

Employees

Profiles

Employers

Home

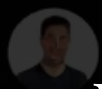
Densión

Employee Software

## Employees Employee Data

Ratio  
10.8%Location  
23.5Salary Amount  
25 21.4Portfolio  
\$735Contract Price  
\$23,57

## Employee Stats



Employee Name

\$55.50%

+

17% Investment

%

1,8713%

# Sistema de Gestión de Empleados – Hiberus

Esta documentación técnica describe el Sistema de Gestión de Empleados Hiberus, una aplicación web desarrollada con Angular y Spring Boot que permite administrar empleados y departamentos, así como visualizar métricas organizacionales clave. El documento abarca la arquitectura del sistema, componentes principales, tecnologías utilizadas, instrucciones de instalación y funcionalidades disponibles.



## Employee in Data



# Introducción y Descripción General

El sistema Hiberus es una aplicación web empresarial diseñada para optimizar la gestión de recursos humanos. Esta solución integral permite a los administradores mantener un control eficiente sobre la estructura organizativa de la empresa a través de una interfaz intuitiva y moderna.

## Funcionalidades principales:

- Gestión de empleados
- Administración de departamentos
- Vinculación bidireccional entre empleados y departamentos
- Visualización de métricas organizativas críticas
- Dashboard interactivo con indicadores clave

La arquitectura del sistema sigue un modelo cliente-servidor, con una clara separación entre el frontend desarrollado en Angular 19 y el backend implementado con Spring Boot 3.4.5, asegurando escalabilidad, mantenibilidad y rendimiento óptimo para entornos empresariales.

# Arquitectura del Backend

El backend del sistema Hiberus está desarrollado con Spring Boot 3.4.5, implementando una arquitectura multicapa que garantiza la separación de responsabilidades y facilita el mantenimiento del código.

## Controllers

Definen los endpoints REST que exponen las funcionalidades del sistema. Implementan los métodos HTTP necesarios para las operaciones CRUD de empleados y departamentos, así como para obtener métricas especializadas.

## Services

Contienen la lógica de negocio que procesa las solicitudes de los controladores. Implementan validaciones, transformaciones de datos y coordinan las operaciones con la capa de repositorios.

## Repositories

Gestionan el acceso a datos mediante JPA/Hibernate, proporcionando una capa de abstracción sobre la base de datos relacional y facilitando las operaciones CRUD de las entidades.

Componentes adicionales incluyen DTOs para transferencia eficiente de datos, manejadores de excepciones centralizados y CORS para garantizar una comunicación segura con el frontend.

# API REST y Endpoints

La API REST del sistema Hiberus proporciona un conjunto completo de endpoints para interactuar con las funcionalidades del sistema. Estos endpoints siguen las mejores prácticas RESTful y están diseñados para ser intuitivos y consistentes.

Método	Endpoint	Descripción
GET	/api/employee/all	Recupera el listado completo de empleados activos
POST	/api/department/create	Crea un nuevo departamento en el sistema
POST	/api/department/delete/{id}	Realiza un borrado lógico de un departamento
POST	/api/employee/create/{departmentId}	Crea un empleado asociándolo al departamento especificado
POST	/api/employee/delete/{employeeId}	Realiza un borrado lógico de un empleado
GET	/api/employee/highestSalary	Obtiene el empleado con el salario más elevado
GET	/api/employee/lowerAge	Devuelve el empleado de menor edad en la organización
GET	/api/employee/countLastMonth	Contabiliza las contrataciones realizadas en el último mes

Todos los endpoints retornan respuestas en formato JSON con códigos HTTP estándar para indicar el resultado de la operación, facilitando su integración con el frontend Angular.

# Arquitectura del Frontend

El frontend del sistema Hiberus está implementado con Angular 19.2.3, aprovechando las últimas características de este potente framework para crear una interfaz de usuario moderna, reactiva y de alto rendimiento.



## Módulos y Componentes

La aplicación se estructura en módulos funcionales como empleados, departamentos y métricas. Cada módulo contiene componentes reutilizables como tablas, formularios y modales.



## Servicios e Interfaces

Los servicios encapsulan la lógica de comunicación con el backend mediante el cliente HTTP de Angular. Las interfaces TypeScript definen contratos claros para los modelos de datos, mejorando la mantenibilidad.



## Experiencia de Usuario

La interfaz utiliza Bootstrap para garantizar un diseño responsivo y consistente. Incorpora componentes interactivos y formularios con validación en tiempo real.

La arquitectura frontend implementa el patrón de diseño Presentador-Servicio-Modelo, separando claramente las responsabilidades y facilitando las pruebas unitarias de cada componente.

# Instalación y Configuración

El sistema Hiberus está diseñado para facilitar tanto el despliegue en producción mediante contenedores Docker como el desarrollo local. A continuación se detallan ambos procedimientos:



## Despliegue con Docker

La forma más rápida y recomendada para desplegar el sistema completo:

```
git clone https://github.com/Adrian31BT/Hiberus.git
cd Hiberus
docker-compose up -d
```

Este proceso despliega automáticamente el frontend (<http://localhost:4200>) y el backend (<http://localhost:8080>).



## Configuración para Desarrollo

Para entornos de desarrollo que requieren acceso directo al código:

- Backend: Ejecutar **mvn clean install** seguido de **mvn spring-boot:run** en el directorio /backend
- Frontend: Ejecutar **npm install** seguido de **ng serve** en el directorio /frontend

## Requisitos del Sistema:

- Node.js v22.12.0
- Angular CLI 19.2.3
- JDK 21
- Maven
- Docker & Docker Compose (para despliegue containerizado)

# Modelo de Datos

El sistema Hiberus utiliza un modelo de datos relacional implementado mediante JPA/Hibernate. Las entidades principales y sus relaciones se describen a continuación:

## Entidad Empleado

- **ID:** Identificador único (PK)
- **Nombre:** Nombre del empleado
- **Apellido:** Apellido del empleado
- **Edad:** Edad en años
- **Salario:** Remuneración mensual
- **Fecha de inicio:** Fecha de contratación
- **Fecha de fin:** Fecha de baja (opcional)
- **Estado:** Activo/Inactivo (borrado lógico)
- **Departamento:** Relación ManyToOne

## Entidad Departamento

- **ID:** Identificador único (PK)
- **Nombre:** Nombre del departamento
- **Estado:** Activo/Inactivo (borrado lógico)
- **Empleados:** Relación OneToMany

La relación entre ambas entidades es bidireccional, permitiendo navegar desde un empleado a su departamento y desde un departamento a todos sus empleados. El sistema implementa borrado lógico mediante el campo estado para mantener la integridad histórica de los datos.

# Funcionalidades y Dashboard

El sistema Hiberus proporciona un conjunto completo de funcionalidades para la gestión eficiente de empleados y departamentos, presentadas a través de interfaces intuitivas y organizadas por categorías.



## Dashboard de Métricas

Panel central que visualiza información crítica como el empleado con mayor salario, el más joven de la organización y un contador de contrataciones recientes, ofreciendo una visión general del estado de la plantilla.



## Gestión de Empleados

Permite crear empleados mediante formularios validados. Incluye funcionalidades para asignar empleados a departamentos y gestionar su información personal y contractual.



## Administración de Departamentos

Facilita la creación y gestión de la estructura organizativa mediante un sistema de departamentos. Incluye visualización de empleados asignados a cada departamento.

Todas las funcionalidades implementan validaciones tanto en el frontend como en el backend para garantizar la integridad de los datos. La interfaz de usuario prioriza la usabilidad mediante diseños intuitivos y flujos de trabajo optimizados para operaciones frecuentes, reduciendo la curva de aprendizaje para los administradores del sistema.