



4-6-2025

Centro de día Bullejos

Documentación técnica



Autor: José Carlos Bullejos Gómez

Índice

| | |
|--|----|
| 1. Introducción | 2 |
| 2. Funciones y rendimientos deseados..... | 2 |
| 3. Objetivos | 3 |
| 4. Planteamiento y evaluación de diversas soluciones..... | 4 |
| 5. Justificación de la solución elegida | 5 |
| 6. Modelado de la solución..... | 6 |
| 6.1 Recursos humanos | 6 |
| 6.2 Recursos hardware..... | 6 |
| 6.3 Recursos software | 9 |
| 7. Planificación temporal. Etapas..... | 11 |
| 8. Ejecución del proyecto y documentación técnica | 12 |
| 8.1 Arquitectura del sistema | 12 |
| 8.2 Diseño y explicación de la base de datos..... | 12 |
| 8.3 Estructura de la aplicación | 19 |
| 8.4 Infraestructura como código (Terraform) | 21 |
| 8.5 Orquestación de contenedores (Kubernetes)..... | 22 |
| 8.6 Integración y despliegue continuo (CI/CD) | 22 |
| 8.7 Prácticas de seguridad | 22 |
| 9. Fase de pruebas | 23 |
| 10. Documentación del sistema..... | 24 |
| 10.1 Introducción a la aplicación | 24 |
| 10.2 Manual de instalación..... | 24 |
| 10.3 Manual de usuario | 29 |
| 10.4 Manual de administración | 34 |
| 11. Conclusiones finales..... | 41 |
| 12. Grado de cumplimiento de los objetivos fijados | 41 |
| 13. Propuesta de modificaciones o ampliaciones futuras | 42 |
| 14. Bibliografía empleada | 42 |

1. Introducción

El presente proyecto tiene como finalidad el diseño, desarrollo y despliegue de una plataforma web integral para la gestión de un centro de día, utilizando tecnologías cloud y metodologías modernas de administración de sistemas.

La solución propuesta se basa en una aplicación web desarrollada en PHP y NGINX, desplegada sobre una infraestructura cloud en AWS, utilizando Kubernetes (EKS) para la orquestación de contenedores, almacenamiento compartido con EFS, y automatización de la infraestructura mediante Terraform.

La plataforma permite la gestión centralizada de usuarios, trabajadores, alumnos, inventario, cursos, tareas y documentación, así como la automatización de copias de seguridad y la integración de un servidor FTP para la gestión de archivos.

El sistema está diseñado para ser escalable, seguro y fácil de administrar, permitiendo su adaptación a las necesidades cambiantes del centro y facilitando la incorporación de nuevas funcionalidades en el futuro.

La elección de tecnologías cloud y de prácticas de Infraestructura como Código (IaC) responde a la necesidad de garantizar la reproducibilidad, la automatización y la facilidad de mantenimiento, alineándose con las competencias profesionales del ciclo ASIR y con las tendencias actuales en la administración de sistemas.

2. Funciones y rendimientos deseados

La plataforma debe cumplir una serie de funciones y requisitos de rendimiento que garanticen su utilidad, eficiencia y seguridad:

- Gestión integral de usuarios: Permitir el alta, consulta, modificación y baja de usuarios, trabajadores y alumnos, con control de acceso según roles y validaciones automáticas de los datos introducidos.
- Gestión académica: Facilitar la creación y gestión de cursos, asignaturas, matrículas y notas, permitiendo el seguimiento académico de los alumnos.
- Gestión de tareas y asignaciones: Permitir la creación, asignación y seguimiento de tareas, facilitando la organización del trabajo y la colaboración entre los diferentes perfiles del centro.
- Gestión de inventario: Controlar el stock de materiales y recursos del centro, permitiendo el registro de entradas y salidas, la actualización de unidades.
- Inscripción online y portal informativo: Ofrecer un portal web accesible para usuarios y familiares, permitiendo la inscripción online, la consulta de información y la descarga de documentación relevante.

- Gestión de archivos y documentación: Integrar un servidor FTP para la subida y descarga de archivos, garantizando la seguridad y la trazabilidad de los documentos.
- Seguridad y control de acceso: Implementar mecanismos de autenticación y autorización, cifrado de contraseñas, validación de entradas y protección contra ataques comunes (inyección SQL, XSS, etc.).
- Escalabilidad y alta disponibilidad: Permitir el escalado horizontal de los servicios y garantizar la disponibilidad del sistema ante fallos o picos de carga.
- Automatización de copias de seguridad: Realizar copias de seguridad automáticas de la base de datos y los archivos, almacenándolas de forma segura en S3 y permitiendo su restauración en caso de necesidad.
- Facilidad de administración y mantenimiento: Proporcionar herramientas y paneles de administración intuitivos, así como documentación clara para la gestión y el mantenimiento del sistema.
- Monitorización y logs: Permitir la monitorización del estado de los servicios y el registro de logs para la detección y resolución de incidencias.
- Rendimiento: Garantizar tiempos de respuesta adecuados, incluso en situaciones de alta demanda, y optimizar el uso de recursos para minimizar costes.

3. Objetivos

Objetivos funcionales:

- Centralizar la gestión de todos los procesos del centro en una única plataforma accesible desde cualquier lugar.
- Automatizar la inscripción de usuarios y alumnos, reduciendo los tiempos de espera y los errores manuales.
- Facilitar la comunicación y el acceso a la información para usuarios, trabajadores y familiares, mejorando la transparencia y la satisfacción de los usuarios.

Objetivos técnicos:

- Desplegar la aplicación en un entorno cloud seguro, escalable y de alta disponibilidad, utilizando las mejores prácticas de la industria.
- Utilizar contenedores y orquestación con Kubernetes para facilitar el despliegue, la gestión y la actualización de los servicios.
- Implementar la infraestructura como código con Terraform para garantizar la reproducibilidad, la trazabilidad y la automatización de los despliegues.
- Integrar almacenamiento compartido con EFS y copias de seguridad automáticas en S3, garantizando la persistencia y la seguridad de los datos.
- Garantizar la seguridad de los datos mediante el uso de cifrado, control de acceso, validaciones y buenas prácticas de desarrollo seguro.

Objetivos operativos:

- Reducir los tiempos de gestión y minimizar los errores manuales mediante la automatización de procesos y la validación de datos.
- Permitir la administración remota y la monitorización del sistema, facilitando la detección y resolución de incidencias.
- Facilitar la recuperación ante fallos o desastres mediante copias de seguridad automáticas y procedimientos de restauración documentados.

Objetivos de aprendizaje y mejora profesional:

- Adquirir experiencia práctica en el diseño, despliegue y administración de infraestructuras cloud y sistemas de orquestación de contenedores.
- Aplicar metodologías de Infraestructura como Código y automatización de despliegues en un entorno real.
- Desarrollar habilidades en la documentación y la formación de usuarios y administradores.

4. Planteamiento y evaluación de diversas soluciones

Durante la fase de análisis se han considerado varias alternativas para la modernización y digitalización del sistema de gestión del centro:

Opción 1: Despliegue tradicional en un único servidor físico o virtual

Ventajas:

- Simplicidad inicial y menor curva de aprendizaje.
- Control total sobre el entorno y la configuración.

Inconvenientes:

- Escalabilidad limitada y dificultad para adaptarse a cambios en la demanda.
- Punto único de fallo, lo que aumenta el riesgo de caídas y pérdidas de datos.
- Mayor dificultad para la recuperación ante desastres y la realización de copias de seguridad automáticas.
- Menor flexibilidad para la integración de nuevas funcionalidades o servicios.

Opción 2: Plataforma cloud-nativa con contenedores, Kubernetes y automatización (solución elegida)

Ventajas:

- Escalabilidad horizontal y alta disponibilidad garantizadas por el propio diseño de la infraestructura.
- Automatización de despliegues, actualizaciones y copias de seguridad.
- Flexibilidad para la integración de nuevos servicios y la adaptación a las necesidades del centro.
- Facilidad de recuperación ante fallos y desastres, gracias a la automatización y la redundancia.
- Alineación con las competencias profesionales del ciclo ASIR y con las tendencias actuales en la administración de sistemas.

5. Justificación de la solución elegida

La elección de una arquitectura cloud-nativa basada en AWS, Kubernetes y Terraform responde a la necesidad de garantizar la escalabilidad, la alta disponibilidad, la seguridad y la facilidad de administración del sistema.

Esta solución permite automatizar todos los procesos de despliegue, actualización y copia de seguridad, reduciendo la carga de trabajo del administrador y minimizando el riesgo de errores humanos.

El uso de contenedores y orquestación con Kubernetes facilita la gestión de los servicios, permitiendo su escalado y actualización sin interrupciones, así como la recuperación automática ante fallos.

La integración de almacenamiento compartido con EFS y copias de seguridad automáticas en S3 garantiza la persistencia y la seguridad de los datos, permitiendo la restauración rápida en caso de necesidad.

La infraestructura como código con Terraform permite definir, desplegar y modificar toda la infraestructura de forma reproducible y controlada, facilitando la trazabilidad de los cambios y la colaboración entre diferentes administradores.

En resumen, la solución elegida permite cumplir todos los objetivos funcionales, técnicos y operativos del proyecto, garantizando la sostenibilidad y la adaptabilidad del sistema a largo plazo.

6. Modelado de la solución

6.1 Recursos humanos

El proyecto requiere la participación de diferentes perfiles:

Administrador del sistema:

Responsable del diseño, despliegue, mantenimiento y monitorización de la plataforma.

Encargado de la gestión de la infraestructura cloud, la seguridad, las copias de seguridad y la resolución de incidencias.

Responsable de la formación y el soporte a los usuarios y trabajadores del centro.

Personal del centro:

Trabajadores y administradores encargados de la gestión de usuarios, cursos, inventario y tareas.

Responsables de la atención a los usuarios y de la actualización de la información en la plataforma.

Usuarios finales:

Alumnos, usuarios y familiares que acceden al portal para consultar información, realizar gestiones y comunicarse con el centro.

Responsables de mantener actualizados sus datos personales y de utilizar la plataforma de forma segura y responsable.

6.2 Recursos hardware

La infraestructura se basa en recursos cloud de AWS, que proporcionan la flexibilidad y la escalabilidad necesarias para el correcto funcionamiento del sistema:

VPC y subredes públicas:

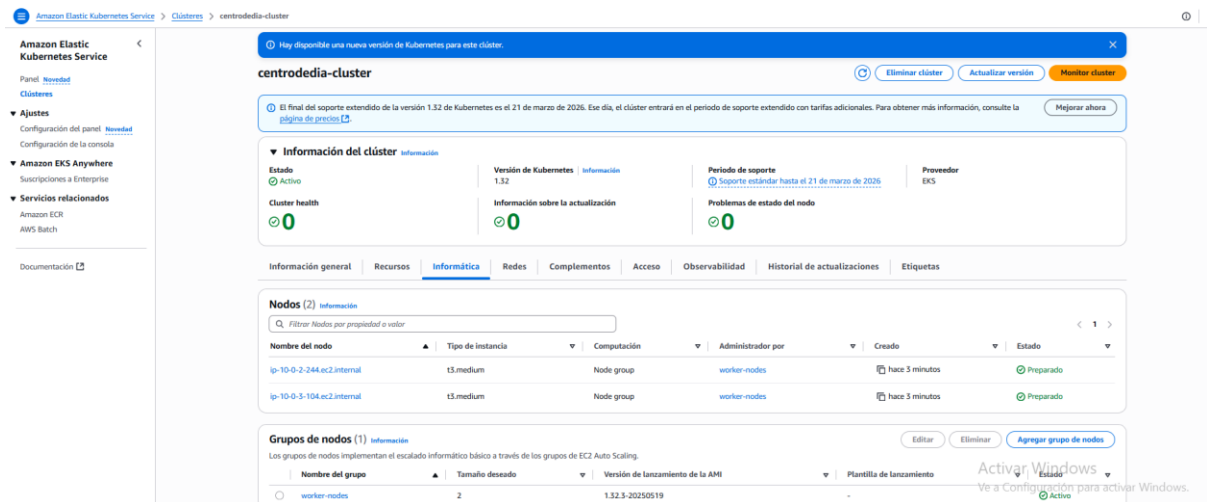
Permiten aislar y segmentar la red, garantizando la seguridad y el control del tráfico.

Facilitan la integración de diferentes servicios y la gestión de la conectividad.

Cluster EKS (Kubernetes):

Orquesta los contenedores de la aplicación web, el servidor FTP y la base de datos.

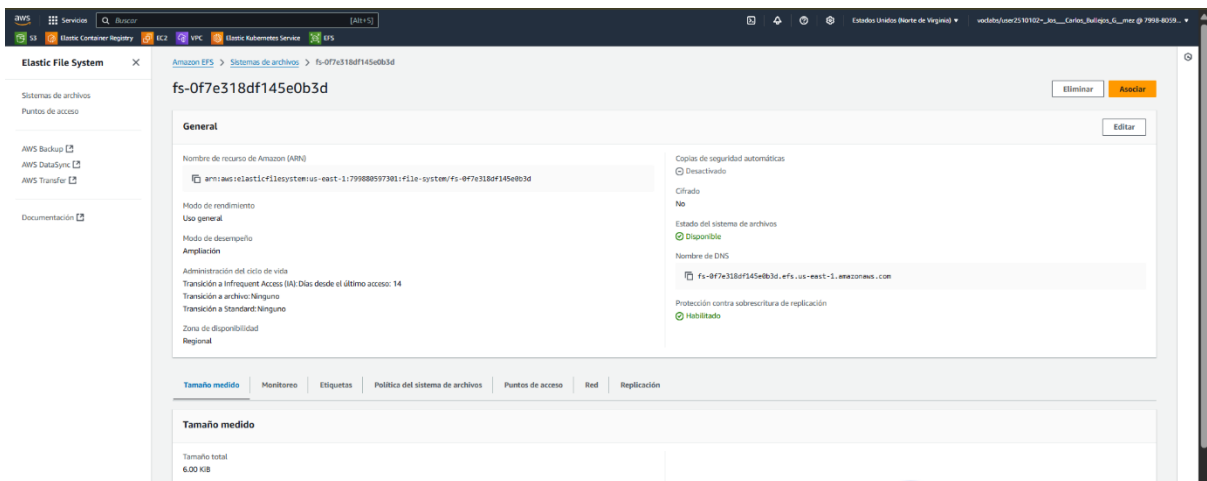
Permite el escalado automático de los servicios y la recuperación ante fallos.



EFS (Elastic File System):

Proporciona almacenamiento compartido y persistente para el código, los archivos subidos y los datos de la base de datos.

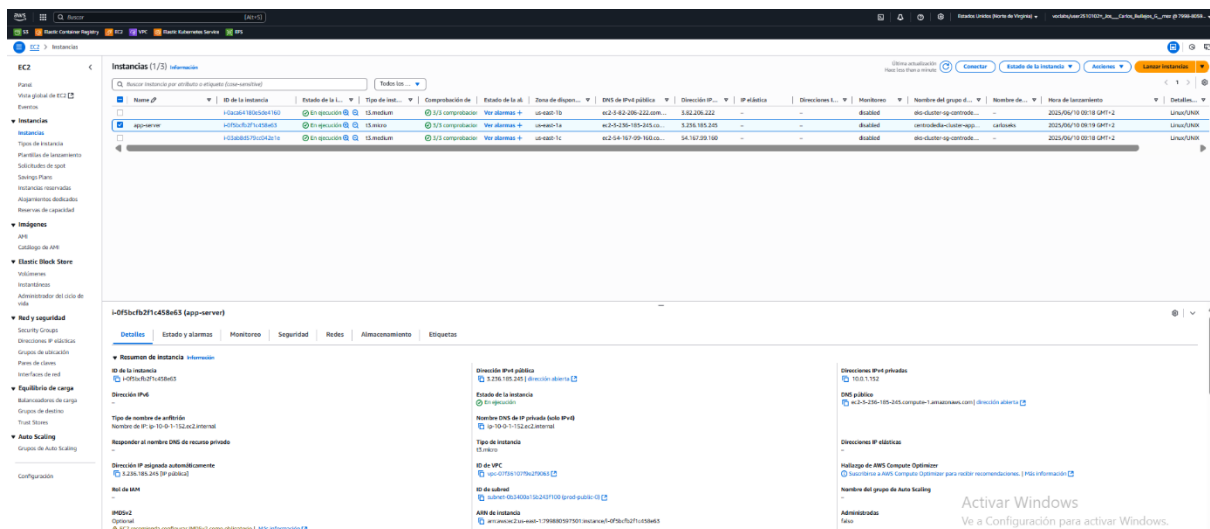
Permite la compartición de archivos entre diferentes servicios y la realización de copias de seguridad automáticas.



Instancia EC2:

Utilizada para tareas de administración, backups y sincronización con S3.

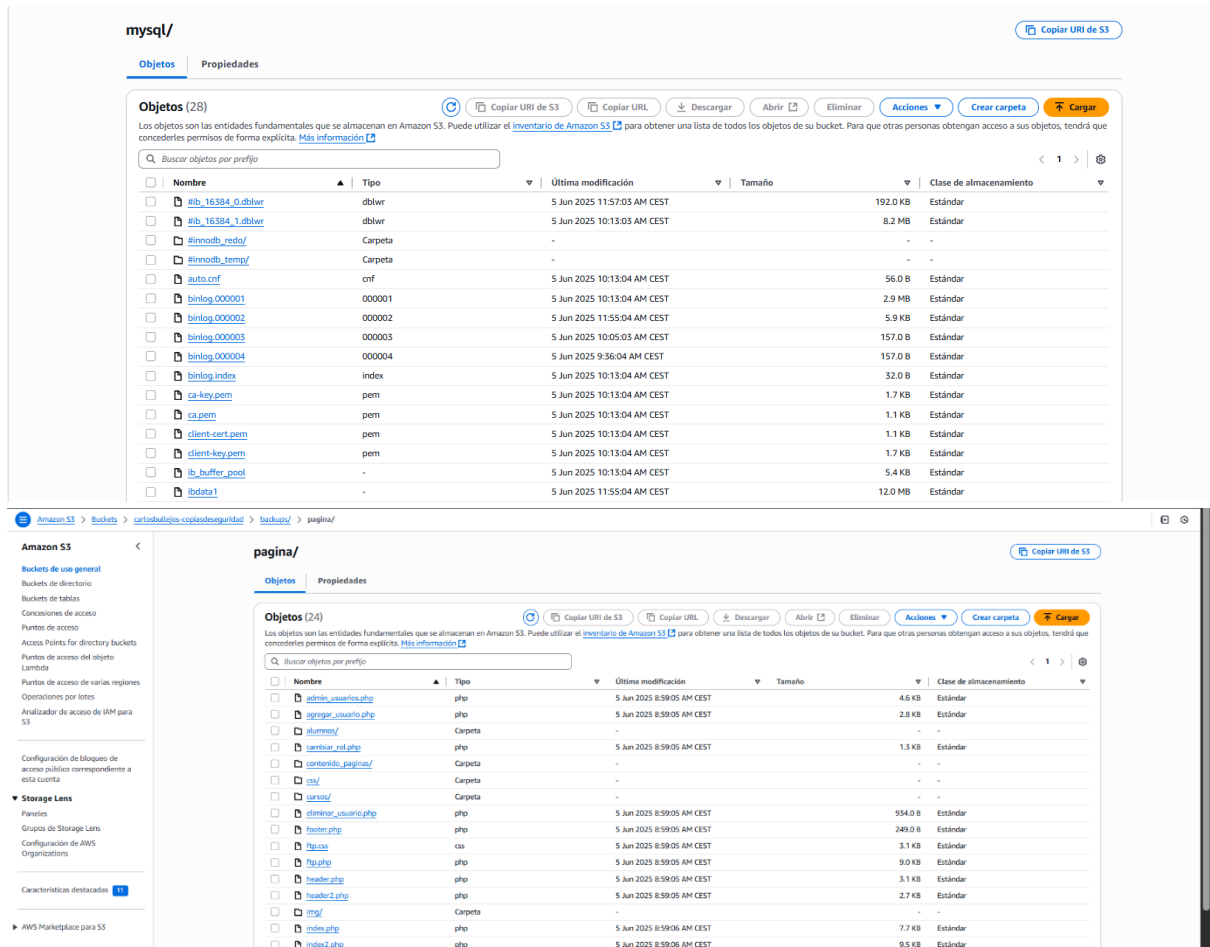
Permite la ejecución de scripts de mantenimiento y la monitorización del sistema.



Bucket S3:

Almacena las copias de seguridad automáticas de la base de datos y los archivos.

Permite la restauración rápida en caso de fallo o pérdida de datos.



Security Groups:

Controlan el acceso a los diferentes servicios, garantizando la seguridad de la infraestructura.

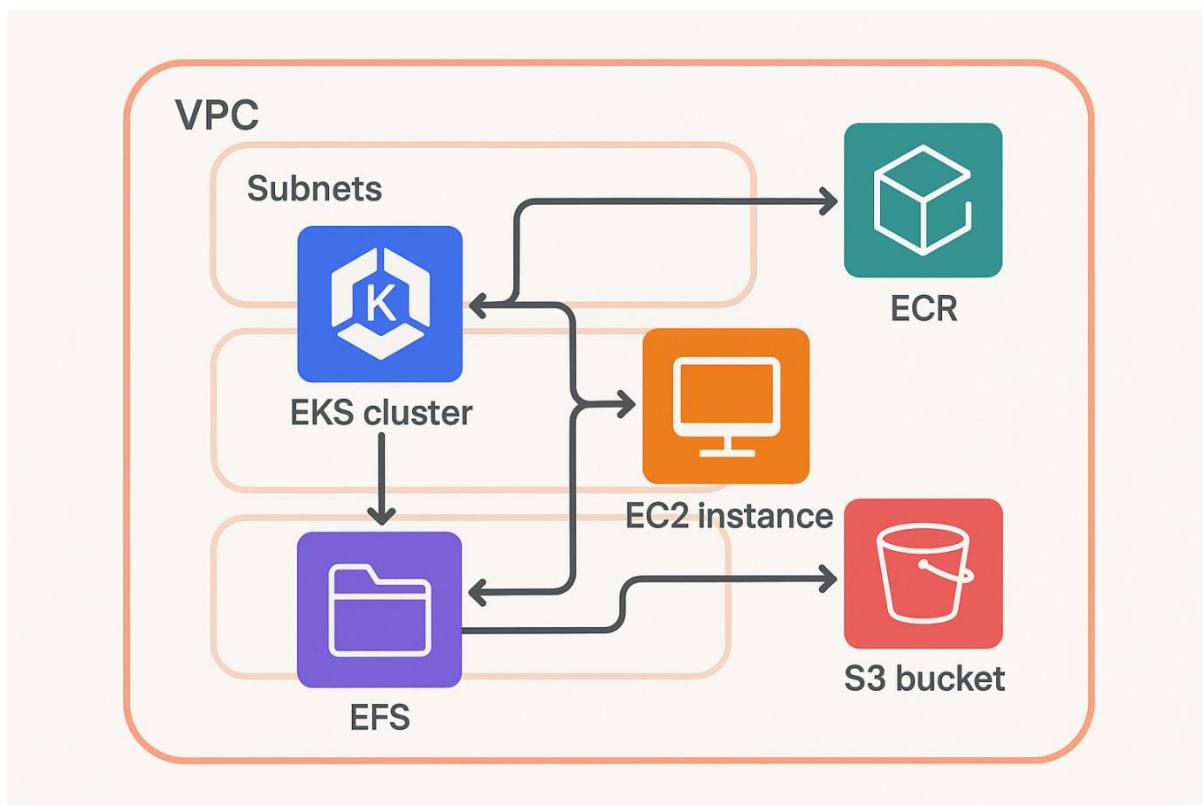
Grupos de seguridad (5) información

Q. Buscar grupos de seguridad por atributo o etiqueta

ID de la VPC = vpc-07f36107f9e2f9063 Quitar los filtros

| <input type="checkbox"/> | Name | ID de grupo de seguridad | Nombre del grupo de seguridad | ID de la VPC | Descripción | Propietario | Número de reglas de entrada |
|--------------------------|-------------------------|--------------------------|---|-----------------------|--|--------------|-----------------------------|
| <input type="checkbox"/> | - | sg-0b8b54464d5f59c5 | centrodedia-cluster-efs-sg | vpc-07f36107f9e2f9063 | SG para mount targets de EFS (abierto) | 799880597301 | 1 Entrada de permiso |
| <input type="checkbox"/> | - | sg-0220beed7ee7db3f4 | default | vpc-07f36107f9e2f9063 | default VPC security group | 799880597301 | 1 Entrada de permiso |
| <input type="checkbox"/> | - | sg-020ac248d42a6ac7b | centrodedia-cluster-nodes-sg | vpc-07f36107f9e2f9063 | SG para worker nodes de EKS (abierto) | 799880597301 | 1 Entrada de permiso |
| <input type="checkbox"/> | eks-cluster-sg-centr... | sg-01d47922ba48813e0 | eks-cluster-sg-centrodedia-cluster-480... | vpc-07f36107f9e2f9063 | EKS created security group applied to ENI T... | 799880597301 | 1 Entrada de permiso |
| <input type="checkbox"/> | - | sg-06c0bc0924282343c | centrodedia-cluster-app-sg | vpc-07f36107f9e2f9063 | SG para instancia EC2 app-server (abierto) | 799880597301 | 1 Entrada de permiso |

Diagrama de la infraestructura:



6.3 Recursos software

El sistema utiliza una combinación de tecnologías y herramientas modernas:

Sistema operativo base:

Linux (Amazon Linux 2, MySQL o Debian en los contenedores), elegido por su estabilidad, seguridad y compatibilidad con las herramientas utilizadas.

Aplicación web:

Desarrollada en PHP 8.x, utilizando Nginx como servidor web y MySQL 8.x como base de datos.

Organizada en módulos para la gestión de usuarios, trabajadores, alumnos, cursos, tareas, inventario y documentación.

Servidor FTP:

Basado en vsftpd, desplegado en un contenedor y configurado para el acceso seguro y controlado a los archivos.

Orquestación:

Kubernetes (EKS) para la gestión y el escalado de los contenedores.

Almacenamiento compartido:

EFS para la persistencia y la compartición de archivos entre servicios.

Infraestructura como código:

Terraform para la definición, despliegue y gestión de toda la infraestructura.

Automatización de despliegues:

GitHub Actions para la integración y el despliegue continuo de la aplicación y la infraestructura.

Gestión de backups:

Scripts de sincronización y cron en EC2 para la realización de copias de seguridad automáticas en S3.

7. Planificación temporal. Etapas

El desarrollo del proyecto se ha estructurado en varias fases, cada una con objetivos y entregables específicos:

Análisis y diseño (1 semana):

Estudio de los requisitos del centro y análisis de las necesidades.

Diseño de la arquitectura de la aplicación y la infraestructura.

Selección de tecnologías y herramientas.

Desarrollo de la aplicación web y base de datos (3 semanas):

Implementación de los módulos de gestión de usuarios, trabajadores, alumnos, cursos, tareas e inventario.

Diseño y creación de la base de datos relacional.

Desarrollo de formularios, validaciones y controles de acceso.

Desarrollo de la infraestructura como código (Terraform) (2 semanas):

Definición de la red, las subredes, los security groups y los recursos de AWS.

Configuración del cluster EKS, EFS, EC2 y S3.

Pruebas de despliegue y automatización.

Configuración de pipelines de CI/CD (1 semana):

Creación de workflows en GitHub Actions para la integración y el despliegue continuo.

Automatización de la construcción y subida de imágenes Docker.

Despliegue automático de los manifiestos de Kubernetes.

Pruebas y despliegue (1 semana):

Ejecución de pruebas funcionales, de seguridad y de infraestructura.

Despliegue final en el entorno de producción.

Verificación del correcto funcionamiento del sistema.

Elaboración de la documentación (1 semana):

Redacción de la memoria del proyecto.

Elaboración de los manuales de usuario y administración.

Preparación de diagramas y capturas de pantalla.

8. Ejecución del proyecto y documentación técnica

8.1 Arquitectura del sistema

La arquitectura del sistema se basa en una estructura modular y escalable, compuesta por varios servicios desplegados en contenedores sobre Kubernetes (EKS).

Cada servicio (aplicación web, servidor FTP, base de datos) se ejecuta en uno o varios pods, permitiendo su escalado y actualización independiente.

El almacenamiento persistente se gestiona mediante EFS, que permite la compartición de archivos entre los diferentes servicios y la realización de copias de seguridad automáticas en S3.

La infraestructura se define y despliega mediante Terraform, lo que permite la automatización y la reproducibilidad de todos los recursos.

La administración y el mantenimiento se realizan desde una instancia EC2, que también se encarga de la sincronización de los datos con S3 y de la monitorización del sistema.

8.2 Diseño y explicación de la base de datos

La base de datos del sistema es uno de los pilares fundamentales de la plataforma, ya que almacena toda la información necesaria para la gestión integral del centro de día.

Se ha diseñado utilizando el modelo relacional, implementado en MySQL, y estructurado en varias tablas interrelacionadas que reflejan las entidades y procesos clave del centro.

2. Tabla trabajadores

Contiene la información del personal del centro.

Campos principales:

id: Identificador único.

nombre, apellidos: Datos personales.

dni: Único, máximo 9 caracteres.

telefono: Teléfono de contacto.

puesto: Cargo o función dentro del centro.

fecha_contratacion: Fecha de inicio de contrato.

especialidad: Especialidad profesional.

fecha_nacimiento: Fecha de nacimiento.

Consideraciones:

El campo dni debe ser único.

Los trabajadores pueden estar relacionados con varios usuarios y alumnos como responsables.

3. Tabla alumnos

Registra a los alumnos que participan en cursos o actividades formativas.

Campos principales:

id: Identificador único.

nombre, apellidos, email, dni, fecha_nacimiento, telefono, direccion, localidad, fecha_ingreso.

profesor_id: Clave foránea a trabajadores (profesor responsable).

Consideraciones:

El campo email y dni deben ser únicos.

El campo profesor_id debe existir en la tabla trabajadores.

4. Tabla cursos

Define los cursos o actividades formativas ofrecidas por el centro.

Campos principales:

id: Identificador único.

nombre: Nombre del curso.

descripcion: Descripción breve.

fecha_inicio, fecha_fin: Fechas de inicio y fin.

5. Tabla asignaturas

Permite dividir los cursos en asignaturas o módulos.

Campos principales:

id: Identificador único.

nombre: Nombre de la asignatura.

codigo: Código único de la asignatura.

6. Tabla cursos_asig

Tabla intermedia para la relación muchos-a-muchos entre cursos y asignaturas.

Campos principales:

curso_id: Clave foránea a cursos.

asignatura_id: Clave foránea a asignaturas.

7. Tabla matriculas

Registra la matrícula de los alumnos en los cursos.

Campos principales:

id: Identificador único.

alumno_id: Clave foránea a alumnos.

curso_id: Clave foránea a cursos.

8. Tabla notas

Almacena las calificaciones de los alumnos en cada asignatura de cada curso.

Campos principales:

matricula_id: Clave foránea a matriculas.

asignatura_id: Clave foránea a asignaturas.

nota: Calificación numérica (0 a 10).

fecha_registro: Fecha de registro de la nota.

Consideraciones:

La combinación de matricula_id y asignatura_id debe ser única (no puede haber dos notas para la misma asignatura y matrícula).

El campo nota debe validarse para que esté entre 0 y 10.

9. Tabla tareas

Gestiona las tareas asignadas a usuarios o trabajadores.

Campos principales:

id: Identificador único.

nombre, descripcion, tipo: Información de la tarea.

10. Tabla tarea_usuario

Tabla intermedia para la relación muchos-a-muchos entre tareas y usuarios.

Campos principales:

tarea_id: Clave foránea a tareas.

usuario_id: Clave foránea a usuarios o trabajadores (según implementación).

11. Tabla inventario

Controla los materiales y recursos del centro.

Campos principales:

id: Identificador único.

nombre: Nombre del producto.

ubicacion: Ubicación física.

unidades: Número de unidades disponibles.

fecha_ingreso: Fecha de entrada en inventario.

proveedor: Proveedor del material.

estado: Estado del material (nuevo, usado, dañado).

imagen: Ruta de la imagen asociada (opcional).

Consideraciones:

El campo estado solo puede tomar valores predefinidos (validación en la aplicación).

El campo unidades debe ser un número entero positivo.

12. Tabla users

Gestiona las cuentas de acceso a la plataforma (autenticación).

Campos principales:

id: Identificador único.

nombre, email, password, ftp_password, rol.

Opciones del campo rol:

Puede ser "admin", "trabajador", "alumno" o "usuario".

Consideraciones:

El campo email debe ser único.

El campo password se almacena cifrado.

El campo rol determina los permisos y el acceso a las diferentes funcionalidades.

13. Tabla usuarios_reserva

Registra las solicitudes de inscripción realizadas a través del portal.

Campos principales:

id: Identificador único.

nombre, email, telefono, descripcion: Datos de la solicitud.

Relaciones entre tablas:

Un trabajador puede ser responsable de varios usuarios y alumnos.

Un alumno puede estar matriculado en varios cursos, y cada curso puede tener varios alumnos (relación muchos-a-muchos a través de matrículas).

Un curso puede tener varias asignaturas, y una asignatura puede estar en varios cursos (relación muchos-a-muchos a través de cursos_asig).

Cada matrícula puede tener varias notas, una por asignatura.

Las tareas pueden estar asignadas a varios usuarios y viceversa (relación muchos-a-muchos a través de tarea_usuario).

El inventario es gestionado por los trabajadores y administradores, pero no tiene relaciones directas con otras tablas.

Las cuentas de acceso (users) pueden estar asociadas a diferentes roles, lo que determina su acceso a los módulos de la aplicación.

Consideraciones generales y buenas prácticas

Integridad referencial: Todas las claves foráneas están definidas para garantizar la coherencia de los datos.

Validación de datos: Se valida la unicidad de campos como dni y email, la longitud de los campos de texto y los rangos de los campos numéricos.

Seguridad: Las contraseñas se almacenan cifradas y nunca en texto plano. Los roles y permisos se gestionan de forma estricta.

Escalabilidad: El modelo permite la incorporación de nuevas entidades y relaciones en el futuro, facilitando la ampliación de la funcionalidad.

Auditoría: Se pueden añadir campos de auditoría (fechas de creación/modificación, usuario que realiza la acción) para mejorar la trazabilidad.

8.3 Estructura de la aplicación

La aplicación web está organizada en varios módulos, cada uno con su propia lógica y formularios:

login/:

Gestión de la autenticación y el registro de usuarios.

Recuperación de contraseñas y control de sesiones.

usuarios/:

Alta, consulta, modificación y baja de usuarios.

Asignación de roles y permisos.

trabajadores/:

Gestión de los datos personales y profesionales de los trabajadores.

Asignación de tareas y seguimiento de la actividad.

alumnos/:

Gestión de los datos personales y académicos de los alumnos.

Matrícula en cursos y consulta de notas.

cursos/:

Creación y gestión de cursos, asignaturas y matrículas.

Registro y consulta de notas.

tareas/:

Creación, asignación y seguimiento de tareas.

Registro de incidencias y finalización de tareas.

inventario/:

Alta, consulta, actualización y baja de materiales y recursos.

Control de stock y generación de alertas.

ftp.php:

Acceso a la documentación y archivos mediante FTP.

Subida y descarga de archivos de forma segura.

index.php:

Portal informativo y acceso general a la plataforma.

Cada módulo cuenta con formularios intuitivos, validaciones automáticas y controles de acceso según el rol del usuario.

La interfaz está diseñada para ser clara y accesible, facilitando la navegación y la realización de tareas.

8.4 Infraestructura como código (Terraform)

Toda la infraestructura se define mediante ficheros Terraform, organizados en módulos reutilizables y parametrizables:

Red (VPC, subredes, gateway):

Define la topología de red, el acceso a Internet y la segmentación de los servicios.

Permite la integración de nuevos servicios y la ampliación de la infraestructura.

Seguridad (Security Groups):

Controla el acceso a los diferentes servicios, permitiendo solo el tráfico necesario.

Facilita la gestión de reglas y la adaptación a nuevas necesidades.

EFS:

Proporciona almacenamiento compartido y persistente para los contenedores.

Permite la compartición de archivos y la realización de copias de seguridad automáticas.

EKS:

Despliega el clúster de Kubernetes y los nodos de trabajo.

Permite el escalado automático y la recuperación ante fallos.

EC2:

Instancia para administración, backups y sincronización con S3.

Permite la ejecución de scripts de mantenimiento y la monitorización del sistema.

S3:

Almacenamiento de copias de seguridad automáticas y archivos importantes.

Permite la restauración rápida en caso de fallo o pérdida de datos.

La automatización permite desplegar, modificar o destruir toda la infraestructura de forma controlada y reproducible, facilitando la colaboración y la trazabilidad de los cambios.

8.5 Orquestación de contenedores (Kubernetes)

Los servicios principales (Nginx, PHP-FPM, MySQL, FTP) se despliegan como pods en Kubernetes, utilizando volúmenes persistentes en EFS para garantizar la persistencia de los datos y la compartición de archivos entre servicios.

El uso de deployments y services permite la escalabilidad y la alta disponibilidad de la aplicación, así como la actualización sin interrupciones.

La configuración de los recursos de Kubernetes se realiza mediante manifiestos YAML, que definen los deployments, services, persistent volume claims y storage classes necesarios para el correcto funcionamiento del sistema.

La monitorización y la gestión de los pods y servicios se realiza mediante kubectl y las herramientas nativas de AWS, permitiendo la detección y resolución rápida de incidencias.

8.6 Integración y despliegue continuo (CI/CD)

Se han configurado pipelines de CI/CD en GitHub Actions para automatizar el despliegue de la aplicación y la infraestructura.

El flujo de trabajo incluye:

Validación y aplicación de cambios en Terraform, garantizando la coherencia y la seguridad de la infraestructura.

Construcción y subida de imágenes Docker a ECR, permitiendo la actualización rápida y segura de los servicios.

Despliegue de los manifiestos de Kubernetes en EKS, asegurando la disponibilidad y la escalabilidad de la aplicación.

Notificaciones y logs de cada paso, facilitando la monitorización y la resolución de incidencias.

Esta automatización reduce los tiempos de despliegue, minimiza los errores humanos y facilita la colaboración entre diferentes administradores y desarrolladores.

8.7 Prácticas de seguridad

La seguridad es una prioridad en todo el proyecto, y se han implementado diversas medidas para garantizar la protección de los datos y la infraestructura:

Uso de prepared statements en todas las consultas SQL para evitar inyección de código.

Escapado de datos en la salida HTML para evitar ataques de cross-site scripting (XSS).

Control de acceso por roles en todas las páginas y funcionalidades, garantizando que cada usuario solo puede acceder a la información y las acciones que le corresponden.

Restricción de acceso a la infraestructura mediante Security Groups y reglas de firewall.

Cifrado de contraseñas y almacenamiento seguro de credenciales.

Copias de seguridad automáticas y almacenamiento seguro en S3, con procedimientos de restauración documentados.

Monitorización de logs y alertas para la detección temprana de posibles incidentes de seguridad.

9. Fase de pruebas

Se ha realizado una batería de pruebas exhaustiva para verificar el correcto funcionamiento del sistema en todos sus aspectos:

Pruebas funcionales:

Alta, consulta, actualización y borrado de usuarios, trabajadores, alumnos, cursos, tareas e inventario.

Inscripción online y gestión de matrículas.

Acceso y descarga de documentación vía FTP.

Acceso restringido según rol y validación de permisos.

Pruebas de navegación y usabilidad en diferentes dispositivos y navegadores.

Pruebas de seguridad:

Verificación de la protección contra inyección SQL mediante la introducción de datos maliciosos en los formularios.

Comprobación de la protección contra XSS mediante la inserción de scripts en los campos de texto.

Pruebas de gestión de sesiones y contraseñas, incluyendo el cambio y la recuperación de contraseñas.

Simulación de intentos de acceso no autorizado y verificación de la respuesta del sistema.

Pruebas de infraestructura:

Despliegue automático de la infraestructura con Terraform y verificación de la creación de todos los recursos.

Escalado de pods en Kubernetes y comprobación de la alta disponibilidad.

Simulación de fallos en los nodos y verificación de la recuperación automática.

Restauración de copias de seguridad desde S3 y comprobación de la integridad de los datos.

Pruebas de rendimiento:

Medición de los tiempos de respuesta de la aplicación bajo diferentes cargas.

Comprobación del uso de recursos y la escalabilidad de los servicios.

10. Documentación del sistema

10.1 Introducción a la aplicación

La aplicación desarrollada permite la gestión integral del centro de día, centralizando todos los procesos administrativos y asistenciales en una única plataforma accesible desde cualquier lugar.

La interfaz está diseñada para ser intuitiva y accesible, facilitando la navegación y la realización de tareas tanto para usuarios experimentados como para personas con menos experiencia tecnológica.

La plataforma está organizada en módulos independientes, lo que facilita su mantenimiento y la incorporación de nuevas funcionalidades en el futuro.

La seguridad y la protección de los datos son una prioridad, y se han implementado diversas medidas para garantizar la confidencialidad, la integridad y la disponibilidad de la información.

10.2 Manual de instalación

Requisitos previos:

Cuenta de AWS con permisos suficientes para crear y gestionar recursos (VPC, EKS, EFS, EC2, S3, ECR, IAM).

Git, Terraform CLI, kubectl y AWS CLI instalados en el equipo de administración.

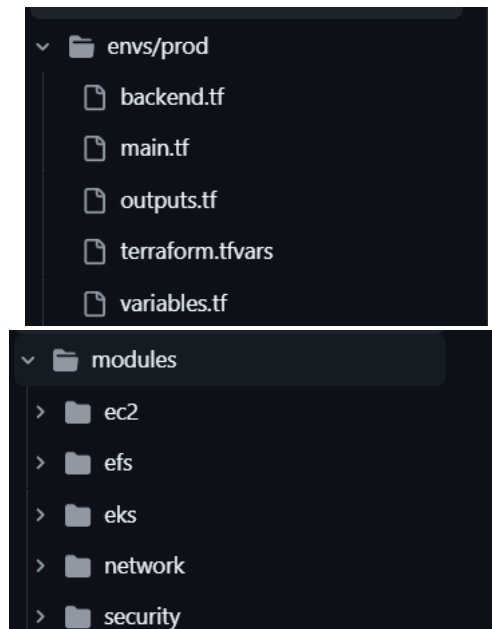
Acceso a los repositorios de código y a los secretos necesarios (tokens de GitHub, credenciales de AWS, etc.).

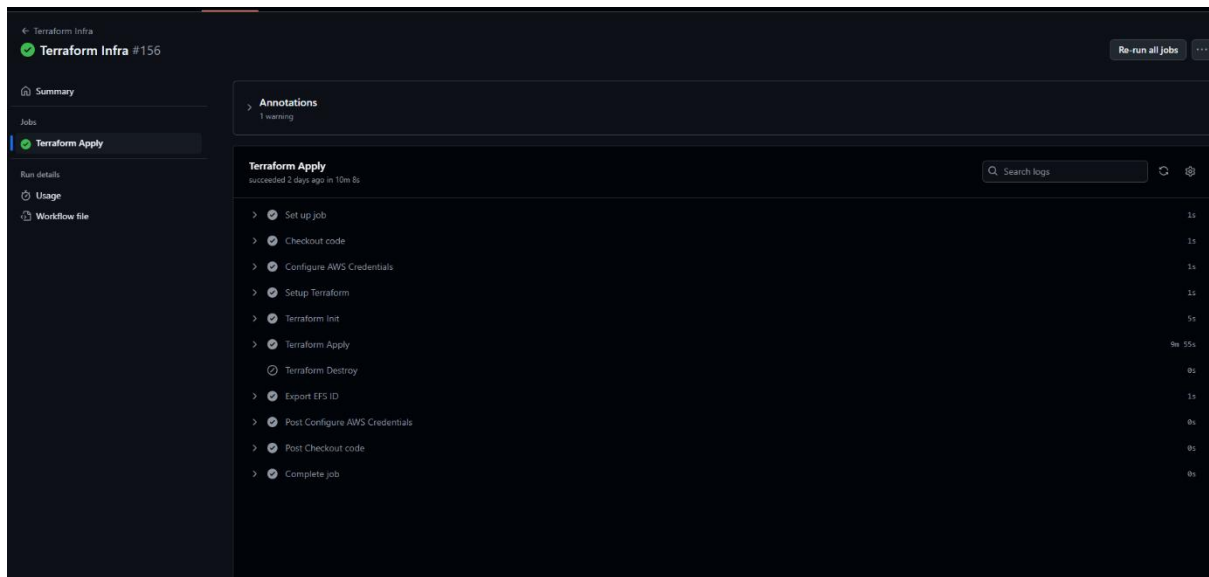
Pasos de instalación:

Clonar el repositorio del proyecto desde GitHub.

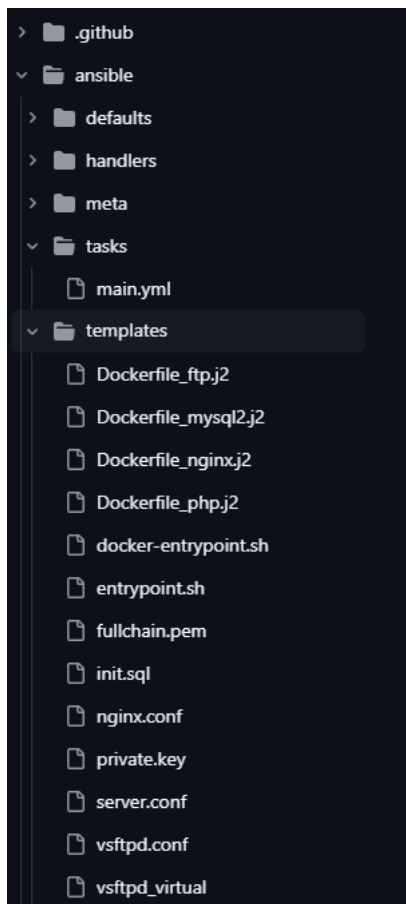
Configurar las variables y credenciales necesarias en los archivos de configuración y en los workflows de GitHub Actions.

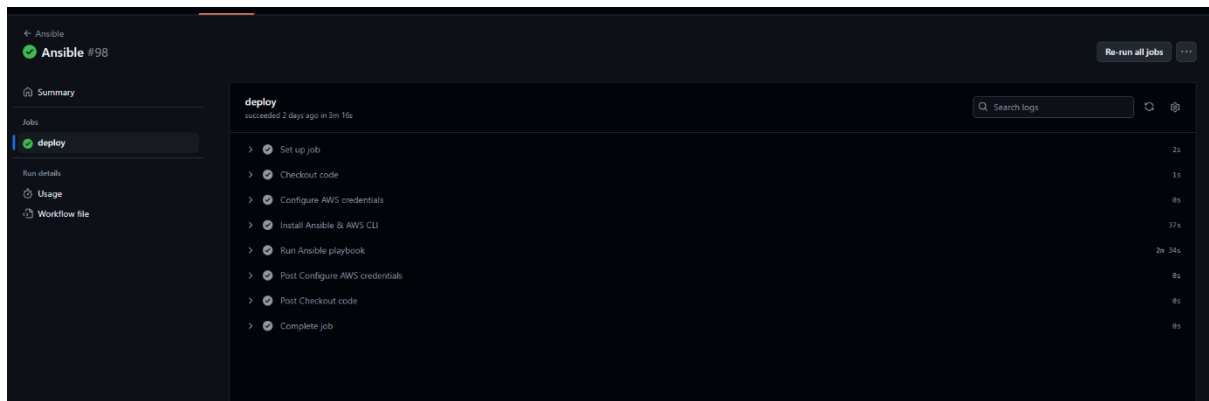
Ejecutar Terraform para desplegar la infraestructura, creando la red, el cluster EKS, EFS, EC2 y S3.



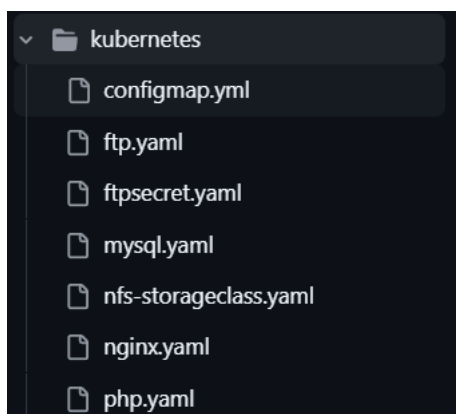


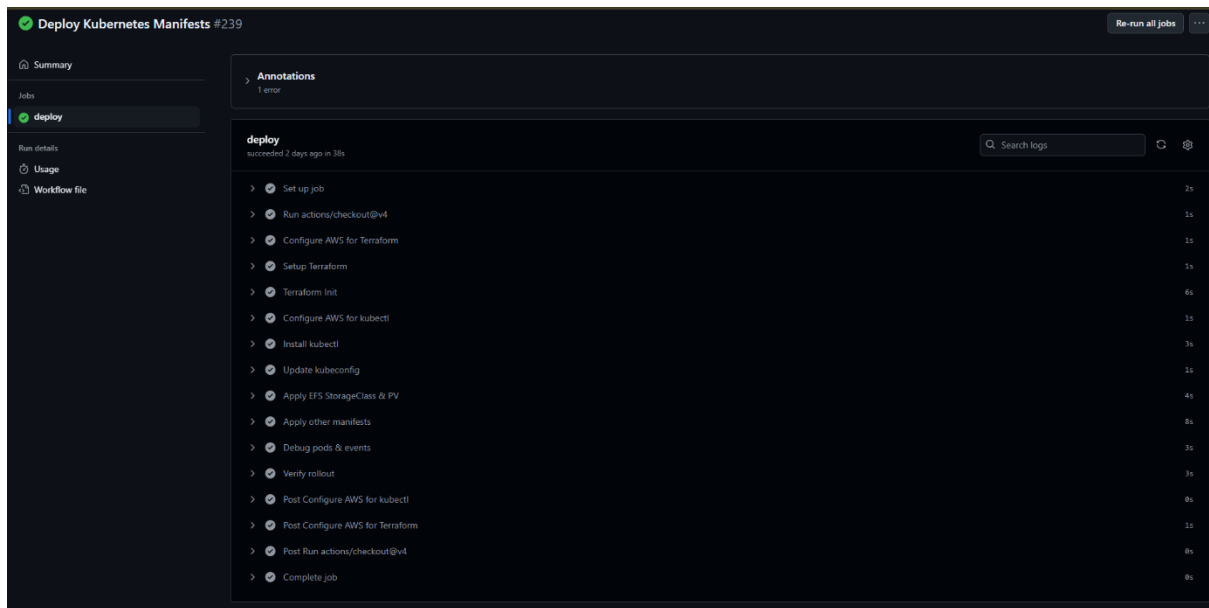
Construir y subir las imágenes Docker a ECR utilizando los scripts y workflows proporcionados.



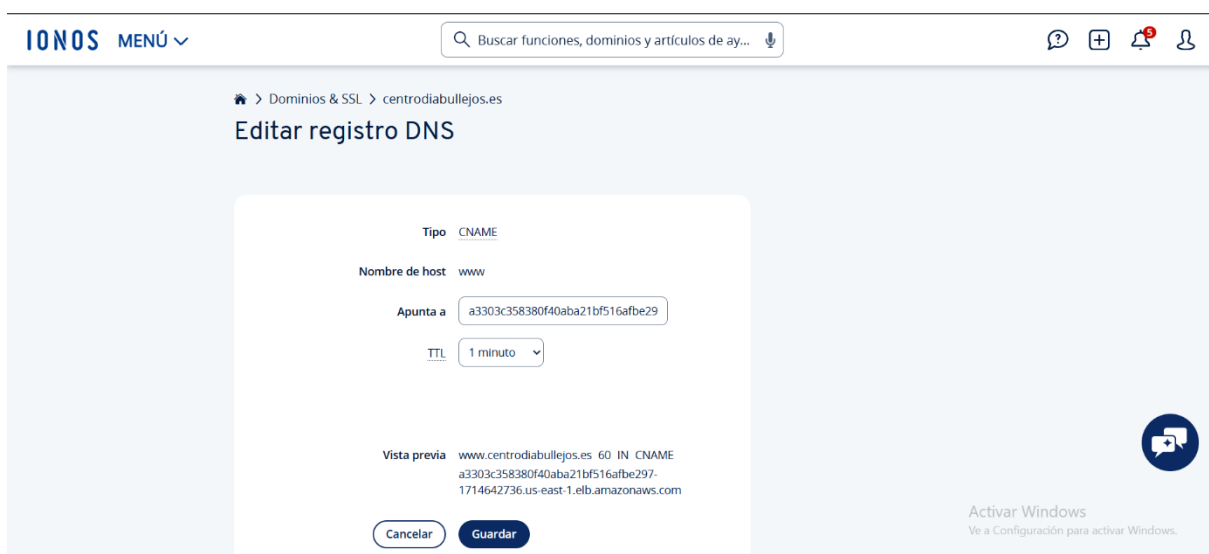


Desplegar los manifiestos de Kubernetes en EKS, creando los deployments, services, persistent volume claims y storage classes necesarios.





Unir IP externa que ofrece el servicio de NGINX al dominio para obtener una IP válida y profesional.



Verificar el acceso a la aplicación y a los servicios asociados, comprobando el correcto funcionamiento de todos los módulos.

```
[root@ip-10-0-1-162 mnt]# kubectl get all
```

| NAME | READY | STATUS | RESTARTS | AGE |
|---|-------|---------|----------|-------|
| pod/carlos-web-deployment-d6b96cd49-mq75t | 1/1 | Running | 0 | 2m32s |
| pod/carlos-web-deployment-d6b96cd49-vxdpl | 1/1 | Running | 0 | 2m32s |
| pod/ftp-server-7d8d8fc8b6-7jdb2 | 1/1 | Running | 0 | 2m34s |
| pod/ftp-server-7d8d8fc8b6-9d9bh | 1/1 | Running | 0 | 2m34s |
| pod/mysql-deployment-87d8d8fb7-ff7wb | 1/1 | Running | 0 | 2m32s |
| pod/php-fpm-deployment-65f8bb8946-4gzbp | 1/1 | Running | 0 | 2m27s |

| NAME | TYPE | CLUSTER-IP | EXTERNAL-IP | PORT(S) | AGE |
|-------------------------|--------------|----------------|---|------------------------------|-------|
| service/kubernetes | ClusterIP | 172.20.0.1 | <none> | 443/TCP | 16m |
| service/mysql-service | ClusterIP | 172.20.166.66 | <none> | 3306/TCP | 2m32s |
| service/nginx-service | LoadBalancer | 172.20.14.137 | a9303c358380f40aba21bf516afbe297-1714642736.us-east-1.elb.amazonaws.com | 80:30524/TCP, 443:31924/TCP | 2m32s |
| service/php-fpm-service | ClusterIP | 172.20.157.23 | <none> | 9000/TCP | 2m27s |
| service/vsftpd | ClusterIP | 172.20.121.236 | <none> | 21/TCP, 21100/TCP, 21101/TCP | 2m34s |

| NAME | READY | UP-TO-DATE | AVAILABLE | AGE |
|---------------------------------------|-------|------------|-----------|-------|
| deployment.apps/carlos-web-deployment | 2/2 | 2 | 2 | 2m32s |
| deployment.apps/ftp-server | 2/2 | 2 | 2 | 2m34s |
| deployment.apps/mysql-deployment | 1/1 | 1 | 1 | 2m32s |
| deployment.apps/php-fpm-deployment | 1/1 | 1 | 1 | 2m27s |

| NAME | DESIRED | CURRENT | READY | AGE |
|---|---------|---------|-------|-------|
| replicaset.apps/carlos-web-deployment-d6b96cd49 | 2 | 2 | 2 | 2m32s |
| replicaset.apps/ftp-server-7d8d8fc8b6 | 2 | 2 | 2 | 2m34s |
| replicaset.apps/mysql-deployment-87d8d8fb7 | 1 | 1 | 1 | 2m32s |
| replicaset.apps/php-fpm-deployment-65f8bb8946 | 1 | 1 | 1 | 2m27s |

```
[root@ip-10-0-1-162 mnt]#
```

Configurar las copias de seguridad automáticas y verificar su funcionamiento.

10.3 Manual de usuario

Acceso a la plataforma:

Los usuarios pueden acceder al portal web desde cualquier navegador, introduciendo la URL proporcionada por el centro.

El registro y la autenticación se realizan mediante email y contraseña, con validaciones automáticas y recuperación de contraseñas en caso de olvido.

Funcionalidades principales:

Usuarios no registrados:

Consultar información general del centro, incluyendo servicios, actividades, instalaciones y contacto.

Nuestros Servicios



[Actividades recreativas para un día pleno.](#)



[Apoyo emocional e integración social.](#)



[Talleres y clases para desarrollo profesional.](#)

Galería de Actividades



Nuestras Instalaciones



Instalaciones

Visita nuestras modernas salas de terapia, talleres, zonas de descanso y mucho más. ¡Descúbrelas!

[Ver instalaciones →](#)

LOCALIZACIÓN Y CONTACTO



Dirección

Centro de Día Bullejos
Calle Santa Rosa, nº 10, 18200 Maracena (Granada)



Teléfonos de contacto

Teléfono: **958 426 906**
Teléfono 2: **627 729 999**



Email de contacto

contacto@centrodiabullejos.es



Realizar solicitudes de inscripción mediante un formulario online.

Formulario de Reserva

Nombre:

Email:

Teléfono:

Descripción:

Enviar

Restablecer

Alumnos:

Consultar sus datos personales y académicos, incluyendo matrículas, cursos y notas.

Mis Notas

| Curso | Asignatura | Nota | Fecha |
|----------|-------------------|-------|------------|
| Diabetes | Tipos de diabetes | 10.00 | 2025-06-02 |

[🔍 Volver al inicio](#)


Solicitar matrícula en nuevos cursos y recibir notificaciones sobre el estado de sus solicitudes.

Descargar documentación y materiales de estudio desde el portal o mediante FTP.

Gestión de Archivos

Directorio actual: /Guías

[← Volver al directorio anterior](#)

 FTP FUNCIONAL.txt



Trabajadores:

Gestionar alumnos, cursos, tareas e inventario desde un panel de administración intuitivo.

Asignar tareas a otros trabajadores o alumnos y realizar el seguimiento de su ejecución.

Consultar y actualizar información relevante sobre los usuarios y los recursos del centro.

Manejar documentación de cursos.

Lista de Inventario

Buscar

| ID | Nombre del Producto | Ubicación | Unidades | Fecha de Ingreso | Proveedor | Estado | Imagen | Borrar | Actualizar |
|----|---------------------|-----------------------|----------|------------------|-----------|--------|---|---|---|
| 1 | Camilla | Salón de fisioterapia | 3 | 2025-06-01 | Ninguno | Nuevo |  |  |  |

Agregar Nuevo Registro

Lista de Tareas

Buscar

| ID | Nombre | Descripción | Tipo | Usuarios | Borrar | Actualizar |
|----|--------|-------------------------------------|-----------------|-----------------|---|---|
| 2 | Yoga | Ejercicios de 1 hora haciendo yoga. | Física y mental | Carlos Bullejos |  |  |

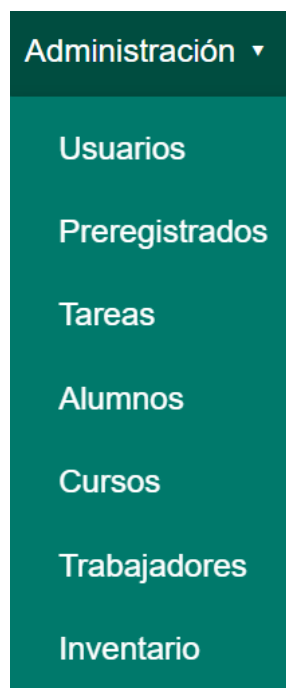
Agregar Nueva Tarea



10.4 Manual de administración

Gestión de usuarios y roles:

Acceso al panel de administración para la gestión de usuarios, trabajadores y alumnos.



Asignación y modificación de roles y permisos, garantizando el acceso restringido a las funcionalidades según el perfil del usuario.

Gestión de Usuarios

✓ Usuario añadido correctamente.

+ Añadir nuevo usuario

| Nombre | Email | Rol actual | Cambiar rol | Eliminar |
|--------|-----------------------------|------------|---------------------------------------|---------------------|
| admin | admin@centrodiabullejos.es | admin | (Yo) | — |
| Carlos | carlos@centrodiabullejos.es | usuario | <div>Usuario</div> <div>Cambiar</div> | <div>Eliminar</div> |

Validación y actualización de los datos personales y profesionales de los usuarios.

Alta de Usuario

Nombre:

Apellidos:

DNI:

Fecha de nacimiento:

dd / mm / aaaa

Teléfono:

Dirección:

Localidad:

Fecha de ingreso:

dd / mm / aaaa

Dependencia:

Patologías:

ID de Trabajador:

Enviar

Restablecer

Actualizar Usuario

Nombre:

Carlos

Apellidos:

Bullejos

DNI:

75932483f

Fecha de nacimiento:

01/05/2025

Teléfono:

647072737

Dirección:

Santa Rosa

Localidad:

maracena

Fecha de ingreso:

01/05/2025

Dependencia:

Grado I

Patologías:

ninguna

ID de Trabajador:

2

Actualizar

Restablecer

Lista de Usuarios

Buscar por nombre o apellido

Buscar

| ID | Nombre | Apellidos | DNI | Fecha Nacimiento | Teléfono | Dirección | Localidad | Fecha Ingreso | Dependencia | Patologías | Trabajador ID | Borrar | Actualizar |
|----|--------|-----------|-----------|------------------|-----------|------------|-----------|---------------|-------------|------------|---------------|--------|------------|
| 4 | Carlos | Bullejos | 75932483f | 1995-09-07 | 647072737 | Santa Rosa | maracena | 2025-05-01 | Grado I | ninguna | 2 | | |

Agregar Nuevo Usuario

Gestión de cursos y matrículas:

Creación y edición de cursos, asignaturas y matrículas, con validaciones automáticas y control de integridad.

Crear Asignatura

Nombre:

Código:

[Cancelar](#)

Crear Curso

Nombre:

Descripción:

Fecha Inicio:

Fecha Fin:


[Cancelar](#)

| Asignaturas del curso "Diabetes" | | |
|--|-------------------|--------|
| + Gestionar asignaturas | | |
| | Nombre | Código |
| <input checked="" type="checkbox"/> | Tipos de diabetes | 1 |
| <input type="button" value="Guardar asignaciones"/> Volver | | |

Gestión de notas y seguimiento académico de los alumnos.

Notas del curso “Diabetes”



| Alumno | Tipos de diabetes |
|-----------------|------------------------------------|
| Carlos Bullejos | <input type="text" value="10,00"/> |

 Guardar Notas  Volver

Control de la inscripción y la asistencia a los cursos.

Matricular alumnos en “Diabetes”

| | Nombre | Apellidos |
|-------------------------------------|--------|-----------|
| <input checked="" type="checkbox"/> | Carlos | Bullejos |

 Guardar Matriculas  Volver

Gestión de tareas e inventario:

Alta, consulta, actualización y borrado de inventario, con registro de todas las acciones realizadas.

Formulario de Inventario

Nombre del Producto:

Camilla

Ubicación:

Salón de fisioterapia

Unidades:

Fecha de Ingreso:

01/06/2025



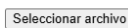
Proveedor:

Ninguno

Estado:

Nuevo

Imagen del Producto:

 Seleccionar archivo camilla.jpg




Enviar

Restablecer

Lista de Inventario

Buscar por nombre

Buscar

| ID | Nombre del Producto | Ubicación | Unidades | Fecha de Ingreso | Proveedor | Estado | Imagen | Borrar | Actualizar |
|----|---------------------|-----------------------|----------|------------------|-----------|--------|---|---|---|
| 1 | Camilla | Salón de fisioterapia | 3 | 2025-06-01 | Ninguno | Nuevo |  |  |  |

Agregar Nuevo Registro

Asignación de tareas a usuarios y trabajadores, con seguimiento del estado y la finalización de las mismas.

Formulario de Tareas

Nombre:

Yoga

Descripción:

Ejercicios de 1 hora
haciendo yoga.

Tipo:

Física y mental

Asignar Usuarios:

Carlos Bullejos

Enviar

Restablecer

Lista de Tareas

Buscar por nombre de tarea

Buscar

| ID | Nombre | Descripción | Tipo | Usuarios | Borrar | Actualizar |
|----|--------|-------------------------------------|-----------------|-----------------|---|---|
| 2 | Yoga | Ejercicios de 1 hora haciendo yoga. | Física y mental | Carlos Bullejos |  |  |

Agregar Nueva Tarea

Gestión de copias de seguridad y restauración:

Acceso a las copias de seguridad almacenadas en S3, con procedimientos documentados para su restauración en caso de fallo o pérdida de datos.

Verificación periódica del correcto funcionamiento de las copias de seguridad y de la integridad de los datos almacenados.

Monitorización y mantenimiento:

Supervisión del estado de los pods y servicios en Kubernetes mediante kubectl y las herramientas nativas de AWS.

Escalado y actualización de los servicios de forma controlada y sin interrupciones.

Revisión de logs y alertas para la detección temprana de posibles incidencias o ataques.

Actualización de la aplicación y la infraestructura mediante los workflows de CI/CD, garantizando la trazabilidad y la reversibilidad de los cambios.

Buenas prácticas de administración:

Realizar copias de seguridad antes de cualquier actualización o cambio importante.

Documentar todas las acciones realizadas y mantener un registro actualizado de la configuración del sistema.

Revisar periódicamente los logs y las alertas para detectar posibles problemas o intentos de acceso no autorizado.

Mantener actualizadas las credenciales y los permisos de los usuarios, eliminando los accesos innecesarios o caducados.

11. Conclusiones finales

El proyecto ha permitido modernizar y centralizar la gestión del centro de día, mejorando la eficiencia, la seguridad y la accesibilidad de la información para todos los usuarios.

La utilización de tecnologías cloud y prácticas de automatización ha facilitado el despliegue, la administración y la recuperación ante fallos, alineándose con las competencias profesionales del ciclo ASIR y con las tendencias actuales en la administración de sistemas.

La plataforma desarrollada es flexible, escalable y fácil de mantener, permitiendo su adaptación a las necesidades cambiantes del centro y la incorporación de nuevas funcionalidades en el futuro.

La documentación elaborada facilita la formación de nuevos usuarios y administradores, garantizando la sostenibilidad y la continuidad del sistema a largo plazo.

12. Grado de cumplimiento de los objetivos fijados

Todos los objetivos funcionales, técnicos y operativos planteados al inicio del proyecto se han cumplido satisfactoriamente.

La plataforma cumple con los requisitos de centralización, automatización, seguridad, escalabilidad y facilidad de administración, permitiendo la gestión integral del centro de día de forma eficiente y segura.

13. Propuesta de modificaciones o ampliaciones futuras

Integración de un sistema de notificaciones por email o SMS para mejorar la comunicación con los usuarios y familiares.

Implementación de un sistema de monitorización avanzada, utilizando herramientas como Prometheus y Grafana para la supervisión en tiempo real de la infraestructura y los servicios.

Mejora de la interfaz de usuario y adaptación a dispositivos móviles, utilizando frameworks modernos de frontend.

Integración con sistemas externos, como servicios de salud o administración pública, para facilitar la interoperabilidad y el intercambio de información.

Automatización de pruebas y despliegues aún más avanzada, incluyendo pruebas automatizadas de seguridad y rendimiento.

Incorporación de nuevas funcionalidades, como la gestión de eventos, encuestas de satisfacción o módulos de formación online.

14. Bibliografía empleada

Documentación oficial de AWS: <https://docs.aws.amazon.com/>

Documentación de Terraform: <https://www.terraform.io/docs/>

Documentación de Kubernetes: <https://kubernetes.io/docs/>

Manual de PHP: <https://www.php.net/manual/es/>

Manual de MySQL: <https://dev.mysql.com/doc/>

Documentación de Docker: <https://docs.docker.com/>

Documentación de GitHub Actions: <https://docs.github.com/en/actions>