

UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL SANTA FE

**PROYECTO FINAL: DESARROLLO DE UNA PLATAFORMA WEB PARA LA GESTIÓN
DEL TRABAJO DIARIO EN UN ESTUDIO CONTABLE - DIARY BOOSTER**

CHAMORRO, LUCAS – FORZANI DAMIAN

2020

**Documento de Arquitectura
de Software**

Tabla de contenidos

| | |
|---|----|
| Versionado | 2 |
| 1. Introducción | 3 |
| 1.1 Propósito | 3 |
| 1.2 Alcance | 3 |
| 1.3 Definiciones, Acrónimos y Abreviaturas | 3 |
| 1.4 Referencias | 3 |
| 2. Representación de la arquitectura | 4 |
| 3. Restricciones y aspectos claves de la arquitectura | 4 |
| 4. Atributos de calidad contemplados | 4 |
| 4.1 Escenarios de modificabilidad | 5 |
| Escenario 1 | 5 |
| Escenario 2 | 6 |
| 4.2 Escenarios de seguridad | 7 |
| Escenario 3 | 7 |
| Escenario 4 | 8 |
| 4.3 Escenarios de usabilidad | 9 |
| Escenario 5 | 9 |
| Escenario 6 | 10 |
| 4.4 Escenarios de accesibilidad | 11 |
| Escenario 7 | 11 |
| 5. Vistas de la arquitectura | 12 |
| 5.1 Vista Módulo | 12 |
| 5.2 Vista componente conector | 13 |
| 5.3 Vista instalación | 13 |
| 5.4 Vista de Despliegue | 14 |

Versionado

| Versión | Fecha | Responsable | Descripción del cambio |
|---------|------------|----------------------------------|--|
| 1.0 | 05/12/2018 | Lucas Chamorro Damian Forzani | Versión inicial del documento. Escenarios y Vista de Despliegue |
| 2.0 | 14/12/2018 | Lucas Chamorro Damian Forzani | Inclusión de las vistas: Módulo, Componente Conector, Instalación |
| 3.0 | 06/03/2019 | Lucas Chamorro Damian Forzani | Actualización de la vista de despliegue con la selección del proveedor. |
| 4.0 | 12/11/2019 | Lucas Chamorro Damian Forzani | Actualización de la vista de despliegue de acuerdo a la refactorización del módulo financiero. |

1. Introducción

DiaryBooster es un sistema para la gestión y seguimiento de las tareas y actividades realizadas en la diaria de un estudio contable, y el impacto de las mismas en todos los ámbitos del negocio. El desarrollo del sistema se originó en base a la solicitud del estudio contable Gaudiano, de la ciudad de Santa Fe, pero luego de ser evaluado funcionalmente, se decidió implementarlo de forma tal que pudiera ser posible su venta a otros estudios contables.

Este documento describirá la arquitectura del software del sistema DiaryBooster. La arquitectura no es otra cosa que la representación de la/s estructura/s que lo componen, sus elementos de software, sus interrelaciones, y las propiedades estructurales de cada uno de ellos.

1.1 Propósito

La intención de este documento es brindar una descripción comprensible acerca de la arquitectura de software del sistema web que llamaremos a lo largo del documento DiaryBooster.

Para mostrar la arquitectura a implementar, se utilizarán las diferentes vistas del modelo V&B, de acuerdo al aspecto del sistema que se desee representar. El principal objetivo de este documento es capturar y transmitir claramente las principales decisiones de arquitectura tomadas para el desarrollo de este sistema de software.

1.2 Alcance

Este documento corresponde al sistema DiaryBooster, un sistema web destinado a la gestión de las tareas diarias de un estudio contable, y particularmente, a la implementación realizada para el estudio contable Gaudiano, de la ciudad de Santa Fe, realizada por los alumnos Chamorro Lucas y Forzani Damián, como proyecto final de carrera para Ingeniería de Sistemas de Información, en la Universidad Tecnológica Nacional Facultad Regional Santa Fe.

1.3 Definiciones, Acrónimos y Abreviaturas

Ver el documento de Glosario.

1.4 Referencias

Glosario

Documento de Requerimientos de Software.

2. Representación de la arquitectura

La arquitectura de DiaryBooster será mostrada en este documento a partir de una serie de vistas, las cuales se corresponden con las establecidas por el modelo V&B. Los gráficos que se mostrarán no se corresponden con una representación formal UML, sino a las representaciones gráficas informales más extendidas para cada tipo de vista.

3. Restricciones y aspectos claves de la arquitectura

Existen ciertas restricciones y aspectos claves del sistema que tienen un impacto importante en el diseño de la arquitectura del sistema. Los mismos, se detallan a continuación:

- El estudio se encuentra ubicado en la ciudad de Santa Fe, pero fue un requisito del cliente, que el sistema pueda ser accedido tanto desde el estudio, como de cualquier otro lugar.
- Si bien la principal funcionalidad del sistema es hacer un seguimiento de las tareas y actividades que tienen que realizar los empleados del estudio, existen diferentes y variadas funcionalidades que el sistema debe implementar.
- Inicialmente, se estableció que el sistema solo podrá ser accedido por empleados del estudio. El acceso para diversas funcionalidades para clientes, está planteado para más adelante.
- El sistema no requiere, al menos en el alcance inicial, integración con otros sistemas.
- Seguridad: Es necesario que la información de los clientes sea resguardada correctamente, y solo pueda ser accedida por los empleados del estudio.

4. Atributos de calidad contemplados

En base a los requerimientos funcionales y no funcionales, y a las restricciones planteadas por el escenario detallado junto con el cliente, y con la proyección a futuro del uso y distribución del sistema, se estableció que los principales requerimientos no funcionales o atributos de calidad esperados de DiaryBooster son los siguientes:

1. Modificabilidad.
2. Seguridad.
3. Usabilidad.
4. Accesibilidad.

A su vez, para identificar tácticas arquitectónicas que resulten apropiadas y útiles para la solución requerida, se desarrollaron dos escenarios para cada atributo de calidad detallado (excepto para accesibilidad, para el cual solo se desarrolló un escenario).

4.1 Escenarios de modificabilidad

Escenario 1

Descripción del escenario concreto: “El usuario requiere introducir nuevos tipos de tareas al sistema (ya en producción), visibles en una semana porque se aproxima el vencimiento de AFIP de las mismas, y configurar la distribución de tareas considerando estas novedades.”.

| Atributo de calidad: Modificabilidad | |
|---|---|
| Parte del escenario | Valor seleccionado |
| Fuente | Usuario |
| Estímulo | Introducir nuevos tipos de tarea al sistema |
| Artefacto | Módulo de configuración, Módulo de gestión de tareas, Modelo y Base de datos. |
| Entorno | Sistema en producción |
| Respuesta | El sistema permite al usuario configurar desde su propia sesión en el sistema el nuevo tipo de tarea, y setear los nuevos parámetros de asignación correspondientes. |
| Medida de la respuesta | <ul style="list-style-type: none"> No implica trabajo para el equipo de desarrollo una vez el sistema está productivo. Nuevo Tipo de tarea cargado en el sistema luego de 5 minutos. |
| Tácticas propuestas | <ul style="list-style-type: none"> Transformar las tipificaciones del sistema que puedan variar en el tiempo en clases del modelo y tablas de BD, y brindar acceso a los ABM correspondientes al usuario. |
| Interrelaciones con otros atributos de calidad | <p>(+) Usabilidad: El usuario tendrá accesos a las modificaciones requeridas en la funcionalidad del sistema en un tiempo mucho menor.</p> <p>(+) Flexibilidad.</p> <p>(-) Performance: Este tipo de desarrollo implica mayor cantidad de consultas a la base de datos, por ejemplo, a la hora de mostrar un formulario que use dichas tipificaciones. El impacto debería ser mínimo gracias a la tecnología disponible y seleccionada (servidor web = servidor BD).</p> |
| Escenario 001: Modificabilidad | |

Escenario 2

Descripción del escenario concreto: “El estudio contable solicita una nueva funcionalidad de envío mails masivos a mediados de mes para clientes que registren un balance negativo en sus cuentas cliente. Tiempo de disponibilidad de la nueva funcionalidad solicitado 15 días, sin dar de baja el uso del sistema de lunes a viernes.”.

| Atributo de calidad: Modificabilidad | |
|---|--|
| Parte del escenario | Valor seleccionado |
| Fuente | Negocio |
| Estímulo | Adicionar una funcionalidad de envío masivo de mails de acuerdo al estado de cuenta de los clientes. |
| Artefacto | Módulo de clientes, Librerías del sistema |
| Entorno | Sistema en producción |
| Respuesta | Seleccionar e instalar (de ser necesario) una librería para el envío de emails, desarrollar la funcionalidad, y programar la tarea CRON. Despliegue de la funcionalidad un viernes a las 20:00 acordado con el cliente. |
| Medida de la respuesta | <ul style="list-style-type: none"> • 2 días de trabajo. • No afectar otras funcionalidades del sistema. |
| Tácticas propuestas | <ul style="list-style-type: none"> • Utilización de un framework de desarrollo que permita la fácil instalación y acoplamiento de nuevas librerías sin afectar las existentes. • Utilizar el patrón MVC, de forma tal que se separen correctamente la lógica del sistema, las vistas de usuario y el modelo. Así, solo se requerirá identificar las entidades del modelo involucradas, y desarrollar la nueva funcionalidad, aislando el impacto que podría tener en otras funciones del sistema. |
| Interrelaciones con otros atributos de calidad | <p>(+) Funcionalidad: este patrón de diseño permite incorporar rápidamente nuevas funcionalidades sin afectar las existentes.</p> <p>(+) Constructibilidad: desarrollar nuevas funcionalidades no requiere de una complejidad adicional a la intrínseca del requerimiento planteado. A la hora de una modificación, es fácil identificar los puntos de cambio en el código.</p> <p>(-) Complejidad: Las soluciones planteadas implican un sistema altamente modular, con diversidad de librerías, directorios, clases, etc., derivando en un sistema considerablemente más grande en cantidad de líneas de código y archivos.</p> |
| Escenario 002: Modificabilidad | |

4.2 Escenarios de seguridad

Escenario 3

Descripción del escenario concreto: “Un usuario empleado intenta acceder al estado de cuenta de usuario de un compañero de trabajo.”.

| Atributo de calidad: Seguridad | |
|---|---|
| Parte del escenario | Valor seleccionado |
| Fuente | Usuario |
| Estímulo | Intento de acceso a funcionalidades restringidas a otro tipo de usuario. |
| Artefacto | DiaryBooster |
| Entorno | Sistema en producción |
| Respuesta | El sistema verifica que el usuario que intenta acceder a esta funcionalidad no posee los permisos adecuados, y lanza un mensaje de error, indicando que no puede acceder a dicha información. |
| Medida de la respuesta | <ul style="list-style-type: none">• 99% de las veces el sistema responderá adecuadamente. |
| Tácticas propuestas | <ul style="list-style-type: none">• Autenticación de usuarios.• División de las funcionalidades del sistema en roles de usuario. Los roles responden a una jerarquía de roles. |
| Interrelaciones con otros atributos de calidad | (+) Modificabilidad: Facilidad para generar nuevos perfiles de usuario. (+) Correctitud: Solo los usuarios que deban acceder a cada porción de la información podrá hacerlo. |
| Escenario 003: Seguridad | |

Escenario 4

Descripción del escenario concreto: “Violación de la capa de seguridad por parte de un tercero, y acceso a la base de datos del sistema, la cual contiene información sensible respecto a los clientes, como ser CUIT, datos fiscales, claves, etc.”.

| Atributo de calidad: Seguridad | |
|---|---|
| Parte del escenario | Valor seleccionado |
| Fuente | DiaryBooster |
| Estímulo | Intento de acceso por parte del DBA a la base de datos de clientes, la cual contiene información sensible. |
| Artefacto | Base de datos |
| Entorno | Sistema en producción |
| Respuesta | El atacante accede a la base de datos, pero no puede hacer uso de la información sensible contenida en la misma. |
| Medida de la respuesta | <ul style="list-style-type: none">Imposibilidad de lectura de los registros de la base de datos al acceder a la misma por cualquier medio que no sea el sistema. |
| Tácticas propuestas | <ul style="list-style-type: none">Codificar los registros sensibles antes de introducirlos a la base de datos, y decodificarlos cuando se presenten al usuario por medio del sistema. |
| Interrelaciones con otros atributos de calidad | (-) Performance: cada registro que posea información sensible deberá tener un procesamiento extra antes de ser insertado en la base de datos. |
| Escenario 004: Seguridad | |

4.3 Escenarios de usabilidad

Escenario 5

Descripción del escenario concreto: “El sistema debe poder ser accedido desde diversidad de dispositivos para las funcionalidades básicas del sistema, y todo el sistema debe poder visualizarse correctamente en distintos formatos de pantallas de PC (desde 800x600 en adelante).”.

| Atributo de calidad: Usabilidad | |
|---|---|
| Parte del escenario | Valor seleccionado |
| Fuente | Usuario |
| Estímulo | Editar una reunión desde un dispositivo móvil. |
| Artefacto | Interfaz de usuario |
| Entorno | Sistema en producción |
| Respuesta | El formulario de la reunión se adapta correctamente al tamaño del dispositivo, permitiendo ingresar los datos correctamente, sin tener que desplazarse por la pantalla. |
| Medida de la respuesta | <ul style="list-style-type: none">• Desplazamiento horizontal nulo.• No hay necesidad de hacer zoom en la pantalla para visualizar correctamente el contenido del formulario. |
| Tácticas propuestas | <ul style="list-style-type: none">• Diseñar las pantallas de la vista aplicando prácticas de diseño web responsive, que adapta el contenido de acuerdo al tamaño de la pantalla que lo está visualizando.• Utilizar el patrón MVC, de forma tal que se separen correctamente la lógica del sistema, las vistas de usuario y el modelo. De esta forma, se puede adaptar las vistas sin necesidad de influir en los demás componentes. |
| Interrelaciones con otros atributos de calidad | <p>(+) Accesibilidad: Posibilidad de acceder al sistema desde diversos dispositivos.</p> <p>(+) Modificabilidad: El patrón MVC favorece la modificabilidad.</p> <p>(-) Complejidad: La solución tiene que ser adaptada para diversos dispositivos, aumentando la complejidad de diseño y construcción de las interfaces de usuario.</p> |
| Escenario 005: Usabilidad | |

Escenario 6

Descripción del escenario concreto: “El sistema debe mantener coherencia a lo largo de todas las entidades, presentando los listados de la misma forma, mismos botones para dar un alta y una baja o modificación.”.

| Atributo de calidad: Usabilidad | |
|---|--|
| Parte del escenario | Valor seleccionado |
| Fuente | Usuario |
| Estímulo | Estandarización en la forma de presentar los datos, y la ubicación de las acciones para todas las entidades. |
| Artefacto | Interfaz de usuario |
| Entorno | Sistema en desarrollo |
| Respuesta | El sistema presenta una interfaz uniforme, presentando los listados con el mismo formato, y las acciones en la misma ubicación. |
| Medida de la respuesta | <ul style="list-style-type: none"> Facilidad del usuario para familiarizarse con el funcionamiento del sistema. |
| Tácticas propuestas | <ul style="list-style-type: none"> Utilizar plantillas estandarizadas para la presentación de las interfaces de usuario, que se reutilicen en todas las pantallas del sistema (TWIG). Utilización de frameworks de diseño de interfaces web (Bootstrap por ejemplo). |
| Interrelaciones con otros atributos de calidad | (+) Modificabilidad: Si cambia el formato genérico, habría que realizar los cambios más grandes en la plantilla general, y luego solamente ajustar detalles en las específicas. |
| Escenario 006: Usabilidad | |

4.4 Escenarios de accesibilidad

Escenario 7

Descripción del escenario concreto: “Más allá de que la mayor parte del tiempo DiaryBooster se utilice desde el estudio contable, el sistema debe poder accederse desde cualquier dispositivo con conexión a internet y un navegador web.”.

| Atributo de calidad: Accesibilidad | |
|---|---|
| Parte del escenario | Valor seleccionado |
| Fuente | Usuario |
| Estímulo | Ingresa al sistema desde una notebook conectada a internet mediante un navegador web. |
| Artefacto | DiaryBooster |
| Entorno | Sistema en producción |
| Respuesta | El sistema presenta el formulario de logueo, y una vez el usuario se autentica, le brinda acceso al sistema. |
| Medida de la respuesta | <ul style="list-style-type: none">• 99% de disponibilidad de acceso mediante un dispositivo con internet. |
| Tácticas propuestas | <ul style="list-style-type: none">• Construir un sistema web que esté disponible a través de internet. |
| Interrelaciones con otros atributos de calidad | (-) Seguridad: Un sistema web tiene muchas más consideraciones de seguridad que un sistema de escritorio, o que un sistema web que no se encuentre expuesto en internet. |
| Escenario 007: Accesibilidad | |

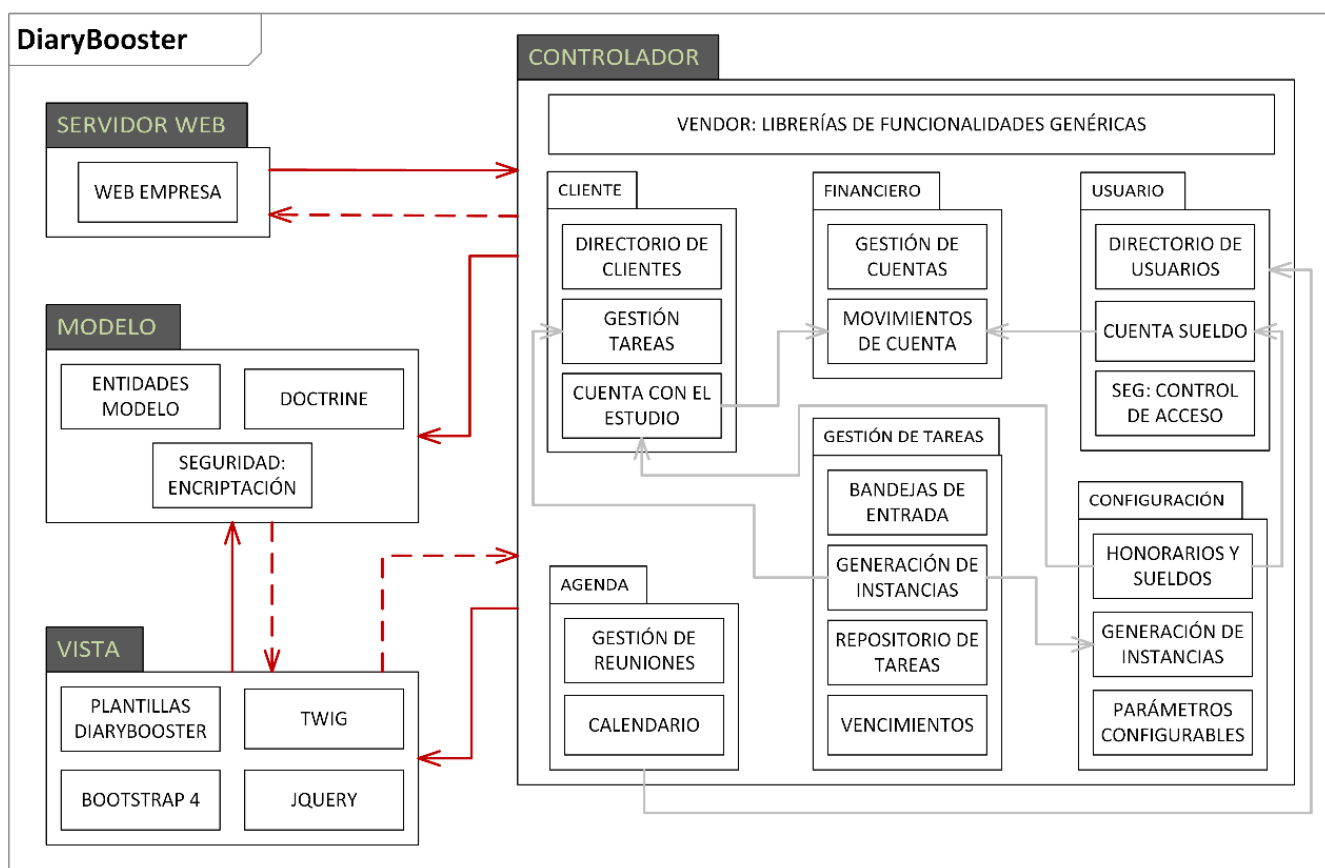
5. Vistas de la arquitectura

Para obtener una visión más detallada acerca de cómo se estructuran los diferentes componentes o elementos que forman DiaryBooster, y aquellos que no son parte, pero se relacionan con el sistema, y como se da esa relación, se presentarán un conjunto de vistas de arquitectura.

Cada una de estas vistas representará diferentes tipos de componentes, desde elementos de software, sistemas, subsistemas, librerías, hardware, conexiones, relaciones, etc. Las mismas se han escogido de acuerdo al modelo Views and Beyond, seleccionando las más relevantes para representar la arquitectura del sistema.

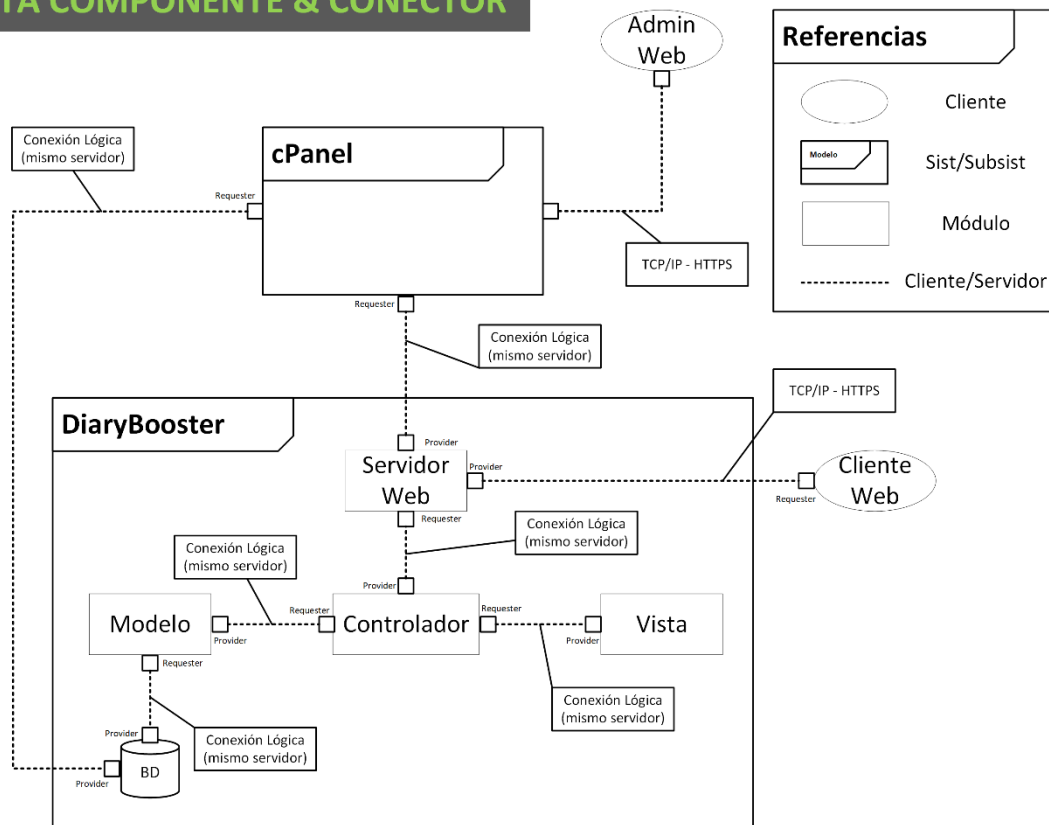
5.1 Vista Módulo

VISTA DE MÓDULO



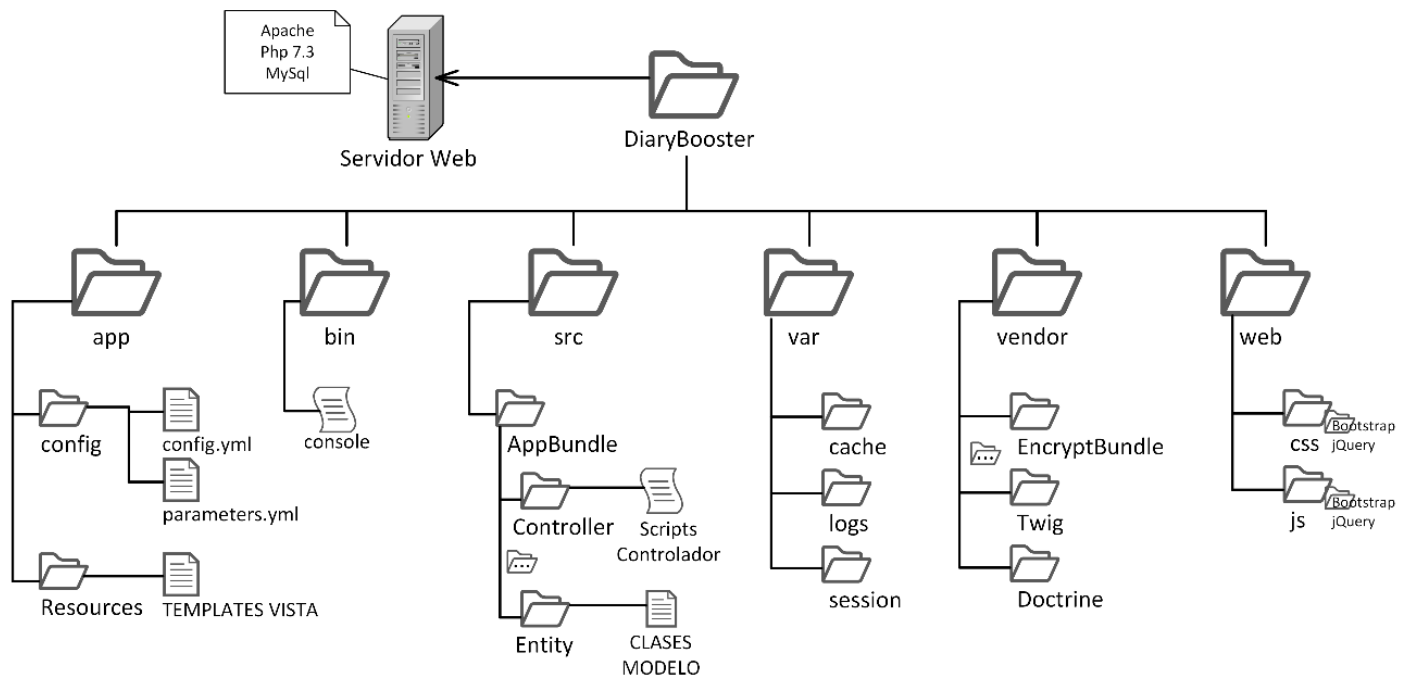
5.2 Vista componente conector

VISTA COMPONENTE & CONECTOR



5.3 Vista instalación

VISTA DE INSTALACIÓN



5.4 Vista de Despliegue

VISTA DE DESPLIEGUE

