser() [componentes/docente/docente.js6](https://github.com/lesquel/proyecto-asistencia-docentes/blob/63ad56c3/components/docente/docente.js#L6-L6)
3. Envuelve los datos del usuario en un Docente de modelo de instancia [componentes/docente/docente.js6](https://github.com/lesquel/proyecto-asistencia-docentes/blob/63ad56c3/components/docente/docente.js#L6-L6)
4. Populate elementos DOM usando el $ función de utilidad [componentes/docente/docente.js9 a 10](https://github.com/lesquel/proyecto-asistencia-docentes/blob/63ad56c3/components/docente/docente.js#L9-L10)

Fuentes: [componentes/docente/docente.js5 a 10](https://github.com/lesquel/proyecto-asistencia-docentes/blob/63ad56c3/components/docente/docente.js#L5-L10)