

# Sistema de fichajes para Nashe Company.



Índice:

<b>1. Detalles de la aplicación.....</b>	<b>3</b>
1.1 Requisitos funcionales.....	3
1.2 Requisitos técnicos.....	4
<b>2. Bases de Datos.....</b>	<b>4</b>
2.1 Diagrama/esquema.....	4
2.2 BD Para pruebas.....	5
<b>3. Descripción del Interfaz Gráfico.....</b>	<b>6</b>
3.1. Pantalla de Acceso (Login):.....	6
3.2. Panel de Control (Dashboard):.....	6
3.3. Visualización de Datos:.....	6
<b>4. Pruebas de Uso.....</b>	<b>7</b>
4.1 Historial para un Rango:.....	7
4.2 y 4.3 Historial Semanal/Mensual:.....	7
4.4 Días ausente:.....	8
4.5 Trabajo fuera de jornada:.....	8
4.6 Olvido o Enfermedad:.....	9

# **1. Detalles de la aplicación.**

## **1.1 Requisitos funcionales.**

La aplicación está diseñada para centralizar el control de asistencia de los empleados de Nashe Company. Sus funciones principales son:

Gestión de Accesos: Autenticación basada en departamento y selección de empleado con validación de contraseña.

Control de Presencia (Punch System): Registro de eventos de entrada (Clock In), salida (Clock Out) y gestión de descansos (Break In/Out).

Lógica de Negocio y Restricciones:

Impedimento de múltiples sesiones de trabajo en un mismo día.

Validación de estados: no se permite registrar descansos sin haber iniciado jornada.

Control de jornada cerrada: no se permiten acciones tras el Clock Out.

Cálculo Automático de Horas:

Horas Efectivas: Cálculo del tiempo total de trabajo descontando los periodos de descanso.

Horas Extra: Identificación de horas trabajadas fuera del horario establecido (09:00 - 15:00) o en días no laborables (Lunes y Viernes).

Detección de Impuntualidad: Registro de entradas posteriores al margen de cortesía (09:10).

Panel de Estadísticas Dinámicas: Visualización de métricas filtradas por día, semana, mes, año o histórico completo.

## **1.2 Requisitos técnicos.**

Arquitectura: Modelo Cliente-Servidor basado en REST API.

Backend: Desarrollado en Java 17 con el framework Spring Boot. Utiliza Spring Data JPA para la persistencia.

Frontend: Interfaz web construida con HTML5, JavaScript (ES6+) y Bootstrap 5 para el diseño responsivo.

Comunicación: Intercambio de datos mediante objetos JSON a través de peticiones asíncronas (fetch).

## **2. Bases de Datos.**

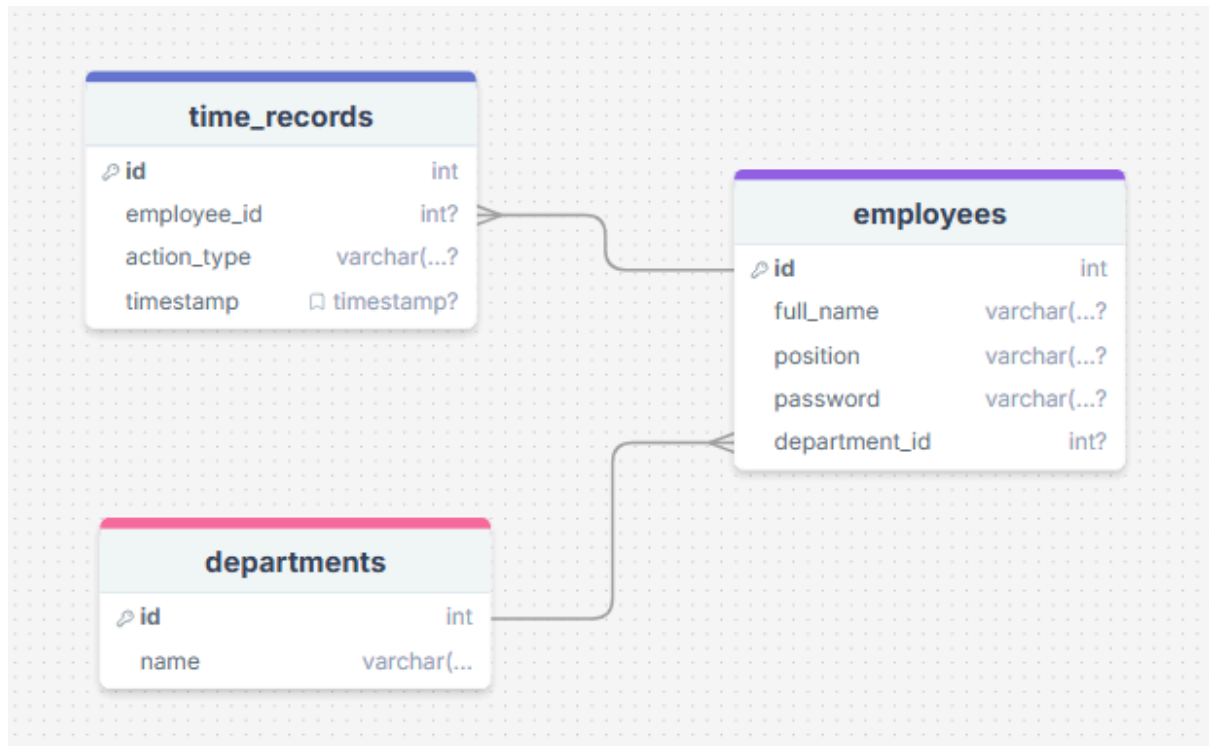
### **2.1 Diagrama/esquema.**

El sistema utiliza un modelo relacional con las siguientes entidades principales:

Department: Almacena los departamentos de Nashe Company (ej. IT, Marketing).

Employee: Contiene la información personal, cargo, departamento asociado y credenciales de acceso.

Time Record: Registra cada evento de fichaje con su marca de tiempo (Timestamp), tipo de acción y relación con el empleado.



## 2.2 BD Para pruebas.

El sistema utiliza H2 Database, una base de datos SQL en memoria que se integra directamente con el motor de Spring Boot. Esto permite que el entorno de pruebas sea volátil y controlado.

Persistencia: Los datos se mantienen mientras la aplicación está en ejecución y se reinician en cada despliegue, garantizando que las pruebas de fichaje empiecen siempre desde un estado conocido.

Carga de Datos (Seed): Al iniciar, el sistema utiliza el archivo data.sql o la lógica de JPA para cargar los departamentos y empleados de Nashe Company definidos en la fase de análisis.

Integridad: Se han definido relaciones de clave foránea entre las tablas Employee y TimeRecord para asegurar que ningún fichaje quede huérfano sin un empleado asociado.

### 3. Descripción del Interfaz Gráfico.

El diseño de la interfaz ha sido desarrollado bajo principios de UX (User Experience) para minimizar errores humanos durante el registro de jornada.

#### 3.1. Pantalla de Acceso (Login):

Selectores Inteligentes: El usuario selecciona su departamento (ej. IT, Administración) y el sistema filtra automáticamente los empleados correspondientes.

Seguridad: Incluye un campo de contraseña enmascarado que valida las credenciales contra el repositorio de empleados antes de liberar el dashboard.

#### 3.2. Panel de Control (Dashboard):

Botonera de Acción: Cuatro botones de colores diferenciados (Verde: Entrada, Amarillo/Azul: Descansos, Rojo: Salida) que se habilitan o deshabilitan según el estado actual del empleado.

Feedback Inmediato: Al realizar un fichaje, el sistema lanza una alerta confirmando la acción o indicando si existe un error lógico (ej. intentar salir sin haber entrado).

#### 3.3. Visualización de Datos:

KPIs Superiores: Resumen estadístico de las medias de trabajo, descanso y extras de los últimos 2 meses.

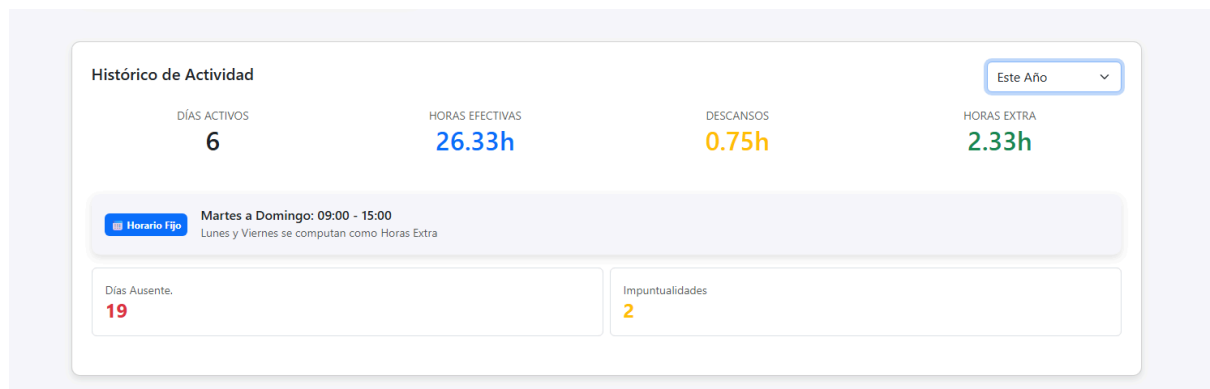
Sección de Horario Fijo: Un cuadro informativo que recuerda al empleado de Nashe Company su horario obligatorio (Martes a Domingo, 09:00 - 15:00).

Histórico Detallado: Tabla dinámica que muestra días activos, horas efectivas y, fundamentalmente, el conteo de Ausencias e Impuntualidades.

## 4. Pruebas de Uso.

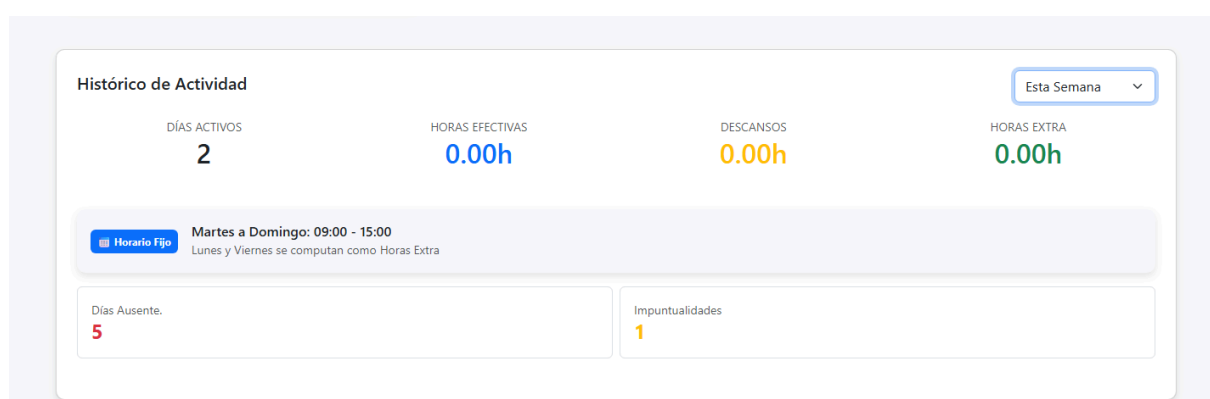
### 4.1 Historial para un Rango:

Selecciona el filtro "Este Mes" o "Este Año" y comprueba cómo cambian los totales.



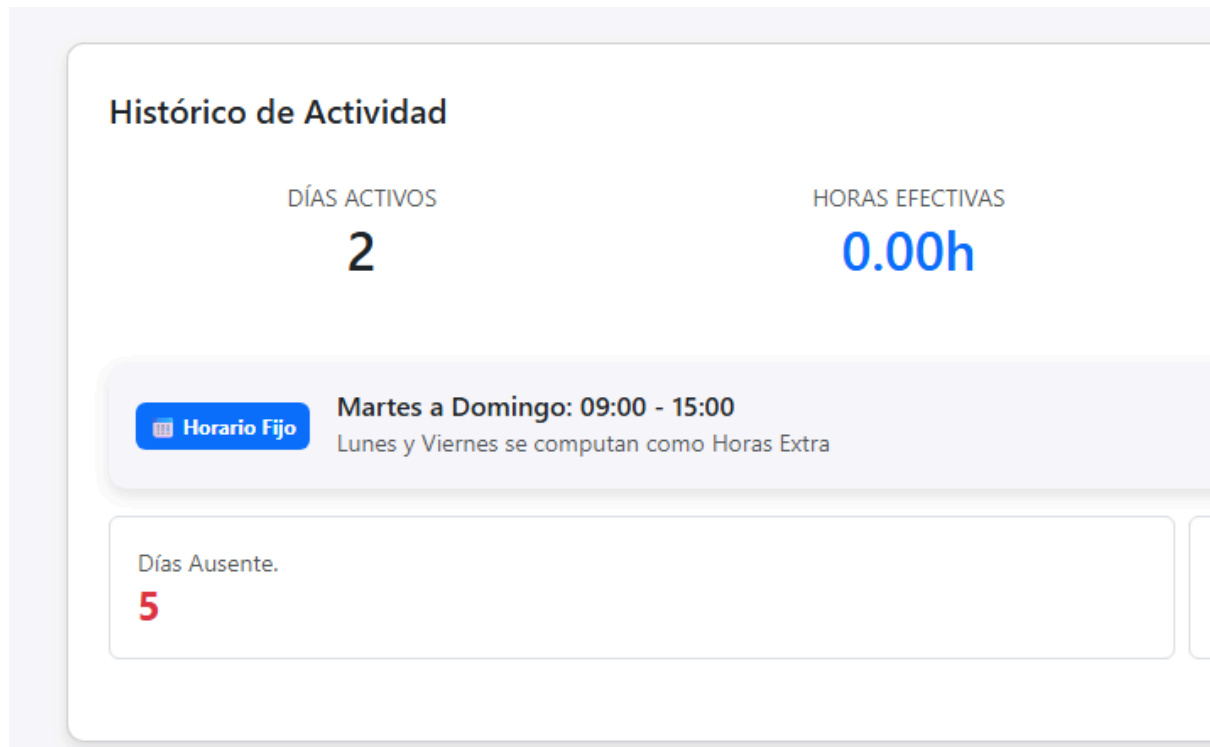
### 4.2 y 4.3 Historial Semanal/Mensual:

Verifica que el sistema sume correctamente las "Horas Efectivas" (restando los descansos) y que el contador de "Días con falta" (absences) sea correcto si no fichaste un miércoles, por ejemplo.



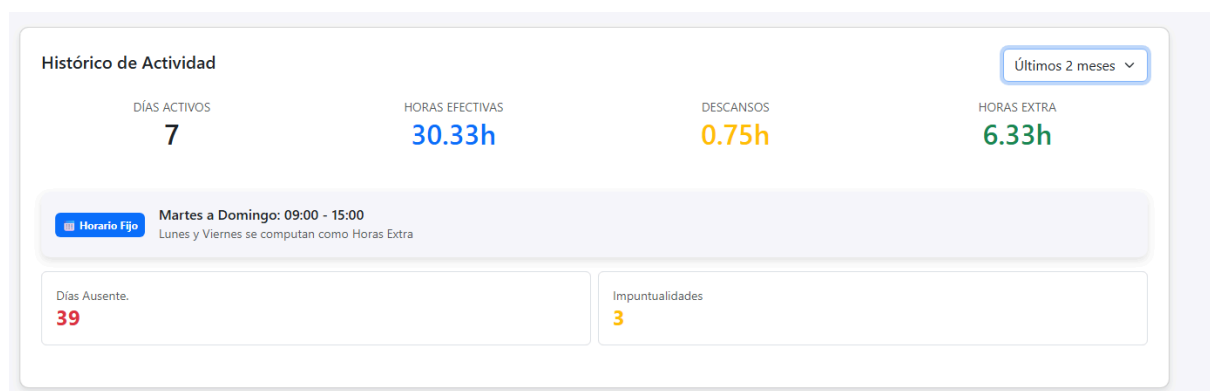
## 4.4 Días ausente:

Realiza una prueba no fichando un día obligatorio (Martes-Domingo) y observa cómo sube el contador de ausencias.



## 4.5 Trabajo fuera de jornada:

Ficha un Lunes o un Viernes y comprueba que esas horas se sumen íntegramente al contador de "Horas Extra".



Data Output Messages Notifications				
SQL				
	id [PK] bigint	employee_id bigint	action_type character varying (255)	timestamp timestamp without time zone
1	1	1	CLOCK_IN	2025-12-02 09:00:00
2	2	1	CLOCK_OUT	2025-12-02 15:00:00
3	3	1	CLOCK_IN	2025-12-03 09:00:00
4	4	1	BREAK_IN	2025-12-03 11:00:00
5	5	1	BREAK_OUT	2025-12-03 11:30:00
6	6	1	CLOCK_OUT	2025-12-03 15:30:00
7	7	1	CLOCK_IN	2025-12-05 10:00:00
8	8	1	CLOCK_OUT	2025-12-05 14:00:00
9	9	1	CLOCK_IN	2026-01-13 09:15:00
10	10	1	CLOCK_OUT	2026-01-13 17:15:00
11	11	1	CLOCK_IN	2026-01-14 09:00:00
12	12	1	CLOCK_OUT	2026-01-14 15:00:00
13	13	1	CLOCK_IN	2026-01-15 08:55:00
14	14	1	BREAK_IN	2026-01-15 10:00:00
15	15	1	BREAK_OUT	2026-01-15 10:45:00
16	16	1	BREAK_IN	2026-01-15 13:00:00
17	17	1	BREAK_OUT	2026-01-15 13:15:00
18	18	1	CLOCK_OUT	2026-01-15 16:00:00
19	19	1	CLOCK_IN	2026-01-25 09:00:00
20	20	1	CLOCK_OUT	2026-01-25 15:00:00
21	21	1	CLOCK_IN	2026-02-03 22:04:21.00217
22	22	1	CLOCK_OUT	2026-02-03 22:04:31.674929
23	23	1	CLOCK_IN	2026-02-04 00:19:39.157044
24	24	1	CLOCK_OUT	2026-02-04 00:19:43.587689

## 4.6 Olvido o Enfermedad:

Análisis: Si un usuario olvida fichar la salida, el sistema le impedirá fichar la entrada al día siguiente por "Sesión Abierta".

Solución propuesta: Se requiere que un administrador de Nashe Company limpie el registro o inserte la marca manualmente en la base de datos, ya que el sistema prioriza la integridad de los datos sobre la flexibilidad manual.