CHIBCHAWEB HOSTING PLATFORM
APLICACIÓN PARA UNA COMPAÑÍA DE HOSPEDAJE DE PÁGINAS WEB

DOCUMENTO DE DISEÑO Y ARQUITECTURA

Grupo A

Ingenieros:
Andrés José Acevedo Ardila
Brayan Steven Sánchez Casilima
Christian Camilo Lancheros Sánchez
Juan Esteban Ordoñez Sandoval
Karen Tatiana Bravo Rodríguez
Luis Felipe Mayorga Tibaquicha

Universidad Distrital Francisco José de Caldas
Facultad de Ingeniería
Ingeniería de sistemas

Bogotá D.C., Agosto 2025

ChibchaWeb
Hosting Platform

Grupo A

Documento de Diseño y Arquitectura

2025

Índice

1 Listado de Figuras

Índice de figuras

2 Listado de Tablas

Índice de cuadros

3 Descripción del Documento

3.1 Propósito y Audiencia

Este documento tiene como propósito describir la arquitectura de software propuesta para el desarrollo de una solución integral que apoye los procesos operativos de ChibchaWeb, una compañía dedicada al hospedaje de páginas web con sede en Sugamuxi y proyección de expansión internacional.
El documento presenta los componentes clave del sistema, los requerimientos funcionales y no funcionales, los patrones de diseño seleccionados, y los modelos arquitectónicos que sustentan la solución. También detalla la metodología de desarrollo adoptada, las decisiones técnicas tomadas y su justificación.
Este documento está dirigido a los siguientes perfiles:

- Equipo de Desarrollo: Para comprender la arquitectura, los patrones y las tecnologías definidas que guiarán la implementación del sistema.
- Clientes y Stakeholders de ChibchaWeb: Para validar que los requerimientos funcionales y no funcionales del negocio han sido entendidos y serán satisfechos por el sistema.
- Equipo de QA y Soporte Técnico: Para conocer los aspectos técnicos relevantes de la solución y planificar las estrategias de prueba y soporte.
- Líderes de Proyecto y Gerencia: Para tomar decisiones informadas sobre tiempos, recursos y alineación con objetivos estratégicos.

3.2 Organización del Documento

Este documento empieza con un panorama general del proyecto consistente en objetivos generales y específicos, seguidos de una contextualización del entorno y conceptos importante del proyecto, esto con el fin de generar un marco teórico que aclare y mejore la organización empresarial.
Luego de esto, se define el modelo de desarrollo y por lo tanto la organización del equipo de desarrollo. Después se tiene en cuenta las motivaciones y los riesgos relacionados con el desarrollo del proyecto, con el fin de estimar la utilidad. Además de las restricciones (tecnológico, negocio) y la gestión de calidad (con parámetros medibles para cumplir objetivos).
Por último, se definen los diagramas principales para empezar el proceso del producto, describiendo los diferentes modelos y entidades del desarrollo. Para terminar, se hace una pequeña retroalimentación con el fin de reconocer errores dentro del desarrollo del proyecto y corregirlos en próximos proyectos.

3.3 Convenciones

3.4 Terminología y Definiciones

4 Generalidades del proyecto

4.1 Problema por resolver

ChibchaWeb, una empresa de hospedaje de sitios web ubicada en Sugamuxi, ha experimentado un crecimiento constante en su base de clientes, principalmente en Colombia y países cercanos, y se proyecta una expansión hacia otros continentes como África. Actualmente administra miles de sitios web bajo distintos paquetes de hosting (Chibcha-Platino, Chibcha-Plata y Chibcha-Oro) sobre plataformas Windows y Unix. Los clientes pueden escoger planes de pago flexibles (mensual, trimestral, semestral o anual), y aunque el método principal de pago es con tarjeta de crédito, se planea habilitar próximamente otros medios como PSE.

El modelo de negocio incluye además una red de distribuidores categorizados como BÁSICOS o PREMIUM, con diferentes esquemas de comisiones. ChibchaWeb también intermedia en el registro y transferencia de dominios a través de terceros registradores, ofreciendo este servicio adicional a sus clientes. A medida que el volumen de operaciones crece y los procesos se diversifican, la compañía enfrenta varios desafíos operativos y administrativos.

Actualmente, ChibchaWeb carece de una solución unificada que automatice y centralice sus procesos comerciales, operativos y de atención al cliente. Esto genera ineficiencias en la gestión de perfiles, pagos, comisiones, atención técnica y comunicación con registradores externos. La falta de un sistema integral limita la capacidad de escalar operaciones, introducir nuevos métodos de pago, y garantizar trazabilidad y control sobre los procesos internos.

4.2 Descripción general del sistema a desarrollar

El sistema a desarrollar para ChibchaWeb es una plataforma integral de gestión de servicios de hospedaje web, diseñada para automatizar, centralizar y optimizar los procesos operativos clave de la compañía. Esta solución tecnológica permitirá gestionar clientes, empleados, distribuidores, dominios, pagos y soporte técnico de manera eficiente, segura y escalable.

El sistema contará con módulos para la creación y administración de perfiles de clientes y empleados, y almacenará la información en una base de datos relacional optimizada incluyendo la validación de información personal y financiera garantizando la integridad evitando ciclos en esta; buscar y adquirir dominios mediante un carrito; adquirir paquetes de hosting.

El sistema contará con roles diferenciados por tipo de cuenta; búsquedas y consultas detalladas de todos los actores del sistema, así como herramientas para la generación de archivos de transferencia electrónica y seguimiento de actividades internas del proyecto; gestión de solicitudes de dominios con generación de archivos XML para su envío a registradores externos; Cálculo automático de comisiones a distribuidores según su categoría

(BÁSICO o PREMIUM); y un módulo de soporte técnico basado en tickets que garantiza el seguimiento y resolución estructurada de problemas reportados.

La solución incluirá una interfaz web en React, backend en FastAPI y base de datos en MySQL. Se aplicarán buenas prácticas de diseño para mantener una l ógica de negocio clara y una estructura coherente en el modelo de datos.

4.3 Objetivos de la solución

4.3.1 Objetivo general

- Desarrollar un sistema web que permita registrar, gestionar y monitorear cuentas de usuarios, adquisición de dominios y facturación de manera eficiente y centralizada.

4.3.2 Objetivos específicos

- Diseñar una base de datos relacional normalizada y sin ciclos innecesarios para soportar la lógica del sistema.
- Desarrollar una interfaz web funcional y responsiva utilizando React y Bootstrap para facilitar la interacción del usuario con el sistema.
- Construir un backend en FastAPI que gestione la lógica del negocio, la autenticación de usuarios y la comunicación con la base de datos.

4.4 Stakeholders

Stakeholder	Descripción
Cliente	Persona que encargó el proyecto y financia económicamente el desarrollo de la aplicación web.
Desarrolladores	Grupo encargado de desarrollar el código y las interfaces necesarias para el proyecto.
Usuarios finales	Posibles compradores interesados en adquirir dominios web.
Mantenedores	Desarrolladores encargados de actualizar el código y la página.
Product owner	Encargado de gestionar los requerimientos y verificar su cumplimiento/correcto desarrollo.

Cuadro 1: Stakeholders

5 Descripción y justificación de la metodología

Para el desarrollo del sistema de información de ChibchaWeb, se ha seleccionado la metodología ágil Scrum. Esta elección se basa en las características del proyecto y las necesidades del cliente, que requieren flexibilidad, entregas parciales funcionales y un involucramiento constante de los interesados en el proceso.

5.1 Scrum

Scrum es un marco ágil de desarrollo iterativo e incremental que permite entregar valor de manera temprana y continua a través de sprints (iteraciones de tiempo fijo, típicamente de 1 a 4 semanas).
En Scrum, el trabajo se organiza en:

- Product Backlog: Lista priorizada de funcionalidades y requerimientos del sistema.
- Sprint Backlog: Conjunto de tareas seleccionadas para el Sprint en curso.
- Sprint Review y Sprint Retrospective: Evaluaciones al finalizar cada sprint, enfocadas en la mejora continua.

5.1.1 Roles en Scrum aplicados al proyecto

- Product Owner: Representa los intereses de ChibchaWeb. Define y prioriza el Product Backlog.
- Scrum Master: Facilita el marco de trabajo Scrum, elimina impedimentos y promueve buenas prácticas.
- Development Team: Grupo autoorganizado de desarrolladores encargados de diseñar, codificar, probar e implementar el sistema.

5.2 Justificación de su uso en este proyecto

- Adaptabilidad a cambios de requisitos: Dado que el negocio de hospedaje web evoluciona rápidamente (nuevas formas de pago, expansión internacional), Scrum permite adaptarse con agilidad.
- Entrega continua de valor: Con cada sprint, se entregan versiones funcionales del sistema que pueden ser probadas por el cliente.
- Comunicación constante con el cliente: La retroalimentación frecuente mejora la alineación entre lo que se construye y lo que el cliente realmente necesita.
- Reducción del riesgo: La entrega incremental permite detectar errores o desviaciones de manera temprana.

- Fomento de la colaboración y mejora continua: Las reuniones retrospectivas y la autoorganización del equipo fortalecen la eficiencia del proceso.

6 Especificación y Recabación de Requerimientos Funcionales

7 Motivadores Arquitecturales

7.1 Motivadores de Negocio

Los motivadores de negocio que se mostrarán a continuación definen las razones por las cuales ChibchaWeb requiere una nueva solución de software. Entre las cuales se encuentran:

7.1.1 Expansión internacional del negocio

ChibchaWeb, actualmente con clientes en Colombia y países vecinos, proyecta una expansión hacia mercados internacionales como África. Esto requiere una plataforma escalable y flexible que soporte la gestión de clientes, dominios y facturación en diferentes regiones.

7.1.2 Automatización de procesos clave

El sistema busca automatizar la creación, edición y eliminación de perfiles de clientes, empleados y distribuidores, reduciendo la carga operativa y los errores humanos, y facilitando la trazabilidad de operaciones críticas.

7.1.3 Mejora en la atención al cliente y soporte técnic

Al permitir el registro y seguimiento de tickets, el sistema mejorará el proceso de resolución de problemas, permitiendo al equipo de soporte priorizar incidencias y mejorar los tiempos de respuesta, alineándose con los niveles de servicio ofrecidos.

7.1.4 Gestión y segmentación de distribuidores

Con la existencia de distribuidores BÁSICOS y PREMIUM, el sistema debe permitir su categorización, cálculo de comisiones correspondientes (10

7.1.5 Flexibilidad en métodos de pago

Actualmente se aceptan tarjetas de crédito, pero se planea incluir otros métodos como PSE. Esto requiere una arquitectura extensible que permita integrar fácilmente nuevos métodos de pago en el futuro sin afectar la operatividad actual.

7.2 Restricciones de Tecnología

El desarrollo de la plataforma para ChibchaWeb estará sujeto a las siguientes restricciones tecnológicas, derivadas de decisiones de arquitectura, compatibilidad, mantenibilidad y recursos disponibles:

Tecnologías definidas por el equipo de desarrollo

7.2.1 Frontend

Se utilizará React.js como framework principal para la construcción de la interfaz de usuario, debido a su eficiencia, modularidad y gran comunidad de soporte.

7.2.2 Backend

Se empleará FastAPI con Python, por su rendimiento, facilidad de definición de servicios RESTful, y compatibilidad con documentación automática.

7.2.3 Base de datos

Se usará MySQL, una base de datos relacional robusta y ampliamente soportada, ideal para mantener la integridad referencial y relaciones entre entidades complejas (clientes, empleados, dominios, etc.).

7.2.4 Control de versiones y colaboración

El proyecto estará versionado mediante GitHub, lo cual permitirá control de cambios, trabajo colaborativo y despliegue continuo si se requiere.

7.3 Restricciones de Negocio

El desarrollo y funcionamiento del sistema estarán sujetos a diversas restricciones impuestas por las necesidades, normativas y condiciones del entorno comercial del proyecto. Estas restricciones afectan la manera en que se diseña y entrega la solución, e incluyen:

7.3.1 Políticas de manejo de cuentas

Solo se permitirán 4 tipos de cuentas: cliente, administrador, empleado y distribuidor, cada una con permisos y accesos claramente definidos. No se admitirán roles personalizados fuera de estas categorías durante la fase inicial del sistema.

7.3.2 Manejo de dominios basado en disponibilidad externa

La compra de dominios dependerá de su disponibilidad en registros externos. Si el dominio ya está tomado, no podrá ser adquirido ni facturado, lo cual impone una dependencia con proveedores externos y puede requerir lógica de validación en tiempo real.

7.4 Atributos de Calidad

Durante el desarrollo del sistema ChibchaWeb, se identificaron ciertos atributos de calidad que orientaron decisiones técnicas y de arquitectura. A continuación, se describen los principales, en función de la experiencia práctica con el sistema ya desplegado y en funcionamiento.

7.4.1 Disponibilidad

El sistema actualmente está desplegado mediante Railway, por lo tanto, su disponibilidad depende del servicio que ofrece esta plataforma. No se cuenta con una infraestructura propia, pero la integración entre base de datos, backend y frontend se mantiene activa y estable mientras Railway esté operativo. Esta disponibilidad es suficiente para los fines actuales, aunque si en un futuro fuera necesario se podría considerar una solución con mayor control sobre el entorno de producción.

7.4.2 Seguridad

Se implementó un sistema de autenticación por correo electrónico, en el cual los usuarios reciben un código de verificación. Solo quienes completen este proceso pueden acceder a las funciones reservadas para clientes. Esta medida ha resultado efectiva para restringir accesos y proteger la información del usuario, y se aplican validaciones en los formularios.

7.4.3 Rendimiento

La aplicación responde de forma fluida gracias a la conexión directa con las APIs del backend. No se ven demoras significativas al realizar operaciones como búsquedas, registros, etc.

7.4.4 Escalabilidad

Actualmente el sistema permite operaciones como búsqueda de dominios y la “compra” como tal, lo cual es una base para un entorno escalable. Aunque todavía no se ofrecen servicios de hosting reales desde la plataforma, el sistema está preparado para incluir esa funcionalidad más adelante, como tal el nucleo del sistema nos permite esto

7.4.5 Mantenibilidad

El código está organizado de forma clara, permitiendo modificar o añadir funcionalidades sin afectar lo ya implementado. Se han realizado ajustes menores y correcciones sin mayores inconvenientes, lo que demuestra que el sistema es mantenible. Además, se pueden actualizar componentes del frontend o backend de manera independiente.

7.4.6 Usabilidad

Desde la perspectiva del cliente, la interfaz es bastante intuitiva. Las funciones están bien distribuidas y no es necesario tener conocimientos técnicos para navegar o realizar operaciones básicas. En el caso del personal de soporte o empleados, hay más funciones que requieren cierta formación, como el manejo de tickets, pero están diseñadas de forma clara.

7.4.7 Interoperabilidad

El sistema se comunica con servicios externos mediante peticiones web, especialmente para consultar la disponibilidad de dominios. Para ello, se implementó un proceso de scraping que permite validar si un dominio está disponible o no.

7.4.8 Recabación y justificació de Requerimientos No Funcionales

Durante la planificación y construcción del sistema ChibchaWeb, se identificaron diversos requerimientos no funcionales a partir de las necesidades específicas del proyecto, la experiencia con despliegues reales, y las condiciones técnicas impuestas por el entorno en el que opera la aplicación. Estos requerimientos no son visibles directamente para el usuario final, pero son determinantes para que el sistema sea confiable, seguro y sostenible en el tiempo.

- Las operaciones realizadas por el usuario (como la búsqueda de dominios, la carga de formularios o la visualización de datos) no es de un tiempo significativo o excesivo, esto mejora la experiencia del usuario, esto gracias al uso de las Apis
- Aunque actualmente se encuentra todo en el servicio de railway para el despligue, fuera de eso el sistema es funcional y estable.
- Solo los usuarios verificados deben tener acceso a funcionalidades sensibles del sistema. Además, los datos sensibles como contraseña están protegidos
- El sistema esta preparado para soportar un aumento progresivo de usuarios y funcionalidades sin comprometer el rendimiento, aunque recalcar que al menos en esta fase la disponibilidad no depende tanto de nosotros, sino del servicio de railway
- En diferentes momentos del proyecto se han realizado cambios o mejoras con relativa facilidad, lo que indica que se ha logrado un diseño mantenible.
- Ya se implementó una primera forma de integración con servicios de disponibilidad de dominios usando scraping. Esta capacidad de interactuar con sistemas externos es la clave para estas funcionalidades.

7.4.9 Escenarios de Calidad

Los escenarios de calidad permiten evaluar cómo se comporta el sistema ante situaciones específicas que afectan atributos no funcionales como el rendimiento, la seguridad o la disponibilidad. A continuación se presentan algunos escenarios relevantes que reflejan condiciones reales o esperadas para la plataforma ChibchaWeb:

Consulta masiva de dominios disponibles

- **Contexto:** Un distribuidor accede a la plataforma y realiza varias búsquedas consecutivas para verificar la disponibilidad de dominios para sus clientes.
- **Estimulación:** Se ejecutan 10 a 15 búsquedas en un corto periodo de tiempo.
- **Respuesta esperada:** El sistema podra responder a estas consultas en un tiempo no tan significativo, sin errores ni bloqueos.

Intento de acceso sin verificación de correo

- **Contexto:** Un usuario se registra pero no verifica su correo electrónico.
- **Estimulación:** Intenta acceder a las funcionalidades de cliente como compra de dominio o visualización del perfil.
- **Respuesta esperada:** El sistema bloquea el acceso y solicita la verificación de correo antes de continuar.

Actualización del backend sin detener el sistema completo

- **Contexto:** Se implementa una mejora en una API del backend.
- **Estimulación:** Se despliega el nuevo backend mientras la base de datos y el frontend siguen activos.
- **Respuesta esperada:** El sistema sigue funcionando normalmente o se interrumpe solo por un corto tiempo. No se generan inconsistencias.

Interacción de un cliente no técnico

- **Contexto:** Un cliente nuevo accede al sistema para registrar su primer dominio.
- **Estimulación:** Usa la interfaz para buscar un dominio, registrar sus datos y simular un pago.
- **Respuesta esperada:** El cliente puede completar la operación sin necesidad de asistencia. Los pasos son claros y comprensibles.

Caída temporal del servicio Railway

- **Contexto:** Railway presenta una interrupción temporal.
- **Estimulación:** Usuarios intentan ingresar al sistema durante el tiempo en que el servicio está inactivo.

- **Respuesta esperada:** El sistema no responde (fuera de servicio), pero se recupera automáticamente cuando Railway vuelve a estar disponible, sin pérdida de información.

8 Contexto

8.1 Escenarios operacionales

Estos Escenarios operacionales representan las interacciones comunes o típicas que pasan entre los diferentes actores de la plataforma de ChibchaWeb, Cada escenario describe como un usuario o actor logra el objetivo deseado utilizando el sistema en un contexto especifico

8.1.1 Escenario 1: Registro y verificación de cuenta

- Actor: Usuario
- Descripción: Un usuario se registra en la plataforma ingresando sus datos personales. Luego, valida su cuenta a través de un token enviado por correo electrónico.
- Resultado esperado: El usuario puede iniciar sesión y acceder a las funciones de su rol.

8.1.2 Escenario 2: Compra de un paquete de hosting

- Actor: Cliente
- Descripción: El cliente agrega un método de pago, selecciona un paquete, realiza el pago y se activa el servicio.
- Resultado esperado: El sistema registra la compra.

8.1.3 Escenario 3: Registro de solicitud de dominio

- Actor: Usuario (Cliente o Distribuidor)
- Descripción: El usuario busca un dominio, valida su disponibilidad y envía una solicitud.
- Resultado esperado: La solicitud es procesada y almacenada para generación de archivos XML.

8.1.4 Escenario 4: Atención de ticket de soporte

- Actor: Usuario
- Descripción: El usuario crea un ticket describiendo su problema. Soporte lo atiende, lo puede escalar y responder. El administrador puede asignarlo.
- Resultado esperado: se le da solución al ticket lo más pronto posible

8.1.5 Escenario 5: Gestión administrativa de usuarios

- Actor: Administrador
- Descripción: Desde el panel de administración, el administrador puede gestionar precios, paquetes y consultar
- Resultado esperado: Los cambios se reflejan en la base de datos y actualizan el aplicativo en tiempo real

8.1.6 Escenario 6: Cálculo y pago de comisiones

- Actor: Administrador
- Descripción: El sistema calcula las comisiones para los distribuidores según su categoría básico o premium (recordar que esto es un descuento)
- Resultado esperado: El administrador tiene listos los valores y la documentación de soporte.

8.1.7 Escenario 7: Consulta de perfil personal

- Actor: Usuario
- Descripción: El usuario inicia sesión, accede a su perfil y visualiza información relevante como nombre, correo, y acceso a sus métodos de pago o registrar uno nuevo
- Resultado esperado: El usuario tiene acceso a un panel completo sobre sus datos de forma clara

8.1.8 Escenario 8: Transferencia de dominio vía correo

- Actor: Usuario
- Descripción: Un usuario puede transferir un dominio en su propiedad al correo de otro usuario.
- Resultado esperado: La transferencia queda registrada y se notifica a los usuarios involucrados

8.2 Casos de Uso: Descripción y Modelo

Los casos de uso del sistema ChibchaWeb describen las funcionalidades clave que pueden ser ejecutadas por los distintos actores, permitiendo alcanzar objetivos específicos mediante la interacción con la aplicativo. Este modelo facilita la comprensión de los límites del sistema, las responsabilidades de cada actor y las relaciones entre funcionalidades.

Actores Del Sistema:

- Usuario: Actor general del que heredan todos los demás.
- Cliente: Usuario registrado que puede adquirir servicios y registrar dominios.
- Distribuidor: Usuario que gestiona múltiples dominios y cobra a sus propios clientes.
- Empleado: Usuario interno con funcione operativas.
- Soporte: Subtipo de empleado, encargado de atender tickets.
- Administrador: Usuario con permisos totales para gestionar perfiles, dominios y comisiones

8.2.1 Resumen De Casos De Uso

Categoría	Caso de Uso	Actor(es)
Autenticación	Iniciar Sesión	Usuario
	Registrarse como Cliente	Cliente
	Registrarse como Distribuidor	Distribuidor
	Verificar Cuenta por Token	Usuario
Gestión de Perfil	Ver Perfil	Usuario
	Buscar Cliente, Distribuidor o Empleado	Administrador
	Gestionar Perfiles de Cliente	Administrador
	Gestionar Perfiles de Distribuidor	Administrador
	Gestionar Perfiles de Empleado	Administrador
	Registrar Empleado	Administrador
Métodos de Pago	Agregar Método de Pago	Cliente
	Realizar Pago	Cliente
Servicios	Comprar Paquete	Cliente
	Cobrar a sus Clientes	Distribuidor
Dominios	Registrar Solicitud de Dominio	Usuario
	Transferir Dominio por Correo	Usuario
	Generar Archivo de Dominio	Administrador
Soporte Técnico	Registrar Ticket de Soporte	Usuario
	Consultar Tickets	Soporte
	Atender y Escalar Tickets	Soporte
	Responder Ticket	Soporte
	Asignar Ticket	Administrador
Comisiones	Calcular Comisiones a Distribuidores	Administrador

Cuadro 2: Resumen De Casos De Uso

9 Descripción y justificación de patrones de diseño adoptados y arquitectura de software definida.