**[Nombre del proyecto]**

**Versión X**

**Informe de arquitectura de Software**

**Recuerde eliminar la letra de color azul, es una simple referencia**

*En esta actividad del ciclo de vida del software es en la cual se analiza el modelamiento realizado en base a los requisitos, para producir una descripción de la estructura interna del software, que sirva de base para su construcción. La salida o resultado es un conjunto de artefactos que registran las principales decisiones adoptadas.*

*Este documento permite describir la arquitectura del software (cómo está compuesto y organizado por componentes), las interfaces y los componentes a un nivel de detalle que facilite con posterioridad su construcción.*

*Técnicamente la arquitectura es la parte central de la ingeniería del software. Durante éste se desarrollan, revisan y documentan los refinamientos progresivos de las estructuras de datos, de la estructura del programa y de los detalles procedimentales. Como resultado se tienen las representaciones, cuya calidad puede ser evaluada.*

* *El diseño arquitectónico (también referido como diseño de alto nivel, de nivel superior o preliminar), el cual describe cómo el software está organizado por componentes.*
* *El diseño detallado describe el comportamiento deseado de estos componentes (algoritmos, diagramas de flujo, casos de uso…).*
* *Durante el proceso y su documentación se debe dar respuesta a los siguientes interrogantes:*
* *¿Existe una arquitectura genérica que pueda ser usada?*
* *¿Cómo será́ distribuido el sistema?*
* *¿Qué estilos arquitectónicos son apropiados?*
* *¿Qué aproximación se utilizará para estructurar el sistema?*
* *¿Cómo se descompondrá́ el sistema en módulos?*
* *¿Qué estrategia de control se utilizará?*
* *¿Cómo se evaluará el diseño arquitectural resultante?*
* *¿Cómo se documentará la arquitectura?*

1. ***Estructura del software y arquitectura***

*Una arquitectura de software es "el conjunto de las estructuras necesarias para razonar acerca del sistema, comprenden elementos de software, las relaciones entre ellos y las propiedades de ambos", (Swebok v3.0, 2014).*

*Estilo arquitectónico por implementar:*

* *Los sistemas interactivos (Modelo-vista-controlador, presentación -abstracción - control)*

1. ***Principios de diseño de la interfaz de usuario que deberán ser tenidos en cuenta para la implementación requerida***

* *Facilidad de aprendizaje: el software debe ser fácil de aprender, de manera que el usuario pueda empezar a trabajar rápidamente.*
* *Familiaridad del usuario: la interfaz debe utilizar términos y conceptos extraídos de las experiencias de las personas que utilizarán el software.*
* *Coherencia: la interfaz debe ser coherente de modo que las operaciones comparables se activan de la misma manera.*
* *Mínima sorpresa: el comportamiento de software no debe sorprender a los usuarios.*
* *Recuperabilidad: la interfaz debe proporcionar mecanismos que permitan a los usuarios recuperar información de errores.*
* *Guía del usuario: La interfaz debe dar retroalimentación significativa cuando se producen errores y proporcionar ayuda relacionada con el contexto.*
* *La diversidad de usuario: la interfaz debe proporcionar mecanismos de interacción adecuados para diversos tipos de usuarios y con diferentes capacidades: ciegos, problemas de visión, sordo, daltónico, etc.*

1. ***Modalidades de interacción con el usuario, se deberá tener en cuenta al menos una de las siguientes modalidades:*** 
   * *Pregunta-respuesta: La interacción es esencialmente restringida a una sola pregunta- respuesta entre el usuario y el software. El usuario emite una pregunta y el software la responde.*
   * *Manipulación directa: Los usuarios interactúan con objetos en la pantalla. La manipulación a menudo incluye un señalador dispositivo tal como un ratón, TrackBall o un señalador en las pantallas táctiles.*
   * *Selección de menú́: El usuario elige un comando entre una lista de menú́ de comandos.*
   * *Formulario de relleno: El usuario diligencia los campos de un formulario. Para los campos que incluyen menús, debe haber botones que indiquen la acción que deber realizar el usuario.*
   * *Lenguaje de comandos: El usuario emite un comando y proporciona parámetros para dirigir el software.*
2. ***Presentación de la información, se deberán aplicar los siguientes principios a las interfaces de presentación de datos de los diferentes módulos del sistema.***

*Puede ser de naturaleza textual o gráfica, de acuerdo con el estilo de presentación de la información que el proyecto defina:*

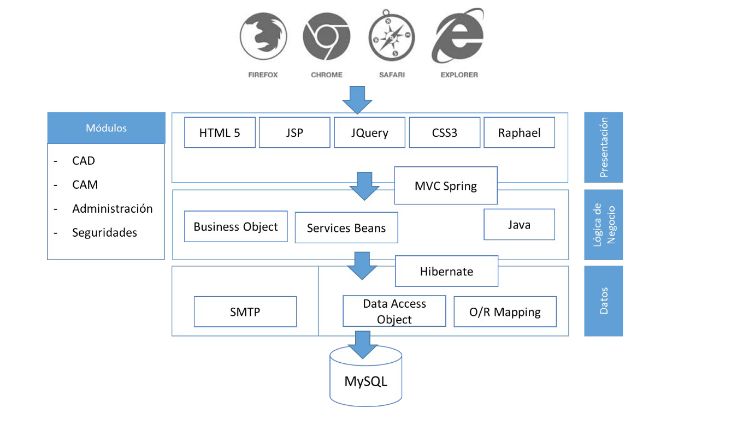
* + *Limite el numeró de colores utilizados*
  + *Utilice el cambio de color para mostrar el cambio de estado*
  + *Use códigos de colores para apoyar la tarea del usuario*
  + *Use códigos de colores en una manera reflexiva y coherente*
  + *Use colores para facilitar el acceso de las personas con daltonismo o deficiencia de color, por ejemplo, utilice el cambio de la saturación y el brillo del color, trate de evitar combinaciones de azules y rojos.*

**Arquitectura del sistema**

1. Arquitectura del desarrollo

Debe entregarse una arquitectura en un sistema diagramado o gráfico que permite entender el proceso del funcionamiento del software

Ejemplo:



1. Descripción de la arquitectura:

|  |  |  |
| --- | --- | --- |
| Proceso o Flujo | Descripción | Resultado |
|  |  |  |
|  |  |  |

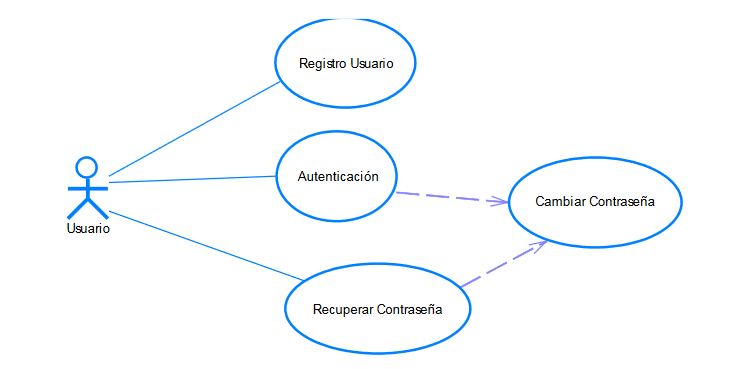
1. Variables de entrada, proceso y salida del sistema.

|  |  |  |
| --- | --- | --- |
| Entrada | Proceso | Salida |
|  |  |  |
|  |  |  |

1. Casos de Uso

En esta sección se muestran los Casos de Uso relevantes para la arquitectura, así como también a los principales Actores para cada uno de los procesos del software.

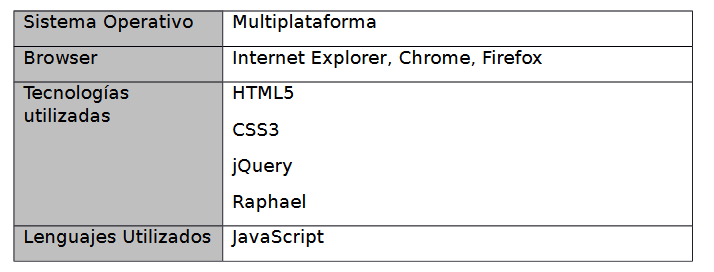
Ejemplo:



1. Arquitectura de infraestructura

En esta sección, debe especificar la infraestructura necesaria para que su software sea operativo.

Ejemplo:

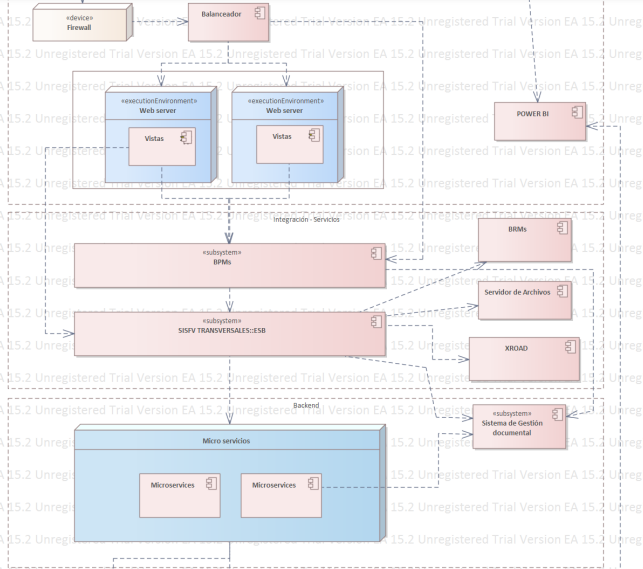


**Vista de implementación o componentes**

La vista de implementación se representa con un diagrama de capas con sus componentes respectivos y se muestra la interrelación de cada uno de estos. Los componentes definidos tienen su asignación de responsabilidades y cumplen lineamientos por cada uno de ellos. Para esta vista se usó el patrón modelo-vista-control MVC

La capa de presentación o FrontEnd, está conformada por un componente que está conformado por las vistas. En determinado caso funcional donde el usuario final utilice una funcionalidad de forma masiva, donde sean servicios utilizados directamente por el ciudadano, esta capa de presentación debe implementar el servicio y exponerlo o tener la capacidad de iniciar el proceso a través de una interface gráfica y pasar el control a las capas subyacentes.

Por ejemplo: A esta capa se ingresa a través de un browser y las peticiones deben pasar por un firewall. El balanceador permite que esta capa sea susceptible de crecer horizontalmente. Adicionalmente se comunica con el gestor de procesos, pasando la información suficiente para que este siga con el flujo funcional. También se puede comunicar con el bus de servicios e incluso podría ir directamente a los microservicios, aunque el lineamiento consiste en que todas las peticiones que van hacia la capa de negocio pasen por el bus de servicios.



1. Cobertura de código y pruebas unitarias

En este espacio el estudiante documentará las pruebas unitarias y tempranas que permitan dará cobertura a él código

Nota : Para cada caso de prueba se debe mostrar la siguiente información:

- Número del caso de prueba: Número secuencia que hace referencia a los casos de pruebas definidos.

- Componente al que hace referencia cada caso de prueba

- Prerrequisitos que se deben cumplir para realizar cada caso de prueba

- Descripción de cada una de las acciones a comprobar

- Los datos que se utilizarán de entrada

- La salida que se espera de ejecutar cada paso

- Las columnas sombreadas, correspondientes a 'Resultados' se rellenarán una vez ejecutadas las pruebas, obteniendo así el Informe de Resultado de Pruebas Unitarias

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Número del Caso de Prueba** | | **Componente** | **Descripción de lo que se Probará** | **Prerrequisitos** | |  | | |
| **<<CA0XX>>** | | <<Componente 1>> |  |  | |  | | |
| **<<CA0YY>>** | | <<Componente 2>> |  |  | |  | | |
|  |  |  |  |  |  | |  |
| **<<CAXX>>** | |  |  |  |  | |  |
| **Nº** | **Descripción** | **Método** | **Datos Entrada** | **Salida Esperada** | **¿OK?** | | **Observaciones** |
| **1** |  |  |  |  |  | |  |
| **2** |  |  |  |  |  | |  |
| **3** |  |  |  |  |  | |  |
| **4** |  |  |  |  |  | |  |
| **5** |  |  |  |  |  | |  |
| **6** |  |  |  |  |  | |  |
|  |  |  |  |  |  | |  |
| **<<CAYY>>** | |  |  |  |  | |  |
| **Nº** | **Descripción** | **Método** | **Datos Entrada** | **Salida Esperada** | **¿OK?** | | **Observaciones** |
| **1** |  |  |  |  |  | |  |
| **2** |  |  |  |  |  | |  |
| **3** |  |  |  |  |  | |  |
| **4** |  |  |  |  |  | |  |
| **5** |  |  |  |  |  | |  |
| **6** |  |  |  |  |  | |  |

Copie las tablas que necesite acorde a las pruebas que vaya a realizar.