MOPC

#ObrasParaLaGente

Sistema Informático de Participación Ciudadana (SIPC)

MANUAL DEL SISTEMA

EXCELSIS

Índice

1. Introducción 3
2. Repositorio 3
3. Recursos asociados a este documento 3
4. Pautas para la codificación 4
4.1 Organización del código fuente 4
5. Entorno de Desarrollo 11
5.1 Creación de la Base de Datos en el servidor PostgreSQL v10.4 11
6. Entorno de Producción 12
6.1 Requerimientos de hardware y software para el servidor de producción 12
6.2 Servidores 13
6.3 Instalación de CentOS 7 en los servidores de producción 14
7.1 Configuración e instalación del software requerido en los servidores de producción 26

1. Introducción

El presente documento tiene por objetivo brindar al equipo de TIC's de MOPC toda la información necesaria acerca de cómo fue desarrollado el sistema **SIPC** y los aspectos fundamentales para su puesta en producción, así como también una guía de configuración del entorno de desarrollo para continuar con el mantenimiento posterior del mismo.

2. Repositorio

El repositorio de código y documentación se encuentra disponible en la dirección web ***http://gitlab.excelsis.com.py/mopc***. A continuación, se listan los proyectos que están en el repositorio.

Ruta	Descripción
RUTA_BASE_REPO*/sipc-core-backend.git*	Implementación base para los Backends
RUTA_BASE_REPO*/sipc-dominio.git*	Dominios de las aplicaciones
RUTA_BASE_REPO /*sipc-comun-backend.git*	Común para los Backends
RUTA_BASE_REPO*/sipc-controlacceso-backend.git*	Control de ACCESO
RUTA_BASE_REPO*/sipc-presentacion-web.git*	Frontend SIPC
RUTA_BASE_REPO*/sipc-simffaso-backend.git*	Backend Simffaso
RUTA_BASE_REPO*/sipc-sigirci-backend.git*	Backend Sigirci
RUTA_BASE_REPO*/sipc-estaticos-web.git*	Elementos estáticos de SIPC
RUTA_BASE_REPO*/documentacion.git*	Documentación del Proyecto.

Tabla 1 *Dirección de los Repositorios*

RUTA_BASE_REPO no es más que la ruta hasta la carpeta raíz donde se descargó el repositorio en la máquina local. En las carpetas principales se encuentra un fichero nombrado **LEAME.txt**, en el cual se describe el contenido de las mismas.

3. Recursos asociados a este documento

En conjunto con este documento se proporcionará una serie de recursos útiles para el desarrollo y puesta en producción del sistema. Todos estos recursos (en lo adelante RT) se encuentran en una carpeta dentro del repositorio creado para el proyecto por el equipo de TIC's de MOPC. Dicha carpeta se encuentra en las siguientes rutas:

RUTA_BASE_REPO\documentacion\productos\mopc\simffaso01\03.implementacion\manual_sistema

4. Pautas para la codificación

En esta sección se detallará cómo se organizó el código fuente de acuerdo a las capas de la arquitectura propuesta. Además, se explicará el objetivo que cumplen las clases principales que conforman el núcleo del sistema y finalmente se expondrá un ejemplo de cómo desarrollar un proceso de alta, baja, modificación y consulta de registros almacenados en la base de datos.

4.1 Organización del código fuente

Conforme a lo establecido en el apartado del documento de arquitectura y al comportamiento de los diferentes componentes que conforman la aplicación, estos se agruparon por los tres niveles de la arquitectura que se describen a continuación.

Nivel Presentación Web

Para la capa de presentación se utilizó el framework JSF, el cual es un marco de trabajo del lado del servidor para crear aplicaciones web Java JEE. Entre sus principales características se encuentran:

- Forma parte del estándar JEE.
- Utiliza un sencillo fichero de configuración para la aplicación (faces-config.xml).
- Utiliza páginas con extensión xhtml para generar las vistas, añadiendo una biblioteca de etiquetas propia para crear los elementos de los formularios.
- Asocia a cada vista con un conjunto de objetos Java manejados por el controlador (managed beans) que facilitan la recogida, manipulación y visualización de los valores mostrados en los diferentes elementos de los formularios.
- Introduce una serie de etapas en el procesamiento de la petición, como por ejemplo la de validación, reconstrucción de la vista, recuperación de los valores de los elementos, etc.
- Es extensible, pudiendo crearse nuevos elementos de la interfaz o modificar los ya existentes.

A las clases Java que se asocian a las vistas JSF se les denomina backend beans ya que es el código Java que está detras del formulario. Estos beans se declaran usando anotaciones que provee JSF, ya que el framework se encarga de su construcción y destrucción automáticas cuando es necesario.

Para facilitar la visualización y recogida de los datos de los formularios JSF introduce un lenguaje de expresiones EL, el cual permite acceder de manera muy sencilla a las propiedades de los backbeans. Un ejemplo de una expresión EL simple se puede escribir como:

#{backbean.propiedad}

Esto permite asignar o leer valores para el intercambio de datos entre las clases Java y las páginas HTML.

Por lo general, cuando se trabaja con JSF, se utiliza una librería de componentes desarrollada para este framework. En el caso de SIPC se utilizó Primefaces. Esta es una librería de componentes visuales open source y como principales características tiene las siguientes:

- Soporte nativo de Ajax.
- Es compatible con otras librerías de componentes, como JBoss Richfaces.
- Es un proyecto open source, activo y bastante estable entre versiones.
- Proporciona una amplia gama de componentes que cuentan con documentación y ejemplos de código (http://www.primefaces.org/showcase).

Núcleo del Nivel: Los archivos correspondientes al núcleo son los encargados de contener la lógica común para todos los módulos del nivel. Estos archivos están compuestos por las clases Bases y Útiles de la aplicación. Las clases principales que conforman dicho núcleo son:

- ✓ ***ApiResponse***: Clase genérica que contiene los principales métodos comunes para el acceso a los servicios web en la aplicación.
- ✓ ***BaseFC:*** Clase genérica de la cual deben heredar todos los controladores de JSF utilizados en el sistema.
- ✓ ***cusConListadoGenericoLazyFC***: Clase genérica de la cual heredan aquellas clases que su lógica de negocio implica listar información asociada a un dominio.
- ✓ ***JsfUtil:*** Clase utilitaria para mejorar el manejo de funciones de JSF.

Presentación: Lo referente a la presentación de la información está representado por los ficheros con extensión **.xhtml**, los cuales contienen la implementación y distribución en pantalla de los componentes que conforman la interfaz de usuario.

Controladores de Vistas: Estos archivos lo constituyen los ***Controladores de JSF*** (Backend beans) que son simples clases Java que se asocian a los formularios.

Fachadas: se encargan del control de los servicios web y son accedidas desde los controladores de JSF.

La distribución física del código quedó de la siguiente manera:

Figura 1 *Estructura del código fuente del frontend*

Donde el paquete **src/main/java** contiene los archivos de código fuente .java además de algunas configuraciones necesarias para el proyecto. A continuación, se describe su contenido.

- **business:** Contiene las fachadas de cada uno de los módulos.
- **components:** Contiene clases personalizadas que sobrescriben el comportamiento por defecto de algunos componentes de Primefaces.
- **core:** Contiene clases del núcleo del sistema que son reutilizables en el resto de la aplicación y que conforman la arquitectura base.
- **genericos:** Contiene clases que son reutilizables, pero que no pertenecen al núcleo (core), sino que están enfocadas al negocio particular del sistema.
- **modulos:** Contiene clases de todos los módulos desarrollados en el sistema.
- **org.primefaces.paradise:** Contiene clases propias de la plantilla de primefaces utilizada para el sistema (**paradise**).
- **security:** Contiene clases utilizadas para el manejo de la seguridad del sistema. Entre las que se puede destacar **SecurityFilter**, que no es más que un filtro de seguridad para la autorización a los recursos del sistema. Todos los recursos que se encuentren dentro de la carpeta **securized** (que se ubica dentro de la carpeta **web**) requerirán como mínimo que el usuario que trate de acceder a ellos esté autenticado en el sistema.

La carpeta **src/main/webapp** contiene todos los recursos que serán procesados por el servidor web (páginas HTML, imágenes, css, etc.), además de algunas configuraciones necesarias para poner en producción el sistema de una manera adecuada. A continuación, se describe su contenido.

- **pages:** Contiene páginas genéricas que son utilizadas para mostrar mensajes en el sistema, como por ejemplo un error de acceso denegado a un determinado recurso o funcionalidad.
- **public:** Contiene los recursos que se acceden sin necesidad de autenticación.
- **resources:** Contiene los recursos que son requeridos por la plantilla utilizada como imágenes, script y css.
- **securized:** Contiene los recursos que son protegidos mediante control de acceso a través del mecanismo de autenticación y autorización.
- **WEB-INF:** Contiene todos los recursos relacionados con la aplicación web que no se han colocado en el directorio raíz y que no deben servirse al cliente. Esto es importante, ya que este directorio no forma parte del documento público, por lo que ninguno de los ficheros que contenga podrá ser enviado directamente a través del servidor web. En este directorio se coloca el archivo **web.xml**, donde se establece la configuración de la aplicación web para su despliegue en el servidor.
- Los *recursos estáticos* de la aplicación van a estar alojados en un servidor **Nginx** y de esta manera no se sobrecarga el servidor de aplicaciones, logrando así un balanceo de carga en la aplicación.

Nivel Lógica de Negocio

Para la capa de lógica de negocio se utilizó el Spring Framework. Esta es una plataforma que nos proporciona una infraestructura que actúa de soporte para desarrollar aplicaciones Java.

Spring Framework es un contenedor ligero ("lightweight container") en contraposición a un servidor de aplicaciones JEE. En el caso de una aplicación web, basta con un contenedor de Servlets como Tomcat o Jetty.

Los componentes en Spring se declaran y se configuran con anotaciones. Luego al iniciar Spring, este crea dichos componentes y sus relaciones en el Application Context. Mediante la anotación @Autowired Spring busca la clase correspondiente, la instancia y la inyecta donde la necesitemos.

Las principales anotaciones para declarar componentes de Spring son las siguientes:

@Component: Componente genérico.

@Repository: Repositorios de acceso a datos.

@Service: Servicios de negocio.

@Controller: Controlador Web MVC.

@RestController: Controlador servicios REST.

@Configuration: Configuración a nivel de clase Java.

@Value: Valor inyectado proveniente del archivo de configuración.

@Autowired: Inyección de componente.

@Qualifier: Inyección de componente con nombre personalizado.

Como herramienta indispensable para el trabajo con Spring hoy día, se encuentra Spring Boot, la cual tiene las siguientes carcateríaticas:

- **Convención sobre configuración:** Provee las configuraciones necesarias bajo diferentes escenarios.
- **Dependency management:** Provee un análisis de las dependencias que los proyectos alrededor de spring utilizan de manera que nosotros solo tenemos que indicar que dependencia necesitamos sin necesidad de indicar la versión.
- **Auto-configuración:** Spring Boot es lo suficientemente inteligente para poder activar las configuraciones necesarias.
- **Starters dependencies:** Agrupa conjuntos de dependencias de acuerdo al contexto que se necesite en la aplicación.
- **Actuator:** Permite visualizar métricas, el estado de nuestra aplicación y los servicios con los que se conecta.
- **Embedded Server:** Da soporte para Tomcat, Jetty y Undertow embebidos.

Conforme lo descrito en el documento de arquitectura, el proyecto base o core del backend está encargado de manejar el núcleo de la arquitectura y los drivers y componentes más importantes que dan soporte a la misma, los cuales se muestran y describen a continuación.

Figura 2 *Proyecto base del backend.*

configs: Contiene clases de configuraciones generales para poder utilizar los diferentes componentes de la arquitectura.

exceptions: Contiene los componentes utilizados para manejar y controlar las excepciones del sistema, así como la jerarquía entre ellos.

repository: Compontes creados por el equipo de arquitectura para abstraer y facilitar los procesos de consultas a fuentes de datos.

utils: En este paquete se encuentran una serie de utilidades que sirven de apoyo al proceso de desarrollo y sobre las cuales se han homogenizado diferentes procesos en vista a agilizar la implementación de varios procesos en el sistema como, por ejemplo: Trabajo con fechas, Filtros de Api, Generadores de claves Hash, Trabajo con JSON (JavaScript Object Notation), Manejo de Reportes, Manejo de plantillas de correo con Velocity, entre otras.

Por otra parte, el dominio engloba las entidades que se relacionan para dar cumplimiento a los procesos de la solución SIPC. Estas entidades serán compartidas por las aplicaciones que se derivan a partir de la separación lógica de los niveles de la aplicación. De esta manera se busca asegurar que todos los subsistemas o soluciones compartan la información con una estructura similar. La organización del dominio quedó estructurada de la siguiente manera a fin de ser entendible y de no repetir entidades:

Figura 3 *Código del dominio del sistema.*

- **controlacceso:** Contiene entidades de transferencia de datos que colaboran para manejar la autenticación y autorización y auditoria del sistema.
- **genericos:** Contiene entidades de transferencias de datos que son compartidas por toda la aplicación como por ejemplo GenReferenciasDTO.
- Las entidades que surjan como parte del proceso de las aplicaciones SIGIRCI y SIMFASSO serán colocadas en paquetes que hagan referencias a estos proyectos.

La solución sipc-comun-backend es la encargada de englobar los componentes comunes a las diferentes soluciones, por ejemplo, la gestión de la auditoria, así como la centralización de los puntos de entradas de los diferentes sistemas con los cuales las soluciones SIGIRCI y SIMFFASO tendrían que interactuar. La finalidad de este paquete es centralizar procesos que hoy comparten todas las soluciones del sistema y las cuales son propias del negocio de SIPC, por tal motivo no puede confundirse con el propósito del core. A continuación, se muestra una imagen de cómo quedó organizado su código.

Figura 4 *Código del proyecto común para el backend.*

El control de acceso es el subsistema que engloba todo el manejo de la seguridad asociada a SIGIRCI y SIMFFASO. La organización de este subsistema quedó estructurada de la siguiente manera:

Figura 5 *Código control de acceso*

- **config:** Configuraciones asociadas a la seguridad. Filtros para interceptar las peticiones. Configuraciones para el trabajo con AD (Directorio Activo) y SICOE. En este paquete se han sobrescrito varias configuraciones por defecto del framework Spring para adaptarlas a las necesidades de las soluciones propuestas.

- **controllers:** Se encarga de establecer el vínculo entre la capa de frontend y backend. Contiene los **"endpoints"** del control de acceso, transfiere la información hacia la capa de lógica de negocio y devuelve hacia los clientes las respuestas http correspondientes.
- **services**: Se encarga de manejar toda la lógica de negocio asociada a los procesos del control de acceso, las validaciones etc. Es el componente encargado de hacer cumplir las reglas del negocio, comunicarse con el acceso a datos y recibir de este los datos provenientes de las fuentes de datos, procesarlos y enviar los datos al controller o en caso de ser requerido lanzar las notificaciones.
- **repository**: Contiene las clases que manejan el acceso a datos. Es el componente encargado de persistir la información, así como leerla, de las distintas fuentes de datos, comunicarse con el service y brindar la misma. Solo en este componente se concentran todas las consultas de acceso a datos.

5. Entorno de Desarrollo

En esta sección se abordarán los elementos necesarios para configurar el entorno de desarrollo. Una vez configurado este entorno, el equipo de TIC's de MOPC estará en condiciones de continuar con el mantenimiento posterior del sistema.

5.1 Creación de la Base de Datos en el servidor PostgreSQL v10.4

Para el correcto funcionamiento del sistema se necesitan crear la base de datos (BD). En la carpeta **Backup** de los RT se encuentran varios backup de la base de datos, que contienen datos iniciales y de prueba útiles para el proceso de desarrollo del sistema. Además, en la carpeta **Documentación BD** de los RT, se encuentran las descripciones y comentarios específicos sobre el diseño de la base de datos, por lo que no se abordarán en este documento.

De acuerdo a los términos de referencia y las características del sistema, se construyó el mismo con 1 bases de datos:

- SIPC: Es la base de datos para registrar información centralizada de la aplicación.

Entre las tablas más importantes que son comunes a estas bases de datos, se encuentran las siguientes:

seg_audit: en esta tabla se almacenan los registros de auditoría sobre el resto de las tablas de la base de datos, con el objetivo de contar con el histórico de inserciones, actualizaciones y eliminaciones. La configuración de las tablas a auditar puede ser gestionada en el sistema por un administrador.

seg_audit_hist: en esta tabla se almacena el histórico de habilitación e inhabilitación de todas las tablas auditables.

seg_usua_token: en esta tabla se almacenan los datos de seguridad asociados a los token de acceso asignados a los usuarios durante los procesos de autenticación y autorización.

gen_referencias: es una de las tablas más importantes en la base de datos, cuyo principal objetivo es almacenar información asociada a las configuraciones del sistema y mantenimientos en general. Su mayor aporte consiste en que aporta flexibilidad para reducir el número de tablas necesarias en la base de datos y que esta no crezca considerablemente con tablas que solo se utilizan para crear nomencladores, mantenimientos, metadatos o clasificadores como comúnmente se conoce. Esta flexibilidad a la vez aporta dinamismo al sistema ya que en un futuro si se desea agregar un nuevo mantenimiento simple no es necesario modificar la estructura de la base de datos si no agregar nuevas entradas en esta tabla con un dominio (campo DOMINIO) específico.

6. Entorno de Producción

En esta sección se abordarán los requerimientos de hardware y software necesarios para la puesta en producción del sistema. De igual forma se explican los pasos a seguir para instalar y configurar el sistema operativo y software necesarios en los servidores de producción.

6.1 Requerimientos de hardware y software para el servidor de producción

A continuación se exponen las principales características de hardware y software que se proponen para el correcto funcionamiento del sistema en la etapa de producción, para lo cual se tuvo en cuenta fundamentalmente, el elevado volumen de documentos que puede ser almacenado y el acceso de múltiples usuarios desde cualquier localidad a través de Internet, tanto al Portal Público, como al frontend de la aplicación, de tal forma que se garantice un rendimiento estable y la integridad de la información registrada.

- **Servidores de aplicaciones**
  - ✓ **Sistema Operativo:** Como sistema operativo para el servidor de aplicaciones, se propone utilizar CentOS 7, ya que es de código abierto y está basado en la distribución Red Hat Enterprise Linux, operándose de manera similar, y cuyo objetivo es ofrecer al usuario un software de "clase empresarial" gratuito. Se define como robusto, estable y fácil de instalar y utilizar. Desde la versión 5, cada lanzamiento recibe soporte durante diez años, por lo que la versión 7 recibirá actualizaciones de seguridad hasta el 30 de junio de 2024.
  - ✓ **Memoria RAM:** Se recomienda contar como mínimo con 16 GB de RAM. Este valor se puede variar en dependencia de la cantidad de usuarios concurrentes y el comportamiento en cuanto a rendimiento durante el uso del sistema.
  - ✓ **Capacidad en Disco:** Debido a que el sistema maneja un gran volumen de documentos de para su correcto funcionamiento, el almacenamiento de los mismos es de suma importancia. Se propone como mínimo contar con 100GB de espacio.
- **Servidor de Base de Datos**
  - **Sistema Operativo:** Al igual que para los servidores de aplicaciones, se recomienda CentOS 7.
  - **Memoria RAM:** El acceso a la base de datos para la obtención de la información es una de las tareas más importantes y que más recursos consume en la aplicación, siendo así, y con el objetivo de lograr una buena respuesta a las peticiones de los usuarios finales se propone

contar como mínimo con 16 GB de RAM. En caso de que ambos servidores se encuentren físicamente en el mismo ordenador, con los 8 GB sugeridos para el servidor de aplicaciones debe ser suficiente para que inicialmente el sistema tenga un rendimiento estable. Al igual que en el caso anterior, dicho valor entrará en evaluación una vez se comience a utilizar el sistema y puede variar en algún momento en dependencia del comportamiento.

- **Capacidad en disco:** Teniendo en cuenta que los archivos digitalizados no se almacenan en el servidor de base de datos, se propone contar con 100 GB de espacio en el disco duro. En caso de que ambos servidores se encuentren físicamente en el mismo ordenador, con los 500 GB sugeridos para el servidor de aplicaciones debe ser suficiente para que inicialmente se pueda almacenar toda la información sin dificultad en la base de datos.

- **Servidor de recursos estáticos**
  - **Sistema Operativo:** Se recomienda usar también CentOS 7.
  - **Memoria RAM:** Se recomienda para este servidor 8 GB de RAM, ya que el procesamiento va a ser mínimo porque solo se manejarían recursos estáticos.
  - **Capacidad en disco:** Teniendo en cuenta que los archivos estáticos se almacenan en dicho servidor, se propone contar con 50 GB de espacio en el disco duro.
- **Servidor de FTP**
  - **Sistema Operativo:** Al igual que el servidor para recueros estáticos, se recomienda usar también CentOS 7.
  - **Memoria RAM:** Se recomienda para este servidor 8 GB de RAM, ya que en este servidor van a estar alojados los documentos y los reportes de las aplicaciones.
  - **Capacidad en disco:** Teniendo en cuenta que este servidor almacenará todos los documentos y reportes, se propone contar con 200 GB de espacio en el disco duro.

Todas las características propuestas anteriormente pueden variar en dependencia de los recursos físicos con que se cuente finalmente para poner en producción el sistema. Cabe mencionar también, a modo de recomendación, la posibilidad de que se realice periódicamente un respaldo, tanto de los documentos del servidor de aplicaciones, como de la información de la base de datos, previniendo de esta manera la perdida de información importante. Se aconseja que la periodicidad de estos respaldos no sea mayor a una semana, de acuerdo a la importancia de la información manejada por la aplicación.

6.2 Servidores

A continuación, se listan los servidores para el entorno de producción.

SERVIDOR	DESCRIPCIÓN	Puertos
sipccontrolacceso.mopc.gov.py	API REST Control de Accesos	80, 8080
sipc.mopc.gov.py	Aplicación web para los funcionarios de la MOPC	80, 8080
sipcestaticos.mopc.gov.py	Recursos estáticos	80, 8080
sipcbd.mopc.gov.py	Base de Datos	5432
sipcjenkins.mopc.gov.py	Jenkins para la integración continua	80

Tabla 2 *Servidores de Producción.*

6.3 Instalación de CentOS 7 en los servidores de producción

Para descargar el ISO [1] de CentOS 7 nos dirigimos a su página de descargas (*https://www.centos.org/download*). Se recomienda descargar la versión *"Minimal"*, ya que esta cuenta con lo básico que proporciona el sistema operativo, siendo lo sugerido cuando se desea configurar un servidor dedicado para poner un sistema en producción. Cabe destacar que es decisión del administrador elegir la versión del sistema operativo en función de las prestaciones con las que cuente el servidor, por ejemplo, instalarlo con interfaz gráfica.

Para iniciar la instalación de Centos 7 solo hay que arrancar el servidor o equipo desde el medio de instalación, ya sea desde un DVD, desde una memoria USB o cargando el ISO en la unidad virtual de CD de la máquina virtual de VMware, Xenserver, Hyper-V o VirtualBox según la elección del administrador. Cuando el servidor o equipo arranque iniciará la instalación y se podrá visualizar la siguiente pantalla.

[1] *Una imagen ISO es un archivo informático donde se almacena una copia o imagen exacta de un sistema de archivos. Se rige por el estándar ISO 9660, que le da nombre. Algunos de los usos más comunes incluyen la distribución de sistemas operativos, por ejemplo: GNU/Linux, FISCH, BSD y Live CD.*

Figura 6. *Instalación de CentOS 7 Minimal - Pantalla 1.*

En esta pantalla se puede seleccionar directamente *"Install Centos 7"* o la opción *"Test this media & install CentOS 7"*, la cual antes de arrancar la instalación verifica el medio de instalación para evitar cualquier problema durante el proceso.

Luego de arrancar se mostrará la interfaz gráfica de la instalación de CentOS 7, como se observa en la siguiente figura.

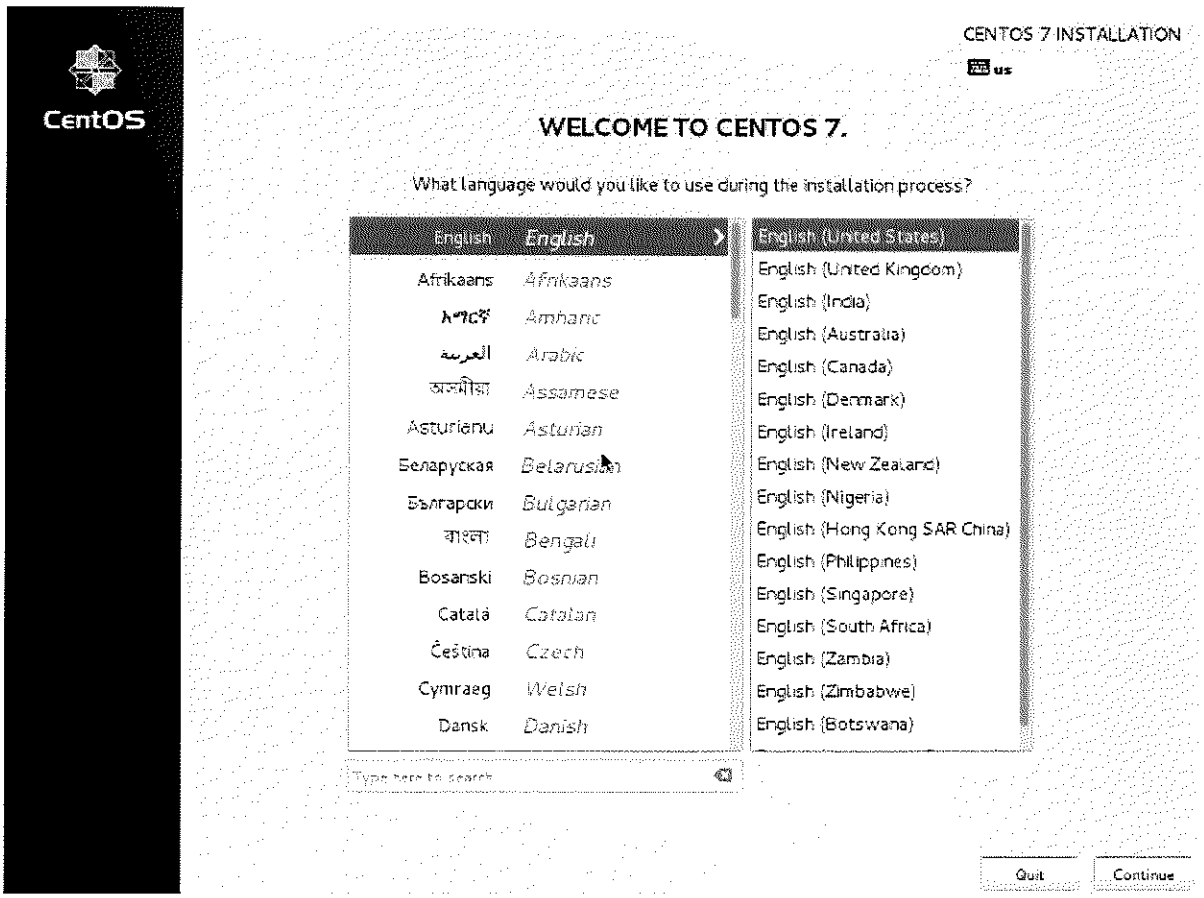

Figura 7. *Instalación de CentOS 7 Minimal - Pantalla 2.*

Por defecto el idioma para el proceso de instalación es el inglés, pero se puede modificar por el de nuestra preferencia, en este caso podemos seleccionar *"Español"* buscando esta opción en el listado. Al hacer esto se configura de forma automática la distribución del teclado, así como las configuraciones de moneda, fecha, etc. En todas las pantallas se debe configurar los elementos necesarios y dar clic en el botón *"Continuar"* o *"Listo"* según corresponda.

Una vez configurado el idioma, se muestra el *"RESUMEN DE LA INSTALACIÓN"* (Figura 8).

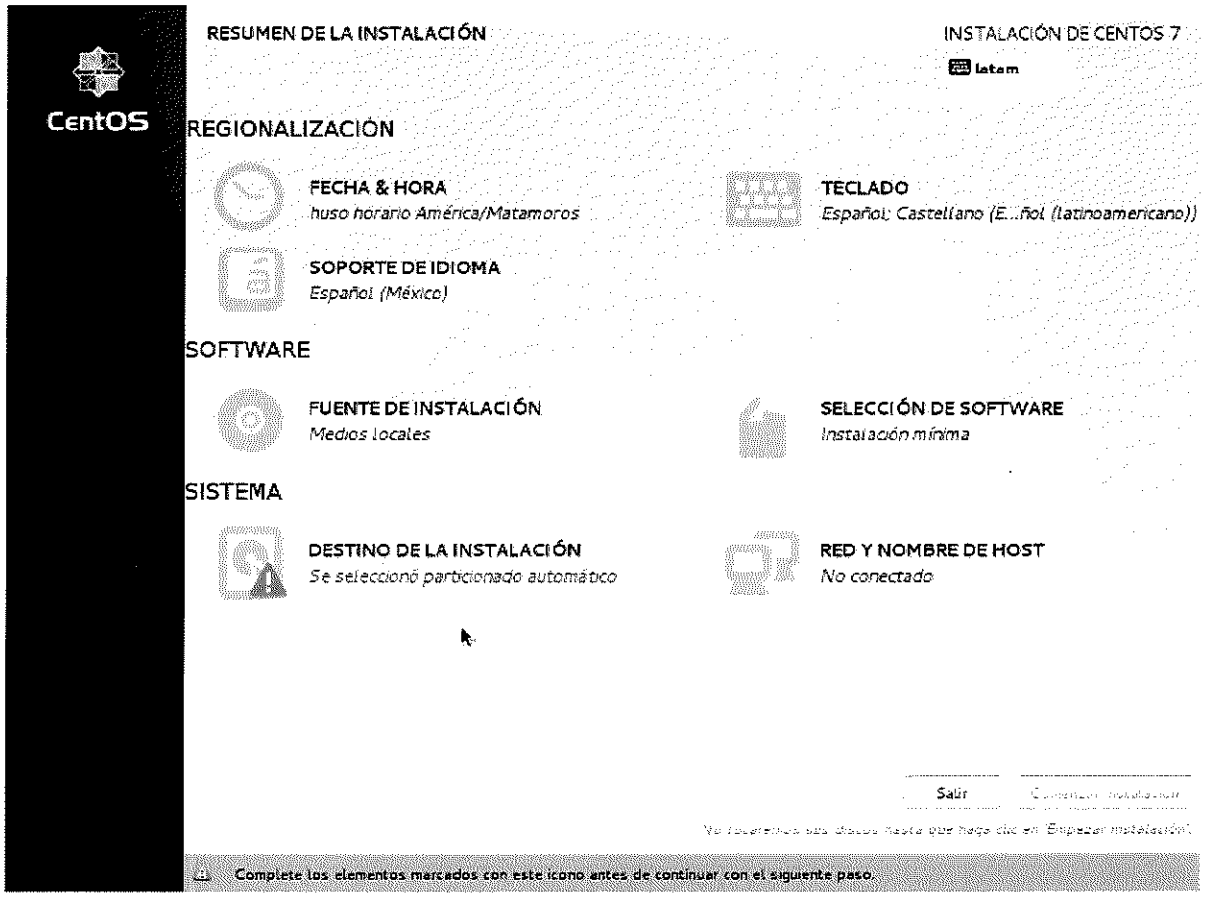

Figura 8. *Instalación de CentOS 7 Minimal - Pantalla 3.*

Esta pantalla permite configurar todos los detalles de la instalación de forma interactiva, es decir, que se puede solo configurar los puntos que se consideren importantes. Para seleccionar cada elemento de la configuración hacemos clic sobre él, por ejemplo, en *"FECHA & HORA"*, *"TECLADO"*, etcétera. Los elementos que tienen un triángulo Naranja y la leyenda en rojo indican que forzosamente se debe configurar para poder continuar con la instalación, como es el caso de *"DESTINO DE LA INSTALACIÓN"*, sobre el cual, al hacer clic, veremos los discos donde podemos instalar (Figura 9).

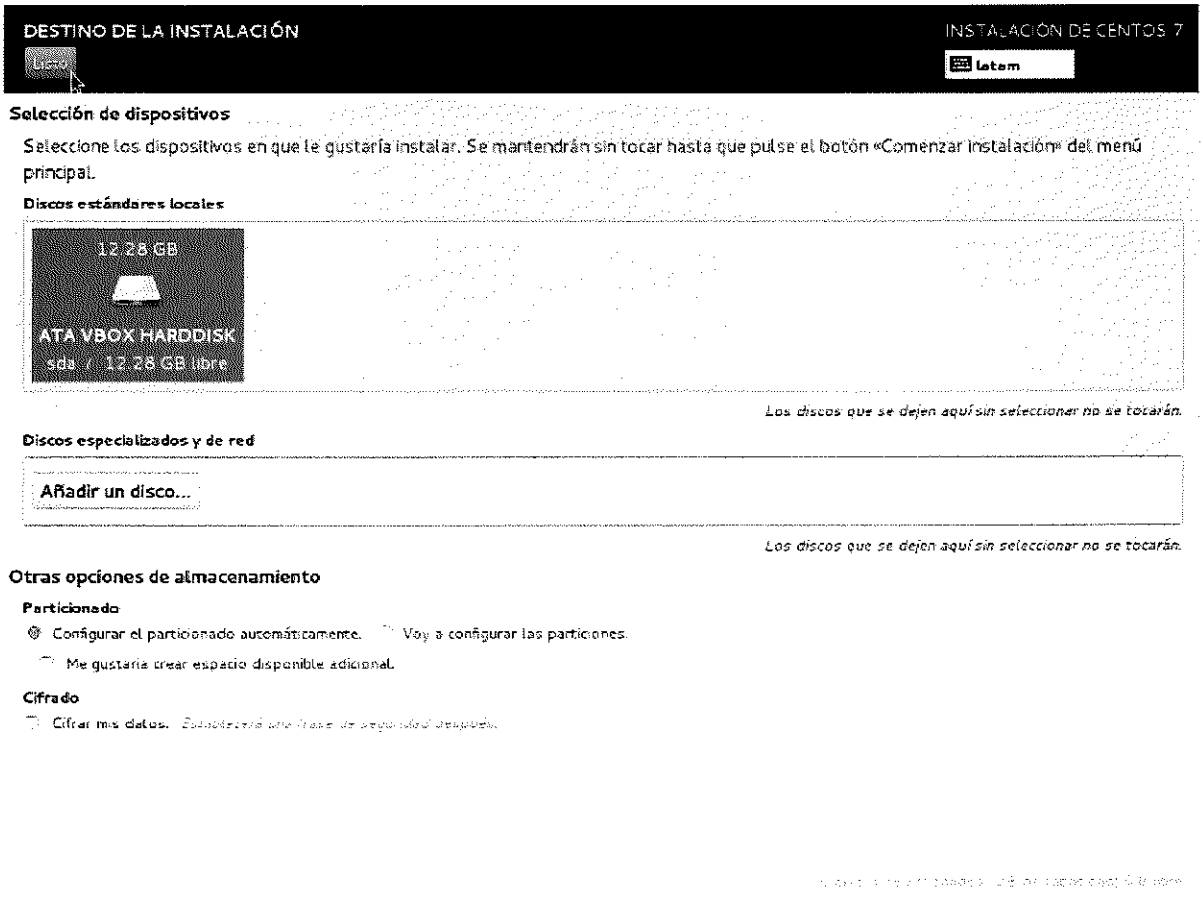

Figura 9. *Instalación de CentOS 7 Minimal - Pantalla 4.*

Por lo general será un único disco, debe estar seleccionado. En caso de querer un esquema diferente de particiones, este es el momento de hacerlo. Una vez realizadas las configuraciones en esta pantalla, hacemos clic en el botón *"Listo"* para guardar los cambios y volver al *"RESUMEN DE LA INSTALACIÓN"*.

La opción "RED Y NOMBRE DE HOST" es importante para la configuración de red. Al hacer clic sobre ella se muestra la siguiente pantalla.

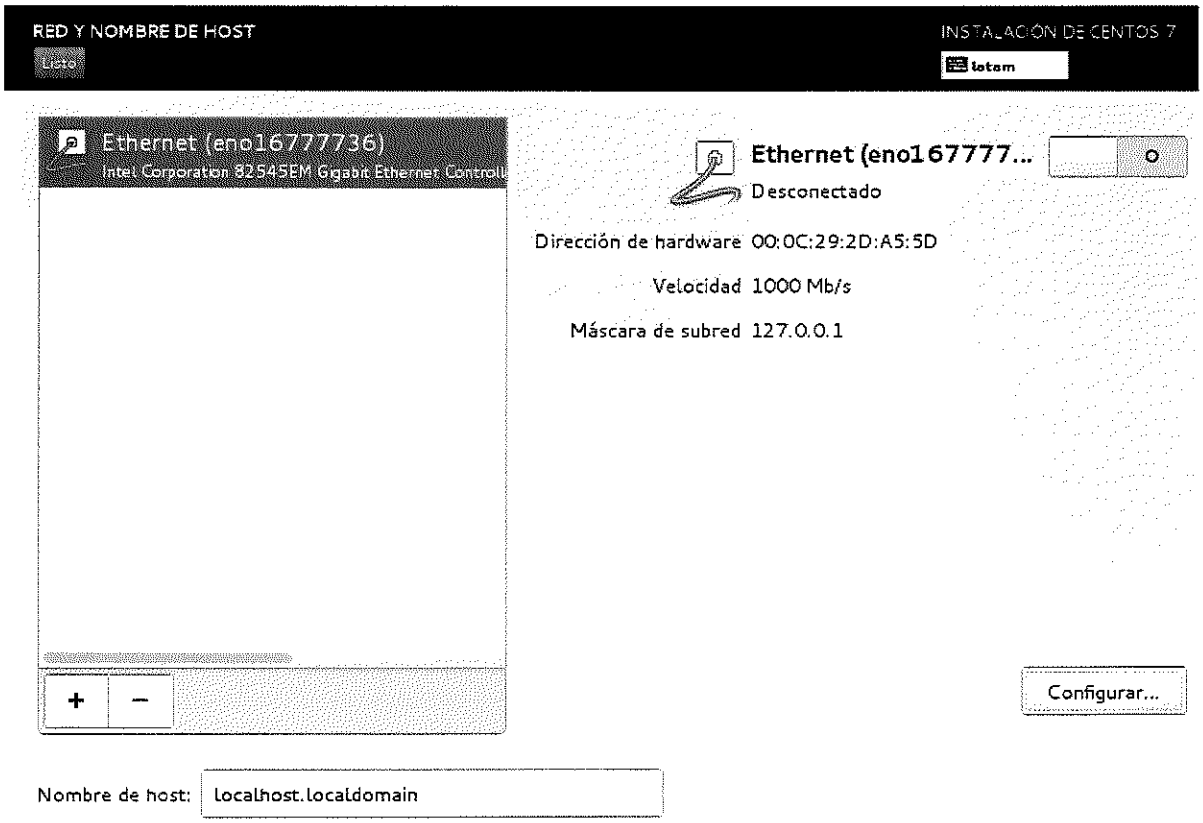

***Figura 10**. Instalación de CentOS 7 Minimal - Pantalla 5.*

Aquí seleccionamos la interfaz de red que se desee configurar. Por lo general, para un servidor no se utilizan direcciones IP dinámicas, sino que se configura una IP fija para cada interfaz de red que el servidor tenga. Para hacerlo solo seleccionamos la interfaz, hacemos clic en el botón *"Configurar..."* y se mostrará una pantalla (Figura 11) donde indicamos la dirección IP que se le asignará a dicha interfaz.

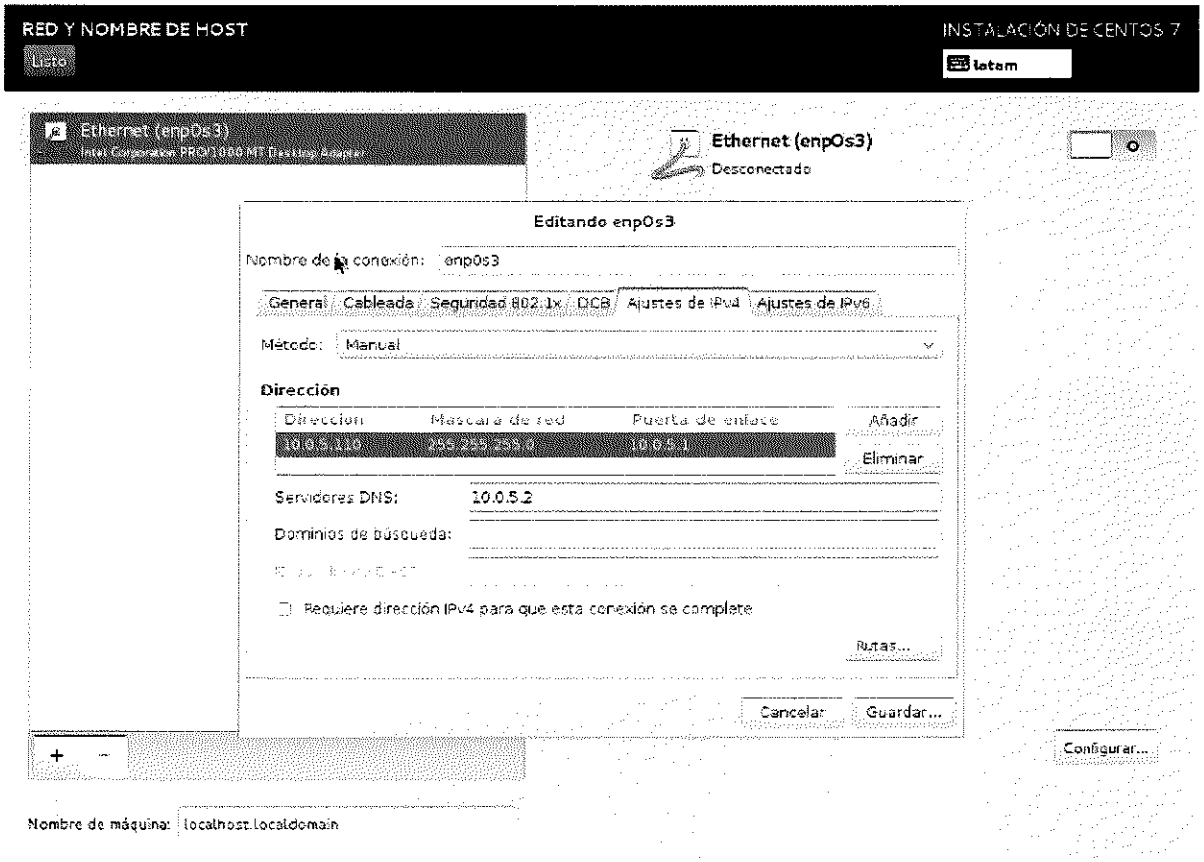

Figura 11. *Instalación de CentOS 7 Minimal - Pantalla 6.*

Al finalizar las configuraciones en esta pantalla hacemos clic en *"Guardar..."* y luego de nuevo al botón *"Listo"*.

Luego podemos configurar la fecha y hora del sistema haciendo clic en la opción *"FECHA & HORA"*, la cual mostrará la siguiente pantalla.

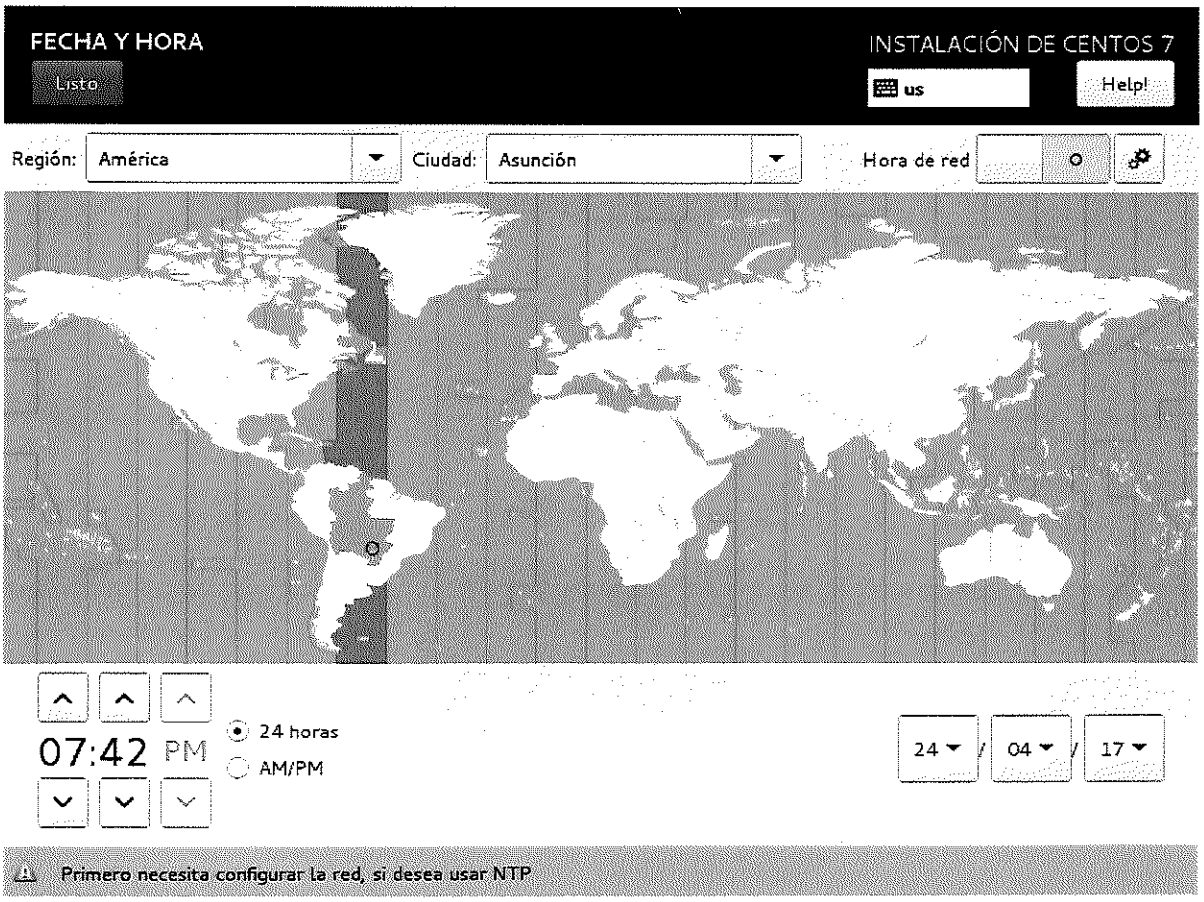

Figura 12. *Instalación de CentOS 7 Minimal - Pantalla 7.*

Aquí se puede seleccionar la región y ciudad donde se encuentra nuestro servidor para tener la zona horaria correctamente configurada, lo cual se realiza haciendo clic sobre el mapa en el punto correspondiente a dicha ciudad. También podemos configurar la hora y la fecha manualmente o indicar que el servidor tome la hora desde un servidor NTP[2] (Network Time Protocol) activando el botón *"Hora de Red"*. Cuando los datos sean los correctos hacemos clic en el botón azul con la leyenda *"Listo"* que se ubica en la parte superior izquierda de la pantalla.

Cuando todos los elementos están configurados se activa el botón *"Comenzar Instalación"* en la parte inferior derecha, al cual debemos dar clic, tal como muestra la siguiente figura.

[2] *NTP (Network Time Protocol) es un protocolo de Internet para sincronizar los relojes de los sistemas informáticos a través del enrutamiento de paquetes en redes con latencia variable. NTP utiliza UDP como su capa de transporte, usando el puerto 123.*

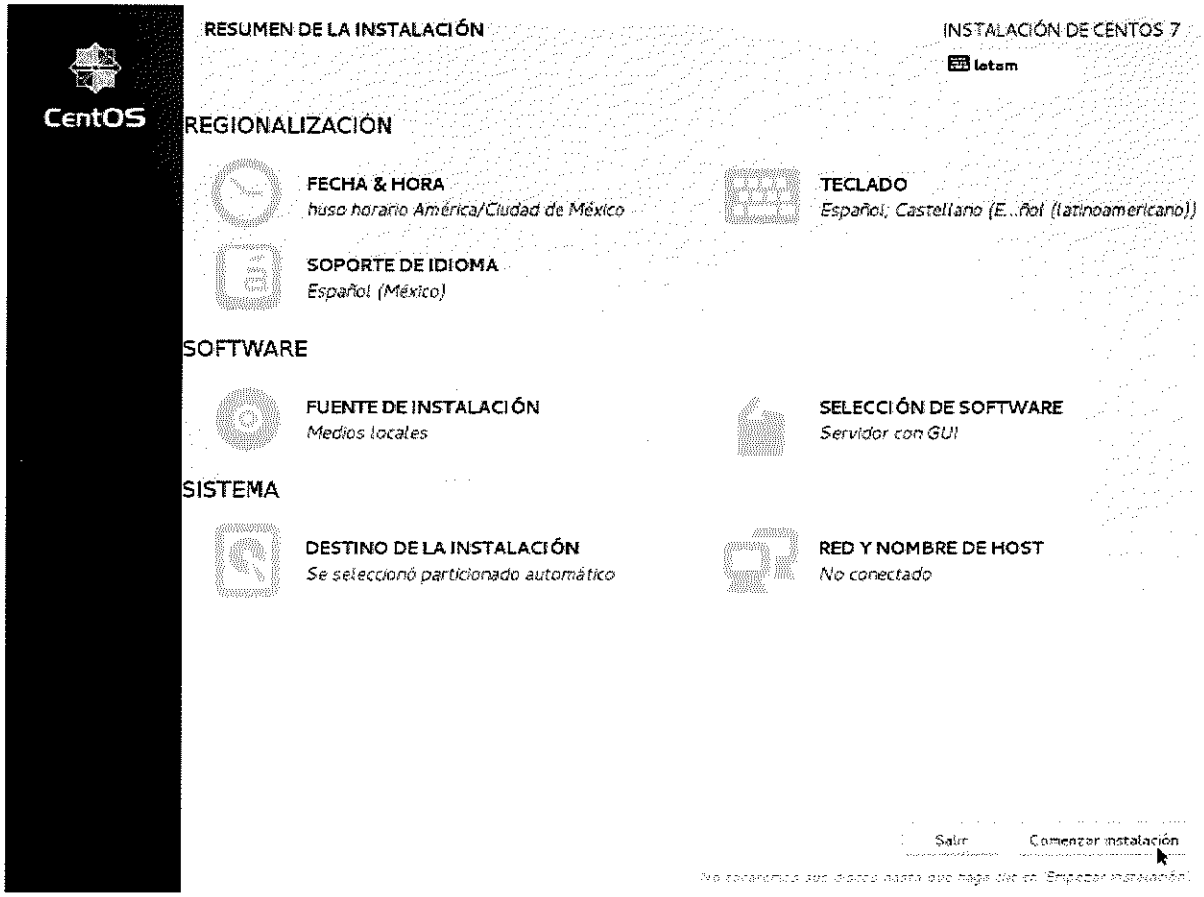

Figura 13. *Instalación de CentOS 7 Minimal - Pantalla 7.*

Al iniciar la instalación podemos indicar la contraseña de root y los usuarios que tendrá el servidor, en caso de que se desee agregar alguno (Figura 14).

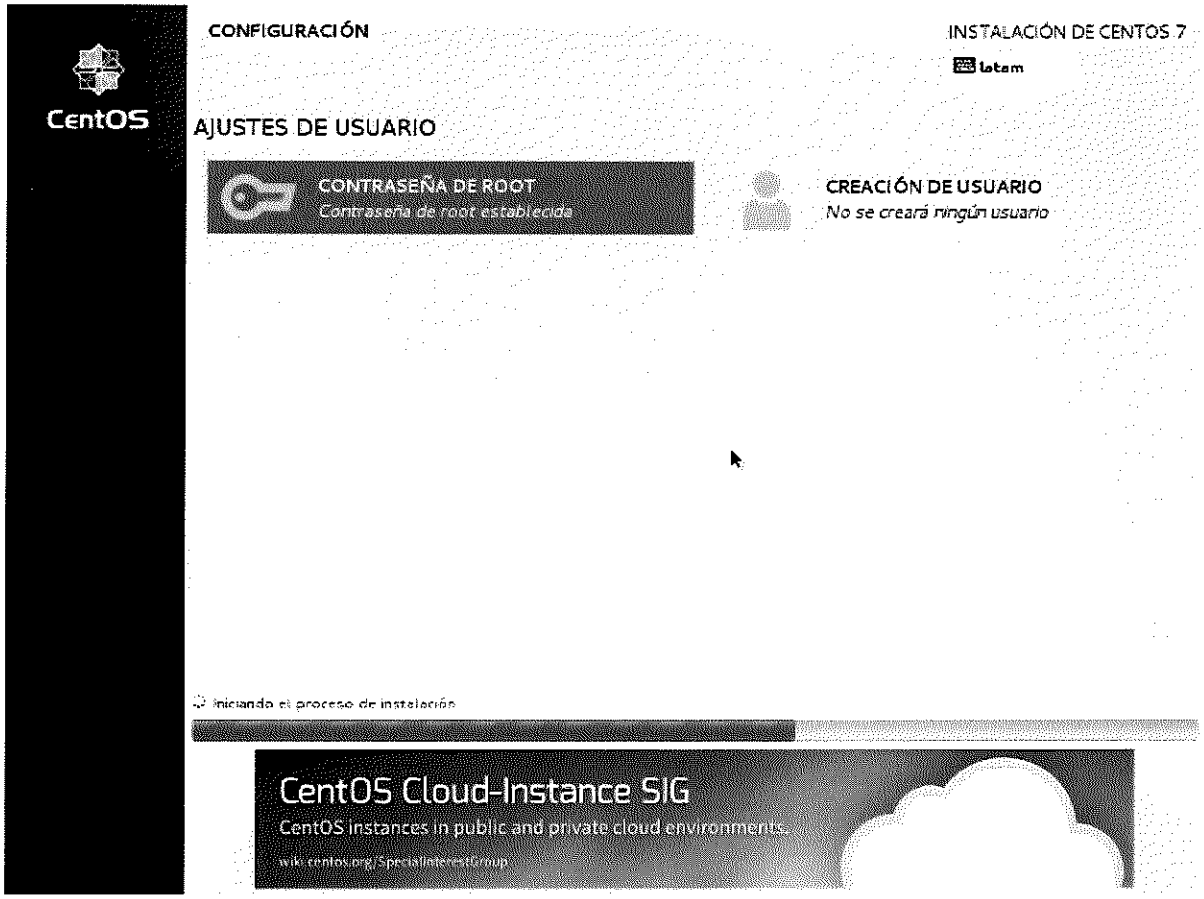

Figura 14.*Instalación de CentOS 7 Minimal - Pantalla 8.*

Para indicar la contraseña de root solo damos clic en el espacio de *"CONTRASEÑA DE ROOT"*, lo cual hará que se muestre la siguiente pantalla.

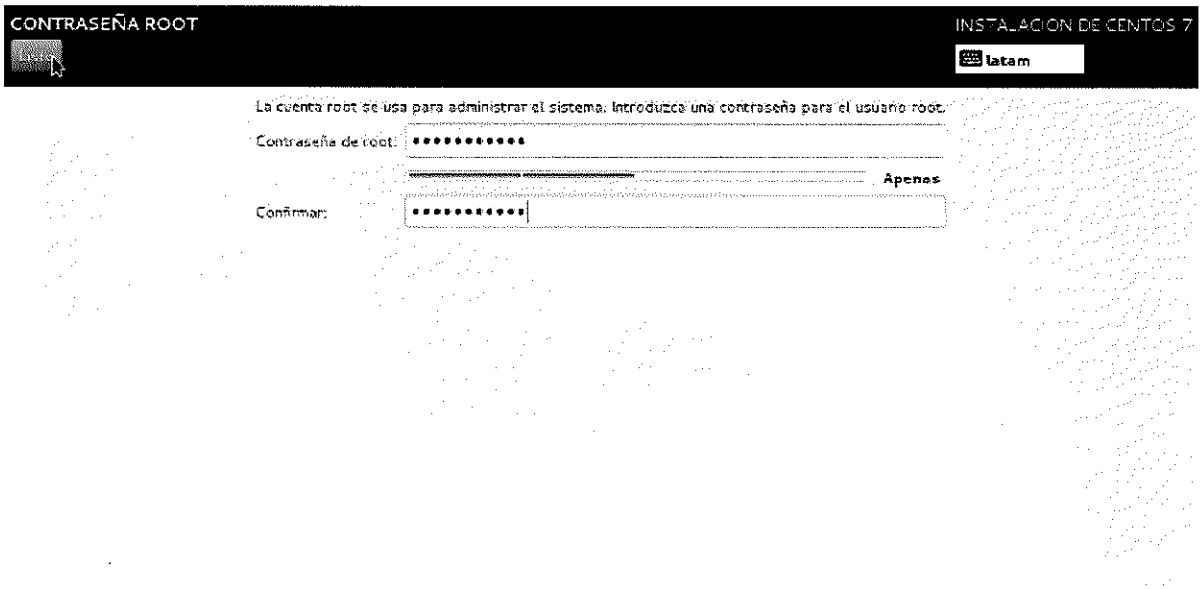

Figura 15.*Instalación de CentOS 7 Minimal - Pantalla 9.*

Aquí tecleamos una cadena de texto en el campo *"Contraseña de root"*. Al teclear se indica si la contraseña que se está escribiendo es lo suficientemente segura. Al terminar, solo debemos confirmarla en el campo *"Confirmar"* y hacer clic en el botón *"Listo"*.

En este punto no crearemos ningún usuario, ya que este paso se hará posteriormente, tal como se explica en la sección ***Configuración e instalación del software requerido en los servidores de producción***, abordada más adelante en este documento.

Una vez que se ha establecido la contraseña de root, se han creado los usuarios (si es necesario) y se han copiado todos los paquetes de software de acuerdo a la configuración realizada, se mostrará un mensaje indicando que la instalación se ha realizado y se podrá reiniciar el servidor dando clic en el botón *"Reiniciar"*, el cual se ubica en la parte inferior derecha de la pantalla que se muestra a continuación.

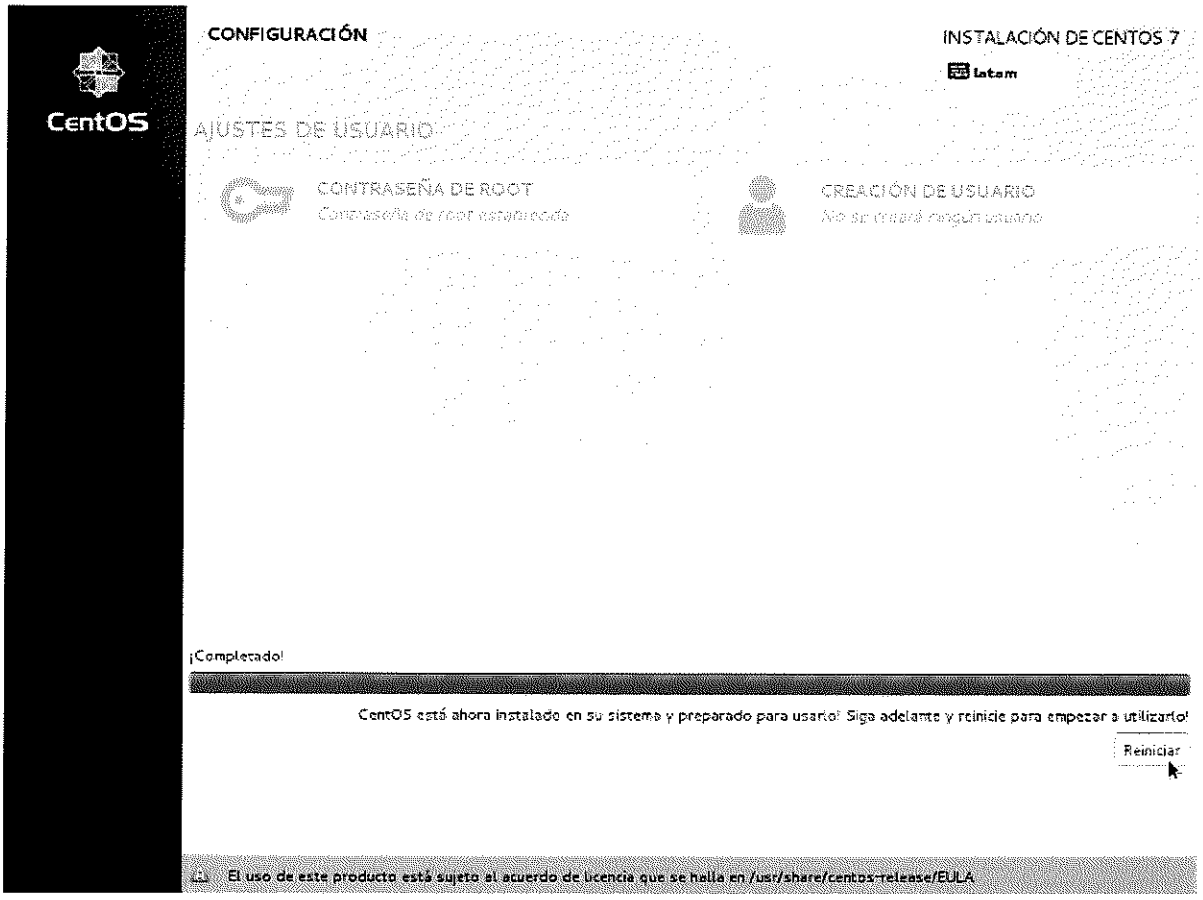

Figura 16. *Instalación de CentOS 7 Minimal - Pantalla 10.*

Al reiniciar el servidor recién instalado, ya no se requiere el medio de instalación, ya sea un DVD, un USB o el ISO montado en la unidad virtual, por lo que se debe retirar. Al arrancar CentOS 7 veremos las pantallas siguientes.

Figura 17.*Instalación de CentOS 7 Minimal - Pantalla 11.*

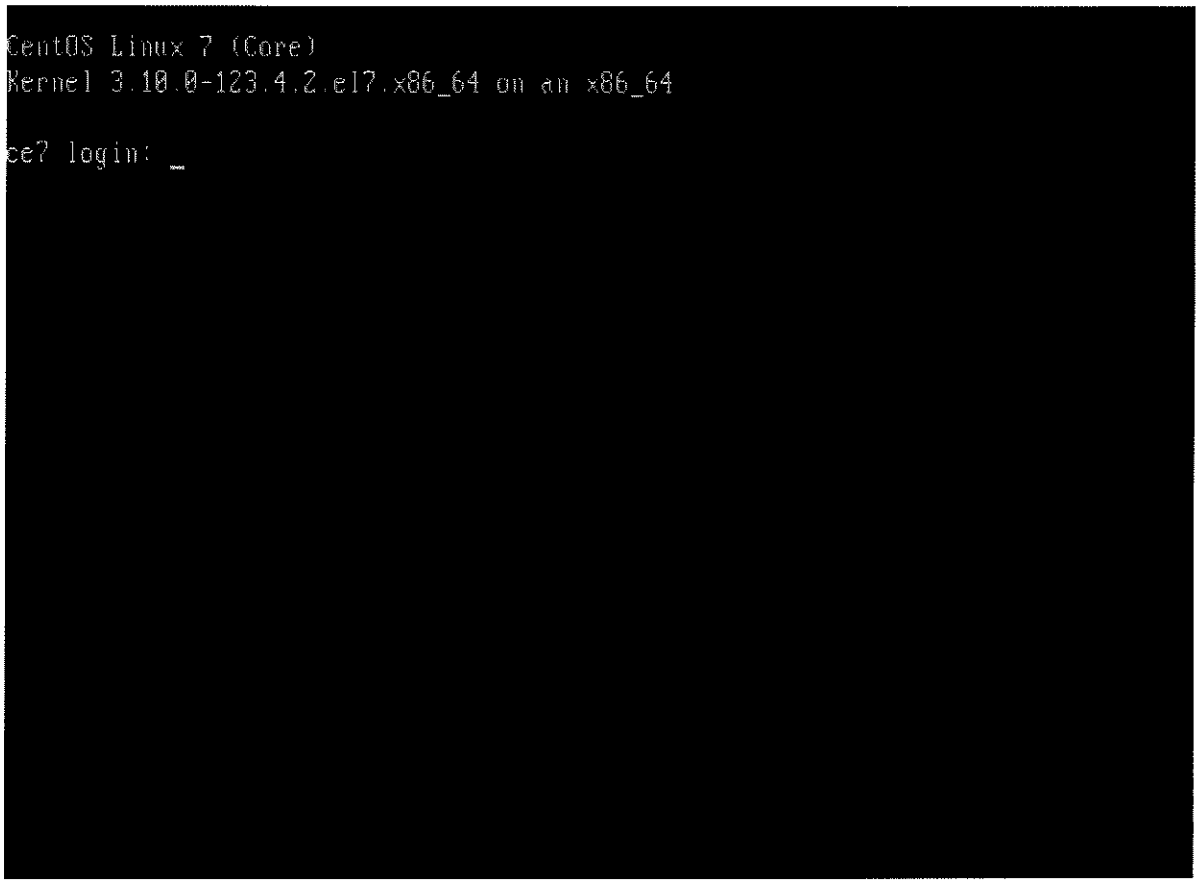

Figura 18.*Instalación de CentOS 7 Minimal - Pantalla 12.*

7.1 Configuración e instalación del software requerido en los servidores de producción

Con el sistema operativo ya instalado, la primera tarea recomendada es actualizar el sistema operativo para corregir algún error o problema de seguridad. Para ello nos autenticamos como root en la consola y ejecutamos el comando:

```
yum -y update
```

y posteriormente reiniciamos con el comando:

```
reboot
```

Una vez que vuelva a iniciar el sistema, nos dirigimos a la carpeta ***RUTA_BASE_REPO***\ *productos \simffaso01\06.despliegue y* ***RUTA_BASE_REPO*** \ *productos* \ *simffaso01*\ *06. despliegue* en la máquina donde tengamos descargado el repositorio del proyecto y copiamos remotamente el directorio *"Recursos de Instalación"* con todo su contenido a cualquier ubicación del servidor.

En este directorio se encuentran los archivos denominados **postgres-install.sh** y **wildfly-install.sh**, los cuales permiten instalar y configurar de manera cómoda el servidor de base de datos PostgreSQL en su versión 10.4 y la versión 13.0.0 Final del servidor de aplicaciones Wildfly, respectivamente. Para ejecutar estos scripts debemos estar autenticados en la consola como root, ubicarnos con el comando **cd** en la carpeta que acabamos de copiar y finalmente los ejecutamos con el comando **sh**, por ejemplo:

```
sh postgres-install.sh
```

Se recomienda instalar primero el servidor de base de datos y luego el de aplicaciones. Al ejecutarse, cada uno de los scripts mencionados va indicando paso a paso en la pantalla las operaciones que va realizando y en caso de que sea necesario que el usuario introduzca algún valor para un parámetro de la configuración durante el proceso, se espera por la especificación de dicho valor para continuar.

Con el Wildfly instalado, se puede utilizar el servicio creado para su administración mediante los siguientes comandos:

service wildfly status (Muestra el estado del servicio)

service wildfly start (Inicia el servicio)

service wildfly stop (Detiene el servicio)

service wildfly restart (Reinicia el servicio)

Esta instalación y configuración contiene las operaciones básicas para dejar listo el entorno de producción y que se pueda utilizar el sistema. No se incluye la configuración del servidor DNS ya que depende de los administradores de red de la institución. Cualquier otra configuración necesaria, es posible realizarla siguiendo la documentación de los distintos softwares utilizados para el correcto funcionamiento de la aplicación.