

UNIVERSIDAD DON BOSCO

Facultad de



Ingeniería

Lenguajes Interpretados en el Servidor

Ciclo 01-2025

GRUPO TEORICO: 01T

TEMA:

Proyecto LIS Clínica Dental.

NOMBRE DEL DOCENTE:

Carlos Eduardo Cortez Esperanza

INTEGRANTES Y ROLES:

David Luciano Raymundo Eguizábal(Líder proyecto).

José Ricardo Aguilar Valdez.

Felipe Alberto Salguero Quezada.

Rodrigo Ignacio Martínez Pérez.

Emanuel Edgardo Luna Villanueva.

Carlos Fabricio Cornejo Miranda.

FECHA DE ENTREGA:

17 de marzo del 2025.

Índice:

3. Descripción del proyecto	_____
4.1. Gestion de la Integración	_____
4.2. Gestion del Alcance	_____
4.3. Gestion del Cronograma	_____
4.4. Gestion de Costos	_____
4.5. Gestion de la Calidad	_____
4.6. Gestion de los Recursos Humanos	_____
4.7. Gestion de la Comunicación	_____
4.8. Gestion de Adquisición y Cierre	_____
5.1. Diseño de la Aplicación	_____
5.2. Arquitectura y Contenedorización	_____
5.3. Gestion de la Base de Datos	_____
6. Desarrollo del Proyecto	_____
7. Pruebas y Control de Calidad	_____
8. Resultados y Control de Calidad	_____
9. Anexos y Documentación de Apoyo	_____

3. Descripción del proyecto.

- **Identificación de la Empresa**

Nombre de la Empresa: Clínica dental Sonrisas.

Giro: PRIMARIA: SERVICIOS DE ODONTOLOGÍA.

Ubicación: San Salvador.

Antecedentes: Esta fue fundada en el año 2010, la Clínica tiene como objetivo brindar un servicio de calidad dándole un enfoque humanístico a través de los diferentes tratamientos dentales.

Contacto: +503 2513-3309 Patricia Eguizábal.

Correo electrónico: pattyeguizabal@yahoo.com

- **Nombre del Proyecto**

Sistema de citas dentales online.

Para este proyecto se desarrollará una aplicación web para la gestión eficiente de las citas dentales, permitiendo así a la odontóloga encargada reservar, modificar o cancelar las citas de sus pacientes de forma online. Esta propuesta busca optimizar la organización del tiempo de la odontóloga encargada por medio de una plataforma accesible y escalable.

- **Problema a Resolver**

En la actualidad, la clínica maneja sus citas de manera manual, usado ya sea agendas físicas y llamadas telefónicas, lo que a largo plazo genera ciertos problemas de organización, errores en la programación de citas o pérdida de citas. Esto lleva a una experiencia un tanto deficiente para los pacientes debido a las dificultades para gestionar las citas de manera conveniente.

La falta de un sistema digital impide la automatización de recordatorios, lo que aumenta la tasa de ausencias y afecta la eficiencia operativa de la clínica. Por eso el sistema digital es fundamental para brindarle a el odontólogo y el equipo administrativo la mejora de gestión del flujo de trabajo y optimizar la disponibilidad de horarios.

- **Objetivo General**

Desarrollar una aplicación web para la **gestión eficiente de citas**, permitiendo al odontólogo y el equipo administrativo poder reservar, modificar y cancelar citas en línea.

- **Objetivos Específicos**

Implementar un sistema digital que permita la gestión eficiente de citas dentales, incluyendo la administración de usuarios, la reserva, modificación y cancelación de citas en tiempo real, garantizando seguridad y protección de datos mediante autenticación y encriptación, además de contenerizar la aplicación con Docker y desplegarla en Kubernetes para asegurar su escalabilidad y disponibilidad.

- **Metas**

1. Solución que el sistema dará al problema

El doctor/a podrá crear y citas, además de poder consultar el historial de citas de los paciente creadas previamente, y específicamente, identificar la última cita registrada. También podrá ver qué procedimiento médico que se llevó a cabo en esa ocasión, la fecha y hora exacta de la consulta, el diagnóstico y el tratamiento recomendado. Adicionalmente, podrá visualizar el costo total del procedimiento realizado y el detalle de cualquier pago efectuado, incluyendo el monto, la fecha de pago y el método utilizado.

2. Accesibilidad y Usabilidad

Diseñar una interfaz intuitiva que permita a la odontóloga y al equipo administrativo gestionar citas.

3. Seguridad y Protección de Datos

Garantizar que la información personal de los pacientes es almacenada de manera segura mediante procesos seguros.

4. Prueba y mejora

Se realizarán pruebas y se harán los ajustes necesarios durante los primeros dos meses de uso del sistema, tomando en cuenta las observaciones de la odontóloga y el equipo administrativo.

5. Capacitación del Personal

Capacitar al personal de la clínica para el uso y el manejo adecuado del sistema de citas en línea.

Completar estas capacitaciones en un lapso de 1-2 semanas para tener el suficiente tiempo para que se familiaricen y así evitar errores o confusiones.

• **Resultados Esperados**

Que la aplicación web se encargue totalmente de la gestión de citas y que las citas no sea más un problema para la clínica, teniendo las ventajas de que las citas se puedan consultar con más facilidad además de programarse de manera más eficiente ya que se evitará que las citas se puedan solapar y en general será más óptimo.

- Vista accesible para consultar las citas pendientes.
- Una forma más fácil de agendar citas.
- Evita el posible error humano de programar dos citas a la misma hora el mismo día.
- Un historial de las citas ya realizadas.
- Una menor carga para la persona encargada de agendar citas.

• **Metodología**

Se utilizará el modelo ágil de Kanban, ya que ayuda en la visualización del trabajo para que pueda entenderse mejor, mostrarse a otros y mantener actualizados a los interesados. De esta manera podemos asegurarnos de eliminar procesos y prácticas innecesarias además de un flujo continuo de todo el proceso de la aplicación web.

• **Área Geográfica y Beneficiarios del Proyecto**

El área metropolitana de San Salvador, Colonia Vista Hermosa Avenida los Helechos, este proyecto beneficiará tanto a la odontóloga como a los pacientes facilitando así la programación de sus citas.

- **Equipo Participante**

Enumere los nombres y roles de los actores clave involucrados en el proyecto, incluyendo a los responsables de la ejecución y cualquier colaborador importante.

Luciano Raymundo - Líder de trabajo, Programador.

Emanuel Luna - Diseño de la aplicación web, Programador.

Ricardo Aguilar - Coordinador de asignaciones de trabajo, Programador.

Rodrigo Martínez - Programador

Carlos Cornejo - Programador

Felipe Salguero - Programador

- **Posibles Fuentes de Financiamiento**

Fondos Propios de la Clínica

La clínica puede usar sus propios recursos financieros para invertir en la optimización de sus operaciones, como la implementación de un sistema digital para la gestión de citas. Esto refleja un fuerte compromiso con la innovación y la eficiencia.

- **Factores Críticos de Éxito**

Los factores que podrían afectar el éxito de este sistema de citas dentales online pueden ser varios y de diferente tipo como, por ejemplo:

Problemas técnicos: Fallos en el sistema o errores en la reservación de citas. Solución: Pruebas exhaustivas, servidores redundantes y monitoreo en tiempo real para detectar y solucionar problemas rápidamente.

Brechas de seguridad: Riesgo de filtración de datos sensibles de los usuarios. Solución: Cumplir con normativas de privacidad, encriptación de datos y autenticación multifactorial.

Falta de sincronización de agendas: Posibles dobles reservas o errores de horarios.

Solución: Integración con sistemas existentes de gestión de citas y actualización en tiempo real.

Baja fiabilidad en pagos: Problemas con el procesamiento de pagos. Solución: Usar plataformas confiables como PayPal y realizar pruebas de seguridad en el sistema de pagos.

Experiencia de usuario deficiente: Si el sistema es complicado, los usuarios lo abandonarán.

Solución: Diseño centrado en el usuario y pruebas de usabilidad constantes.

Competencia y saturación del mercado: Dificultad para destacar entre otros sistemas similares.

Solución: Diferenciación con funcionalidades exclusivas y una estrategia de marketing bien dirigida.

Falta de soporte adecuado: Problemas de atención al cliente.

Solución: Soporte multicanal rápido y sección de FAQs y tutoriales completos.

Cambios regulatorios: Modificaciones legales que afecten al sistema.

Solución: Monitoreo constante de normativas y consultoría legal especializada.

Alta tasa de cancelaciones: Pérdida de ingresos y dificultades en la planificación.

Solución: Penalizaciones por cancelaciones tardías y envío de recordatorios frecuentes.

4.1. Gestión de la Integración.

En cuanto a la integración de Equipos y herramientas se usará GitHub para el control de versiones, será la herramienta principal para el control de versiones, permitiendo un manejo eficiente del código fuente, la colaboración entre los miembros del equipo y la integración continua. La estructura de repositorio será la siguiente:

Branches principales: Se utilizarán ramas main (producción) y develop (desarrollo), donde main representará el código listo para producción y develop contendrá las últimas actualizaciones y nuevas funcionalidades.

Ramas de características: Cada nueva funcionalidad o corrección de error se desarrollará en ramas específicas (por ejemplo, feature/login, feature/calendar), las cuales se fusionarán con la rama develop mediante pull requests. Esto asegura que el código nuevo sea revisado antes de su integración.

Revisión de código (Code Review): Los pull requests estarán sujetos a revisiones por parte de otros desarrolladores antes de ser fusionados con develop. Esta práctica ayudará a mantener la calidad del código, detectar errores y mejorar las mejores prácticas.

En cuanto al seguimiento del proyecto y gestión de tareas usaremos Trello, específicamente se utilizará para la gestión de tareas diarias, asignación de tareas y el seguimiento de avances mediante columnas como "Por hacer", "En progreso" y "Completadas". Cada tarjeta incluirá información detallada sobre la tarea y su progreso, y se asignará a un miembro del equipo.

Como metodología ágil, el proyecto va a ser gestionado por la metodología Scrum que permitirá una entrega iterativa y continua de funcionalidades. Las actividades clave del Scrum son:

- **Sprints:** Cada sprint será de dos semanas, donde se definirá un conjunto de funcionalidades que deben completarse. Al final del sprint, se realizará una revisión con el equipo y las partes interesadas.
- **Daily Standups:** Reuniones diarias de 15 minutos donde cada miembro del equipo discutirá lo que ha hecho, lo que hará a continuación y si tiene algún bloqueo.
- **Sprint Reviews y Retrospectives:** Al final de cada sprint, el equipo se reunirá para revisar los avances y discutir cómo mejorar en el próximo sprint.

Para el entorno de contenedores usaremos Docker para crear contenedores que faciliten el desarrollo y el despliegue de la aplicación de citas en línea. La configuración de Docker incluirá:

- **Dockerfile:** Se creará un archivo Dockerfile que especificará cómo construir la imagen de la aplicación. Esto incluirá la instalación de dependencias, la configuración de variables de entorno y la exposición de puertos.
- **Docker Compose:** Para manejar múltiples servicios (como el servidor de la aplicación, base de datos, y cualquier otro microservicio necesario), se utilizará Docker Compose. Esto facilitará la ejecución de la aplicación en un entorno local de desarrollo con varios contenedores interconectados.

Mientras que Kubernetes se implementará como sistema de orquestación de contenedores para gestionar el despliegue, escalado y mantenimiento de la aplicación en producción. La configuración de Kubernetes incluirá:

- **Pods y Deployments:** Los contenedores se agruparán en **Pods**, y se gestionarán mediante **deployments** para asegurar que haya siempre una cantidad mínima de instancias corriendo.
- **Escalabilidad:** Kubernetes facilitará el escalado automático de los pods en función de la demanda. Esto asegurará que la aplicación pueda manejar un número variable de usuarios sin problemas de rendimiento.
- **Servicios:** Kubernetes se encargará de exponer la aplicación a través de **Servicios** que permitirán el acceso a los usuarios finales, manejando la distribución de tráfico de forma eficiente.
- **Despliegue Continuo:** Con herramientas como **Helm**, se facilitará el proceso de despliegue continuo, permitiendo actualizaciones sin tiempo de inactividad.

En cuanto a la entrega coordinada esta se gestionará bajo un plan de entregas coordinadas. El coordinador del equipo de desarrollo será David Luciano Raymundo Eguizabal, quien se encargará de supervisar el avance del proyecto, asegurar que las entregas se realicen en tiempo y forma, y garantizar que los sprints se mantengan enfocados en los objetivos acordados.

El coordinador será responsable de:

- **Gestión de los Sprints:** Asegurará que el equipo cumpla con las metas establecidas para cada sprint y facilitará la resolución de bloqueos o problemas que puedan surgir.
- **Revisión de las entregas:** Al final de cada sprint, el coordinador liderará las revisiones con las partes interesadas para asegurar que el producto está alineado con las expectativas.

- **Planificación de entregas:** Trabajaré con el Product Owner y los desarrolladores para planificar futuras entregas, ajustando la carga de trabajo en función del progreso.

4.2. Gestión del Alcance.

En esta aplicación de citas en línea para la clínica dental contará con las siguientes funcionalidades

- Autenticación de usuarios
 - **Registro y Login:** Los usuarios deben poder registrarse mediante un formulario de registro y posteriormente acceder a la plataforma con sus credenciales (usuario y contraseña). La autenticación debe ser segura utilizando tecnologías como JWT (JSON Web Tokens) para mantener la sesión activa de forma segura.
 - **Recuperación de contraseña:** Los usuarios deben poder recuperar su contraseña mediante un proceso de restablecimiento basado en su correo electrónico.
- **Roles de Usuarios y Permisos:**
 - **Paciente:** Los pacientes podrán realizar citas en línea, consultar sus citas futuras, ver su historial de citas y actualizar su información personal.
 - **Administrador:** Los administradores tendrán permisos para gestionar las citas de los pacientes, asignar tareas a los empleados, y supervisar la disponibilidad de los médicos. Además, podrán gestionar los usuarios y las configuraciones del sistema.
 - **Personal Dental:** Los dentistas y personal encargado podrán visualizar sus citas asignadas, marcar citas como completadas y gestionar su disponibilidad.
- **Asignación de Tareas y Gestión de Citas:**
 - Los administradores y el personal dental podrán asignar citas de manera automática o manual según la disponibilidad de los dentistas.
 - Los pacientes podrán seleccionar su dentista preferido (si aplica) y elegir la fecha y hora que mejor se ajuste a su agenda.
 - Las citas podrán ser modificadas o canceladas bajo ciertas condiciones, que se definirán en el flujo del negocio.
- **Notificaciones y Recordatorios:** El sistema enviará recordatorios automáticos a los pacientes sobre sus citas próximas, y también notificará a los administradores si alguna cita es modificada o cancelada.
- **Interfaz de Usuario (UI):** Una interfaz sencilla e intuitiva que permita a los pacientes navegar por el sistema de citas sin dificultades. Los dentistas y administradores deben tener una vista funcional que les permita gestionar sus tareas con facilidad.
- **Integración con Calendarios:** Los usuarios deben poder ver un calendario donde se muestre su disponibilidad y las citas programadas.
- **Historial Médico Básico:** El paciente debe poder acceder a su historial médico (de forma limitada), mostrando las citas anteriores y cualquier tratamiento recibido.

Límites de proyecto:

Contenedores Docker y Kubernetes

El alcance del proyecto está limitado al uso de **Docker** y **Kubernetes** como herramientas de contenedorización y orquestación. Estas herramientas serán configuradas como parte

de la infraestructura de desarrollo y producción, y no se permitirá el uso de otras herramientas sin el consenso del equipo de desarrollo.

- **Docker:**
 - Todos los servicios que conforman la aplicación (servidor de la API, base de datos, sistema de notificaciones, etc.) serán ejecutados en contenedores Docker.
 - Cada contenedor será configurado para garantizar que la aplicación se ejecute de manera aislada, sin interferir con otros servicios, utilizando imágenes específicas para cada servicio.
- **Kubernetes:**
 - Kubernetes será la herramienta encargada de orquestar los contenedores Docker, gestionando el escalado de los servicios, la distribución de carga, el equilibrio de tráfico y los despliegues continuos.
 - Se establecerá un clúster de Kubernetes en el entorno de producción que permita el despliegue automático de nuevas versiones de la aplicación sin tiempo de inactividad.
- **Infraestructura Fija:** El uso de Docker y Kubernetes no está sujeto a cambios. No se agregarán otras herramientas de contenedorización u orquestación sin la aprobación unánime del equipo de desarrollo. Además, la infraestructura del proyecto se basará en estos servicios, por lo que no se implementarán soluciones alternativas sin una evaluación conjunta.

Herramientas Fijas

El alcance también incluye el uso de un conjunto fijo de herramientas para el desarrollo, pruebas y despliegue:

- **GitHub:** Para el control de versiones y la colaboración en el código.
- **Trello:** Para la gestión de tareas, planificación de sprints y documentación.
- **Docker y Kubernetes:** Para contenedorización y despliegue. El uso de otras herramientas de orquestación no está permitido sin consenso grupal.
- **Frameworks y Librerías Específicas:** El proyecto utilizará frameworks y bibliotecas previamente seleccionados, como **React** (para el frontend), **Node.js** y **Express** (para el backend), y **MongoDB** o **PostgreSQL** (para la base de datos). No se realizarán cambios en las tecnologías de desarrollo sin una evaluación grupal.

Cambios en el Equipo o Herramientas

No se permitirá realizar cambios en el equipo o en las herramientas utilizadas sin el consenso explícito del grupo. Esto incluye:

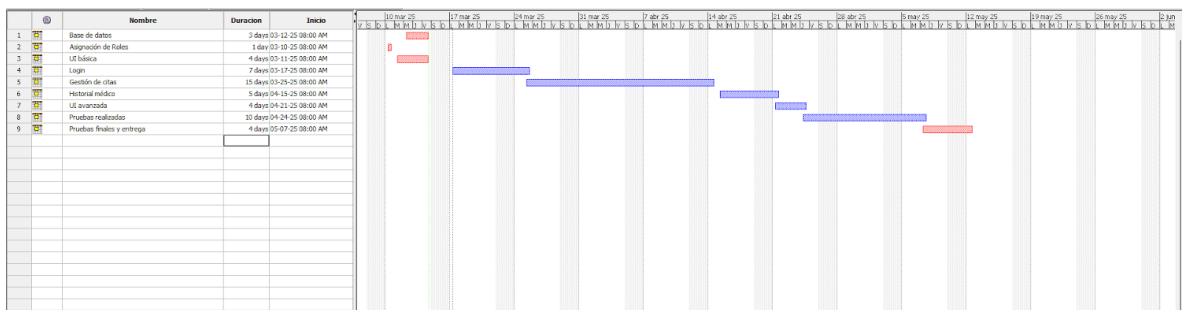
- **Cambios de Personal:** Si algún miembro del equipo debe ser reemplazado o reasignado, se discutirá en grupo para garantizar que el cambio no afecte el progreso ni el alcance del proyecto.
- **Cambios Tecnológicos:** Cualquier sugerencia de cambio en las herramientas, tecnologías o frameworks debe ser analizada y aceptada por todo el equipo. En este

proyecto, el uso de Docker, Kubernetes, GitHub, Trello y Notion es obligatorio, y no se podrán sustituir sin una evaluación y acuerdo grupal.

3. Control de Modificaciones en el Alcance

Para evitar desviaciones en el alcance, se implementará un proceso de **gestión de cambios** donde cualquier solicitud de modificación al alcance del proyecto deberá ser aprobada por el equipo. Las solicitudes de cambios se discutirán en las reuniones de Sprint y se decidirá si se incorpora al backlog del proyecto, siempre y cuando no impacte negativamente en los plazos o la calidad.

4.3.Gestión del Cronograma.



4.4.Gestión de Costos.

Concepto	Costo
Sistema Operativo (Windows) (Ej. 3)	\$342/computadora/año
Clúster Kubernetes (Amazon EKS)	\$864/año
Balanceadores de Carga (AWS Elastic Load Balancer)	\$240–\$360/año
Servicio CI/CD (GitLab Premium)	\$228/usuario/año
Almacenamiento Persistente (Google Cloud) (500GB en SSD)	\$1,020/año
Seguridad en la nube (AWS)	\$660–\$4,000/año (dependiendo de servicios y tráfico)

Soporte Técnico y Mantenimiento	\$1,200/año
Marketing	\$1,000/año
Capacitación del Personal:	\$2,000/año
Total Estimado	\$8,238-\$11,698 / año

4.5. Gestión de la Calidad.

La gestión de la calidad en el desarrollo de nuestro proyecto se enfoca en garantizar que el código y su funcionalidad de la aplicación cumpla con los estándares de desarrollo. Para lograrlo se implementan diversas estrategias de revisión de código, pruebas y buenas prácticas

Revisión de Código:

Para garantizar la calidad del código, se establecerá un proceso de revisión continua a través de GitHub, se aplicaran los siguientes controles

- **Uso de Pull Requests:** Cada cambio en el código será enviado mediante una PR y revisado por al menos un miembro del equipo antes de su aprobación.
- **Code Reviews:** Se realizarán revisiones de código para identificar errores, optimizar el rendimiento y mejorar la legibilidad.
- **Historial y Versionado:** GitHub permitirá mantener un control de versiones efectivo, permitiendo revertir cambios si es necesario.

Estándares de Codificación:

Para garantizar un código limpio y estructurado, se seguirán estándares de programación

- **Documentación en el Código:** Se agregarán comentarios adecuados y documentación en los archivos para facilitar la comprensión del código.
- **Linters y Herramientas:** Herramientas como PHP CodeSniffer y Prettier, para detectar errores y poder corregirlos de forma eficiente y asegurar que siga el formato del código.

4.6. *Gestión de los Recursos Humanos.*

La gestión de los recursos humanos en el proyecto es fundamental para garantizar una distribución eficiente de las responsabilidades y una comunicación fluida entre los integrantes del equipo

Asignación de Roles:

- Líder del proyecto:
David Luciano Raymundo Eguizaba, Coordina las actividades del equipo, supervisa las entregas y gestiona la comunicación al docente.
- Desarrollador Backend:

Jose Ricardo Aguilar Valdez, Felipe Alberto Salguero Quezada, Responsables de la lógica del servidor y la implementación de la base de datos en PHP.
- Desarrollador Frontend:

Emanuel Edgardo Luna Villanueva, Carlos Fabricio Cornejo Miranda, Encargados de la interfaz de usuario y la experiencia del usuario en la aplicación
- DevOps:

Rodrigo Ignacio Martinez Perez, Maneja la configuración y despliegue en Docker y Kubernetes, asegurando la escalabilidad y estabilidad del sistema.

Colaboración y Comunicación

Para mejorar la eficiencia del trabajo en equipo, se establecerán los siguientes mecanismos de comunicación y colaboración:

- **WhatsApp (Grupo del equipo):** Se utilizará para comunicación rápida entre todos los miembros del equipo y resolver dudas de manera inmediata.
- **Discord (Servidor del equipo):** Será el canal principal para reuniones semanales, verificación de avances y sesiones de discusión en caso de dudas más complejas.
- **Aula Digital:** Medio oficial para entregar avances y contacto con el docente.
- **GitHub:** Manera en la que nos compartiremos el código.

4.7.Gestión de la Comunicación.

Cada semana el coordinador del proyecto se encargará de actualizar el estado del proyecto en una reunión semanal a través de Discord, se usará pantalla compartida para explicar el progreso y mostrar los avances. se crearán los siguiente canales:

- Un canal **#informe-semanal** donde se publicarán actualizaciones.
- Un canal de voz, como **#reunión-semanal**, para las sesiones semanales.
- Un canal **#preguntas-generales** para cualquier tipo de duda que se tenga durante el desarrollo.

Los miembros del equipo pueden actualizar su progreso de manera diaria, indicando lo que han hecho durante el día y los problemas que han resuelto a través de WhatsApp.

Además, a través de Discord, el coordinador configurará recordatorios automáticos usando bots o notificaciones para asegurar que todos los miembros del equipo actualicen su progreso y revisen los informes, además de las notificaciones que se tienen mediante el tablero creado para el proyecto en Trello donde se tiene un mejor detalle del curso que tiene el proyecto y qué miembro del grupo está realizando cada tarea en específico.

4.8.Gestión de la Adquisición y Cierre.

INTEGRANTES Y ROLES:

David Luciano Raymundo Eguizábal(Líder proyecto).

José Ricardo Aguilar Valdez.

Felipe Alberto Salguero Quezada.

Rodrigo Ignacio Martínez Pérez.

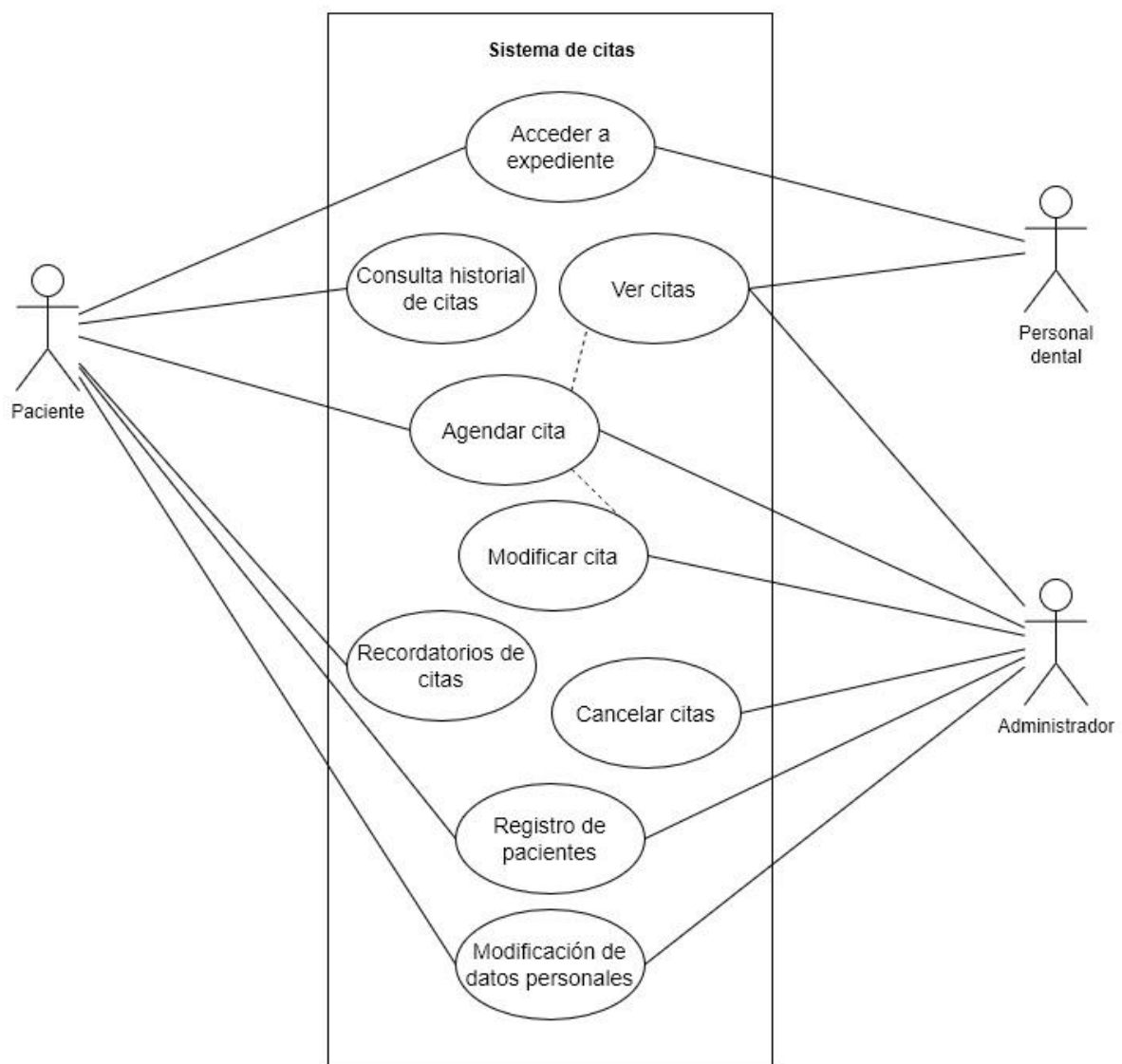
Emanuel Edgardo Luna Villanueva.

Carlos Fabricio Cornejo Miranda.

Licencia:

Proyecto Clínica Dental © 2025 by: David Luciano Raymundo Eguizábal(Lider proyecto). José Ricardo Aguilar Valdez. Felipe Alberto Salguero Quezada. Rodrigo Ignacio Martínez Pérez. Emanuel Edgardo Luna Villanueva. Carlos Fabricio Cornejo Miranda. is licensed under Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International <https://creativecommons.org/licenses/by-nc-sa/4.0/?ref=chooser-v1>

5.1.Diseño de la Aplicación.



Paleta de colores



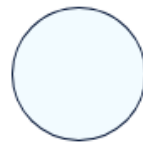
Principal
#21A2DC



Secundario
#F0D3F7



Terciario
#001242



Blanco
#E3EFF5



Oscuro
#040F16

Paleta de colores en fondo oscuro



Principal
#21A2DC



Secundario
#F0D3F7



Terciario
#001242



Blanco
#E3EFF5



Oscuro
#040F16

Degradado



#21A2DC

#137EA5

#115472

#092E3F



#F0D3F7

#18A576

#0B7250

#093F2D

Degradado fondos oscuros



#21A2DC

#137EA5

#115472

#092E3F



#F0D3F7

#18A576

#0B7250

#093F2D

Logotipos



Version clara



Version oscura

Wireframes principales



Clinica Dental Sonrisas

Inicio de sesión /registro

Informacion

Ubicacion

Enlaces de contacto / footer



Clinica Dental Sonrisas

**Formulario inicio de
sesion/ registro
paciente**

Enlaces de contacto / footer



**Formulario vista/
agendar cita**

Enlaces de contacto / footer

5.2.Arquitectura y Contenedorización.

- Docker: Configuración del Backend en Contenedores.

```
DockerFile X
DockerFile
1  # Imagen base con PHP y Apache
2  FROM php:8.1-apache
3
4  # Instalar extensiones necesarias para SQL Server
5  RUN apt-get update && apt-get install -y \
6      unixodbc \
7      unixodbc-dev \
8      libgssapi-krb5-2 \
9      && docker-php-ext-install pdo pdo_sqlsrv
10
11 # Copiar código fuente de la aplicación
12 COPY . /var/www/html
13
14 # Exponer puerto 80 para acceso HTTP
15 EXPOSE 80
16
17 # Iniciar Apache
18 CMD ["apache2-foreground"]
```

- Kubernetes: Archivos YAML para la Configuración(Backend).

```
! Despliegue.yaml X
! Despliegue.yaml
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: php-backend
5  spec:
6    replicas: 2
7    selector:
8      matchLabels:
9        app: php-backend
10   template:
11     metadata:
12       labels:
13         app: php-backend
14     spec:
15       containers:
16         - name: php-backend
17           image: mi-registro/php-backend:latest
18           ports:
19             - containerPort: 80
20           env:
21             - name: DB_HOST
22               value: "sql-server-service"
23             - name: DB_USER
24               value: "sa"
25             - name: DB_PASSWORD
26               valueFrom:
27                 secretKeyRef:
28                   name: sql-secret
29                   key: password
```

```
! Servicio.yaml X
! Servicio.yaml
1  apiVersion: v1
2  kind: Service
3  metadata:
4    name: php-backend-service
5  spec:
6    selector:
7      app: php-backend
8    ports:
9      - protocol: TCP
10        port: 80
11        targetPort: 80
12    type: ClusterIP
```

! hpa.yaml X

! hpa.yaml

```
1  apiVersion: autoscaling/v2
2  kind: HorizontalPodAutoscaler
3  metadata:
4    name: php-backend-hpa
5  spec:
6    scaleTargetRef:
7      apiVersion: apps/v1
8      kind: Deployment
9      name: php-backend
10   minReplicas: 2
11   maxReplicas: 5
12   metrics:
13     - type: Resource
14       resource:
15         name: cpu
16         target:
17           type: Utilization
18           averageUtilization: 50
19
```

- Archivos YAML para configurar SQL server.

```
! DespliegueSQL.yaml X
! DespliegueSQL.yaml
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: sql-server
5  spec:
6    replicas: 1
7    selector:
8      matchLabels:
9        app: sql-server
10   template:
11     metadata:
12       labels:
13         app: sql-server
14     spec:
15       containers:
16         - name: sql-server
17           image: mcr.microsoft.com/mssql/server:2019-latest
18           ports:
19             - containerPort: 1433
20           env:
21             - name: ACCEPT_EULA
22               value: "Y"
23             - name: SA_PASSWORD
24               valueFrom:
25                 secretKeyRef:
26                   name: sql-secret
27                   key: password

! ServicioSQL.yaml X
! ServicioSQL.yaml
1  apiVersion: v1
2  kind: Service
3  metadata:
4    name: sql-server-service
5  spec:
6    selector:
7      app: sql-server
8    ports:
9      - protocol: TCP
10        port: 1433
11        targetPort: 1433
12    type: ClusterIP
```

```
! SecreSQL.yaml X
! SecreSQL.yaml
1  apiVersion: v1
2  kind: Secret
3  metadata:
4    name: sql-secret
5    type: Opaque
6  data:
7    password: c2VndXJvMTIz # "seguro123" en Base64
```

- **API: Comunicación entre Componentes**

La API facilita la comunicación entre el backend y otros componentes del sistema, asegurando una interacción fluida mediante el protocolo HTTP/HTTPS. Los endpoints principales permiten gestionar el acceso a la información de pacientes, citas y tratamientos, garantizando un flujo de datos eficiente y seguro.

- **Protocolo:** HTTP/HTTPS
- **Endpoints:**
 - *POST /login*: Permite a los usuarios autenticarse en la plataforma y obtener un token de sesión.
 - *GET /pacientes*: Recupera la lista de pacientes registrados en la base de datos.
 - *POST /citas*: Permite agendar una nueva cita asociada a un paciente.
 - *GET /tratamientos*: Proporciona un listado de los tratamientos disponibles en la clínica.

- **Autenticación y Seguridad**

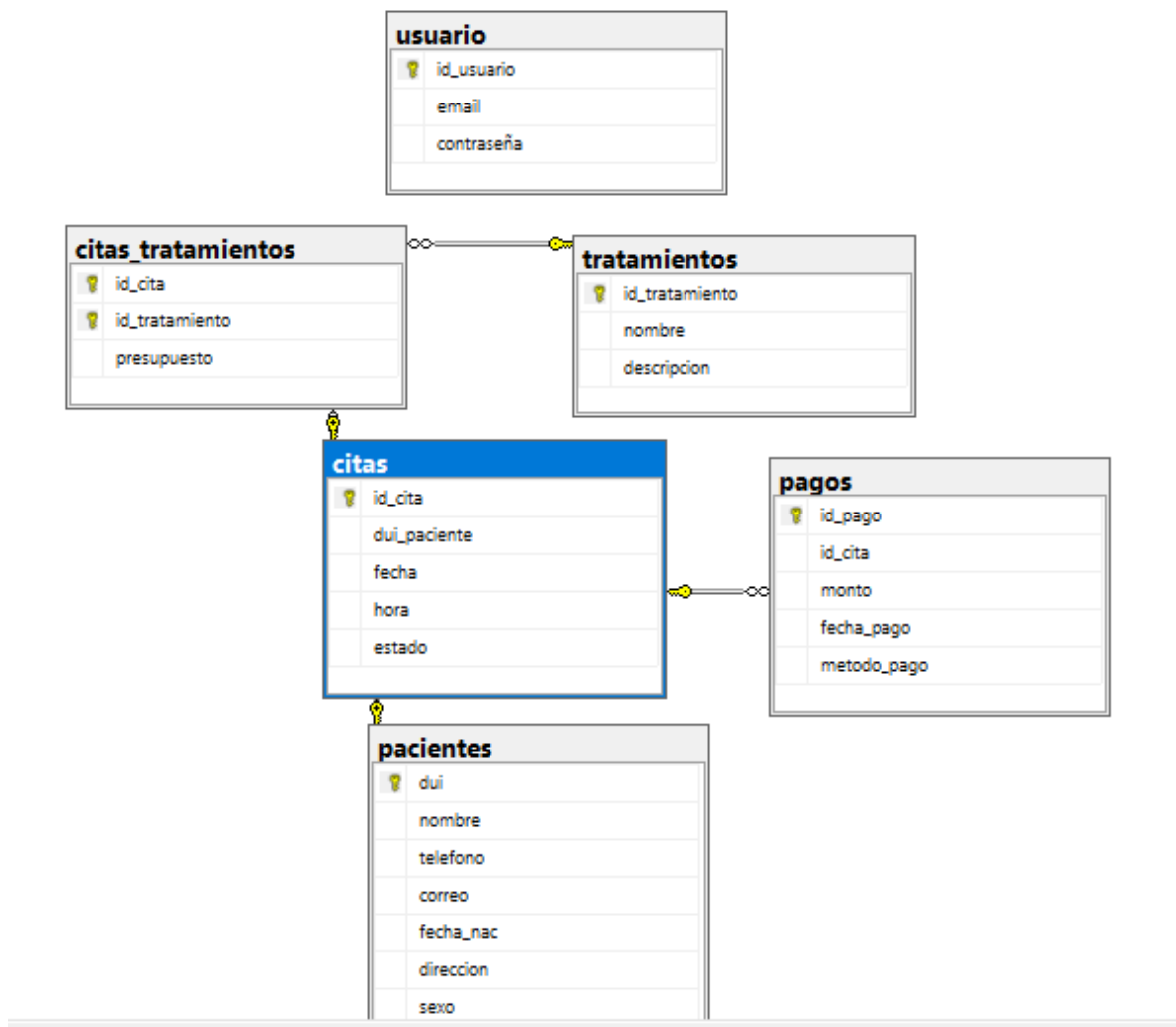
Para garantizar la seguridad de la aplicación, se implementan varias medidas de autenticación y protección contra ataques informáticos.

- **JWT para autenticación de usuarios:** Se utiliza JSON Web Token (JWT) para validar las sesiones de los usuarios y restringir el acceso a recursos protegidos.
- **Protección contra ataques:**
 - **Filtrado de inputs:** Se validan y sanitizan las entradas de usuario para evitar ataques de SQL Injection.
 - **Headers de seguridad:** Se configuran cabeceras HTTP para prevenir XSS (Cross-Site Scripting) y CSRF (Cross-Site Request Forgery).

- **HTTPS obligatorio en producción:** Todas las comunicaciones en el entorno productivo se realizan a través de **HTTPS**, asegurando la integridad y confidencialidad de los datos transmitidos.

5.3. Gestión de la Base de Datos.

Diagrama ER (nota la tabla usuarios no se relaciona con las demas tablas ya que sera usada para almacenar el usuario que usara el programa):



Diccionario de datos:

Tabla	Nombre del Campo	Tipo de Datos	Descripción	Tamaño	Valores Permitidos	Restricciones	Relaciones
pacientes	dui	CHAR(10)	Identificación única del paciente	10	-	PK, NOT NULL	-
pacientes	nombre	VARCHAR(100)	Nombre completo del paciente	100	-	NOT NULL	-
pacientes	telefono	VARCHAR(9)	Número de teléfono del paciente	9	-	NOT NULL	-
pacientes	correo	VARCHAR(100)	Correo electrónico único	100	-	UNIQUE	-
pacientes	fecha_nac	DATE	Fecha de nacimiento	-	-	NOT NULL	-
pacientes	direccion	TEXT	Dirección completa	-	-	NOT NULL	-
pacientes	sexo	CHAR(1)	Género del paciente	1	M, F	CHECK (sexo IN ('M', 'F'))	-
citas	id_cita	INT	Identificación única de la cita	-	-	PK, IDENTITY(1,1)	-
citas	dui_paciente	CHAR(10)	Paciente asociado a la cita	10	-	FK -> pacientes(dui)	FK -> pacientes(dui)
citas	fecha	DATE	Fecha de la cita	-	-	NOT NULL	-
citas	hora	TIME	Hora de la cita	-	-	NOT NULL	-
citas	estado	VARCHAR(20)	Estado de la cita	20	Pendiente, Confirmada, Cancelada	CHECK (estado IN ('Pendiente', 'Confirmada', 'Cancelada'))	-
tratamientos	id_tratamiento	INT	Identificación única del tratamiento	-	-	PK, IDENTITY(1,1)	-
tratamientos	nombre	VARCHAR(100)	Nombre del tratamiento	100	-	NOT NULL	-
tratamientos	descripcion	TEXT	Descripción del tratamiento	-	-	NOT NULL	-
citas_tratamientos	id_cita	INT	Cita asociada	-	-	PK, NOT NULL	FK -> citas(id_cita)
citas_tratamientos	id_tratamiento	INT	Tratamiento asociado	-	-	PK, NOT NULL	FK -> tratamientos(id_tratamiento)
citas_tratamientos	presupuesto	DECIMAL(10,2)	Costo estimado del tratamiento	-	-	NOT NULL	-
pagos	id_pago	INT	Identificación única del pago	-	-	PK, IDENTITY(1,1)	-
pagos	id_cita	INT	Cita asociada al pago	-	-	FK -> citas(id_cita)	FK -> citas(id_cita)
pagos	monto	DECIMAL(10,2)	Monto pagado	-	-	NOT NULL	-
pagos	fecha_pago	DATE	Fecha del pago	-	-	NOT NULL	-
pagos	metodo_pago	VARCHAR(20)	Método de pago	20	Efectivo, Transferencia	CHECK (metodo_pago IN ('Efectivo', 'Transferencia'))	-
usuario	id_usuario	INT	Id del usuario	-	-	PK, IDENTITY(1,1)	-
usuario	email	VARCHAR(100)	Email para el usuario	100	-	UNIQUE, NOT NULL	-
usuario	contraseña	VARCHAR(255)	Contraseña para el usuario	255	-	NOT NULL	-

6.Desarrollo del Proyecto.

Requerimientos del usuario

- ¿Cómo podríamos simplificar el proceso para agendar una cita?
- ¿Cómo podríamos minimizar los tiempos perdidos por citas canceladas?
- ¿Cómo podríamos recordarles a los pacientes de sus citas para confirmarlas?
- ¿Cómo podríamos agendar citas de seguimiento para los pacientes?
- ¿Cómo podríamos mantener un registro y expediente de cada paciente y el procedimiento realizado?
- ¿Cómo podríamos sugerir la próxima cita a agendar?

Implementación del Backend

1. Implementación de contenedor por medio de Docker con imagen de PHP 8.1 y servidor web Apache.
2. Desarrollo de conexión a base de datos en SQL Server.
3. Desarrollo de componentes de registro de usuarios.
4. Desarrollo de formularios de modificación de datos.
5. Desarrollo de funcionalidades de calendario y horarios de citas.
6. Desarrollo de vistas de horarios de citas disponibles.
7. Desarrollo de creación y sugerencia de horarios para citas
8. Desarrollo de vistas de citas agendadas.
9. Desarrollo de comprobación de citas no duplicadas o con datos erróneos.
10. Confirmaciones de citas y recordatorios.
11. Recordatorios automáticos de citas a los pacientes.
12. Vista de expedientes e historial de citas.
13. Configuración de balanceo con Kubernetes.
14. Desarrollo de funciones de monitoreo de rendimiento y registro de eventos o errores.

7.Pruebas y Control de Calidad.

Pruebas por desarrollar

- Creación de usuario.
 - Paciente nuevo

- Paciente registrado por administrador.
- Recuperación de contraseñas
 - Recuperación de contraseñas de todos los roles.
- Modificación de datos
 - Datos personales del paciente, modificados por personal dental, administradores o el mismo paciente.
- Creación de cita
 - Creación de cita por paciente, por personal dental y por administrador.
- Cancelación de cita
 - Cancelación por cualquier rol.
- Modificación de cita
 - Modificación por administrador
- Envío de recordatorio
 - Recordatorio automático al paciente.
- Visualización de citas
 - Citas propias del paciente
 - Vista de citas diarias para personal dental.
- Visualización de expediente.
 - Expediente personal por paciente.
 - Expediente visualizado por personal dental.
- Prueba de optimización de horarios de cita.
 - Sugerencia de espacio a agendar.
- Pruebas de carga para registro de citas
 - Escalabilidad de registro.
- Pruebas de carga para registro de usuarios
 - Escalabilidad de registro.
- Pruebas de carga para visualización de citas.
 - Escalabilidad de visualización de citas.

8.Resultados y Conclusiones.

Resultados en términos de funcionalidad:

Hasta el momento, el proyecto se encuentra en la fase de planificación y diseño. Se ha establecido una base sólida de lo que se espera lograr con la aplicación web El análisis de los requerimientos y la estructuración de las funcionalidades principales nos ha permitido identificar las diferentes funciones clave que se implementarán en el sistema:

- **Autenticación de Usuarios**
- **Gestión de Citas en Tiempo Real**
- **Historial Médico y Citas**

Aunque el sistema aún está en su fase inicial, la planificación detallada de estas funcionalidades asegura que, cuando se inicie el desarrollo, la aplicación será capaz de resolver eficientemente los problemas actuales de la clínica, como la falta de organización y la alta tasa de ausencias. La funcionalidad de gestión de citas, junto con la autenticación de usuarios y el historial médico, contribuirá a una experiencia de usuario mucho más fluida y mejorará la eficiencia tanto para los pacientes como para el personal administrativo.

Resultados en términos de desempeño:

Se han definido elementos clave para asegurar un buen desempeño del sistema. Esto incluye la planificación del entorno técnico, Diseño básico de la aplicación, las herramientas a utilizar y la infraestructura que se emplea para garantizar el rendimiento de la aplicación:

- **Escalabilidad y Desempeño:** Se ha planificado el uso de Docker y Kubernetes para la contenedorización y la orquestación del sistema. Esto permitirá que la aplicación se despliegue de manera eficiente y escale según la demanda.
- **Disponibilidad del Sistema:** La integración de Kubernetes garantizará que el sistema esté disponible en todo momento, incluso si hay algún fallo en los contenedores o en los servicios.

La planificación detallada de la infraestructura con Docker y Kubernetes asegura que la aplicación pueda manejar múltiples usuarios y una carga alta sin comprometer el desempeño. La escalabilidad de la solución y la optimización del tiempo de respuesta son aspectos clave que contribuirán a la eficiencia operativa de la clínica.

Conclusiones

El proyecto Sistema de Citas Dentales en línea para la Clínica Dental Sonrisas será una excelente oportunidad de aplicar prácticas de desarrollo ágiles y tecnologías de vanguardia en un contexto real de negocio. El uso de herramientas como GitHub, Trello, Docker y Kubernetes; es fundamental para mantener el flujo de trabajo ordenado y asegurar la correcta gestión de versiones e infraestructura de la aplicación.

Desde el inicio del proyecto nos enfrentamos a varios desafíos relacionados con la integración de sistemas, especialmente con la sincronización de un calendario de citas. Sin embargo, la implementación de un sistema de notificaciones y recordatorios automáticos ayudará significativamente a minimizar estos problemas.

A través de la retroalimentación del equipo de la clínica, se ha aprendido que la interfaz debe ser simple y accesible resultando ser más efectivo que añadir demasiadas funcionalidades.

Mejoras Identificadas:

Una de las principales es la necesidad de incorporar otras funcionalidades como la integración de otros sistemas médicos que podrían utilizar en el futuro ya se historial clínico electrónico más completo para visualizado por los pacientes, sistema de notificaciones para el equipo de la clínica y los pacientes y un servicio de pago en línea seguro. Además, el sistema de autenticación podría contar con más opciones como la autenticación biométrica, para mejorar la seguridad y la facilidad de acceso hablando de la parte de los usuarios.

9. Anexos y Documentación de Apoyo.

Diseño de vistas de la app (pagina a figma.com).

<https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwjTz8G7npGMAxXiTDABHXP8MvYQFnoECAoQAQ&url=https%3A%2F%2Fwww.figma.com%2F&usg=AOvVaw2ixWJnr7CgjXMg3QXfTz4u&opi=89978449>

Link para la licencia creative commons.

https://chooser-beta.creativecommons.org/?lang=es_CO

Ejemplo de despliegue sobre kubernetes.

<https://adictosaltrabajo.com/2022/09/27/despliegue-de-aplicaciones-sobre-kubernetes/>

Ejemplo de contenedores de desarrollo.

<https://code-visualstudio-com.translate.goog/docs/devcontainers/tutorial? x tr sl=en& x tr tl=es& x tr hl=es& x tr pt\>