**UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS**

**FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA**



**GESTIÓN DE REQUERIMIENTOS**

**Tool de Identificación de requerimientos de Software (TIRS)**

**DOCUMENTO**

**DOCUMENTO DE ARQUITECTURA**

**Versión <1.0>**

**HISTORIAL DE REVISIONES DEL DOCUMENTO**

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha** | **Versión** | **Descripción** | **Autor** |
| 17/07/2020 | 1.0 | Elaboración del documento de la arquitectura. | -Todos los integrantes del grupo |

Tabla de contenido

[Introducción](#_nmf14n) **4**

[Propósito](#_37m2jsg) 4

[Alcance](#_1mrcu09) 4

[Definiciones, acrónimos y abreviaturas](#_46r0co2) 5

[Definiciones](#_2lwamvv) 5

[Acrónimos](#_111kx3o) 5

[Referencias](#_3l18frh) 5

[Visión General del documento](#_206ipza) 6

[Representación de la arquitectura](#_4k668n3) **6**

[Metas y Restricciones de la arquitectura](#_2zbgiuw) **7**

[Metas](#_1egqt2p) 7

[Restricciones](#_3ygebqi) 7

[Vista de datos](#_39lg46dewqm) **8**

[Modelo Relacional](#_nhhtybbrftrc) 8

**Documento de Arquitectura de Software**

# **Introducción**

Uno de los desarrollos más importantes dentro de la construcción del software es el desarrollo de la arquitectura de software, que permite representar la estructura del sistema, sirviendo de comunicación entre las personas involucradas en el desarrollo y ayudando a realizar diversos análisis que orienten el proceso de toma de decisiones.

El presente documento nos da una vista de alto nivel de la Arquitectura de la Plataforma para el Tool de Identificación de Requerimientos de Software. Se muestran los objetivos, los principales procesos de negocio, el estilo arquitectónico que se aplicará en el Sistema y las distintas vistas basadas en el modelo vista controlador.

## **Propósito**

Este documento tiene el objetivo de proporcionar una visión general de la arquitectura global de Tool de Identificación de Requerimientos de Software con el fin que no sólo un personal o revisor técnico sea capaz de conocer y entender el presente proyecto, sino que cualquier colaborador relacionado al área de aplicación del proyecto tenga una idea general de éste.

## **Alcance**

En el presente documento abordaremos los casos de uso más importantes para el Tool de Identificación de Requerimientos de Software en base al modelo mvp ,además de mencionar los procesos de negocio del sector, modelo de datos e información adicional.

## **Definiciones, acrónimos y abreviaturas**

A continuación se brindan definiciones y acrónimos de términos usados en el presente documento que necesiten de alguna explicación para su correcta interpretación.

### **Definiciones**

|  |  |
| --- | --- |
| Actor | Entidad externa al sistema que guarda una relación con éste y que le demanda una funcionalidad. |
| Meta | Finalidad por la que se implementa la arquitectura del sistema. |
| Restricción | Especificaciones a las que está sujeta la arquitectura del sistema. |

### **Acrónimos**

|  |  |
| --- | --- |
| ERS | Especificación de Requisitos de Software |
| RUP | Rational Unified Process |
| CSRF | Cross site request forgery |
| TIRS | Tool de Identificación de Requerimientos de Software |

## **Referencias**

Las referencias aplicables son:

1. Documento de Especificación de Requerimientos funcionales y no funcionales de TIRS.
2. Documento de Especificación de Requisitos de TIRS

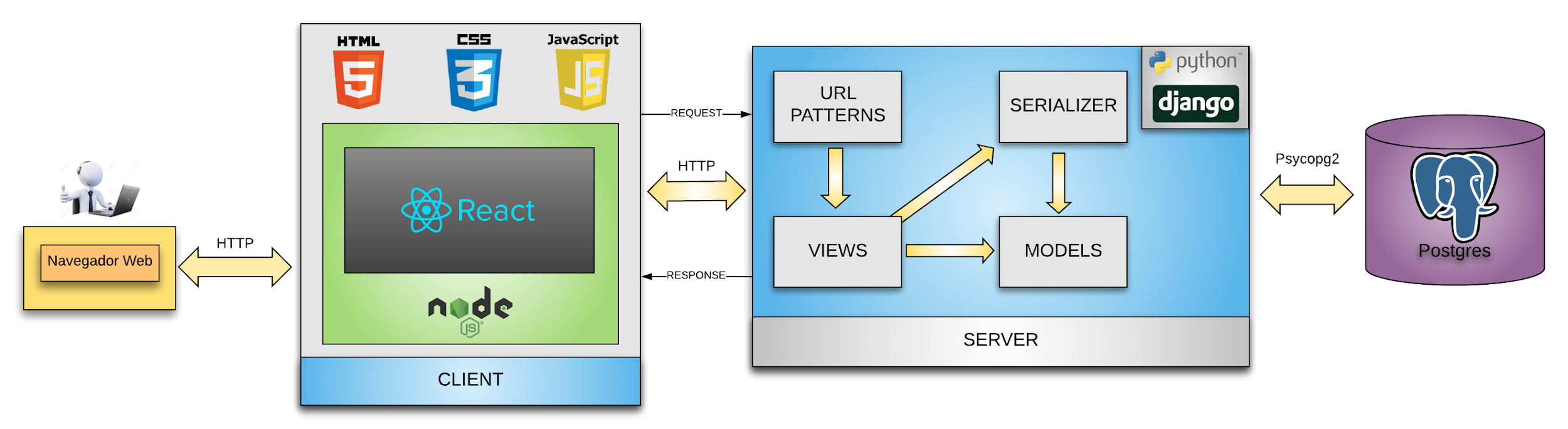
## **Visión General del documento**

El propósito de este documento es definir el marco referencial en el que se desarrollará el proyecto Tool de Identificación de Requerimientos de Software (TIRS). Se presenta la especificación del servicio que prestará el sistema. Para ello se presenta de manera clara la arquitectura del sistema, empleando un lenguaje sencillo y directo, así como gráficos y vistas de acuerdo a la metodología utilizada.

# **Representación de la arquitectura**

Para el desarrollo del proyecto se aplicará el tipo de arquitectura cliente-servidor. En este caso, el cliente de esta aplicación será implementado en la nube para que distintos colaboradores trabajen en conjunto.

Se emplea un solo servidor centralizado.



La arquitectura estará basada en un cliente-servidor que se comunican entre sí mediante peticiones HTTPs. Por la parte del cliente estaremos utilizando una aplicación React soportado en Node js; y en la parte del servidor estaremos utilizando Django que trabaja con una arquitectura interna de MVT que se conecta con la base de datos Postgresql.

# **Metas y Restricciones de la arquitectura**

Se tomarán en cuenta las siguientes metas y restricciones para el diseño de la arquitectura del sistema:

## **Metas**

La arquitectura del sistema de Tool de Identificación de Requerimientos de Software tiene las siguientes metas:

* Proporcionar un sistema web de fácil acceso mediante dispositivos que tengan conexión a internet.
* Proporcionar un sistema web seguro, libre de ataques maliciosos como SQL Injection, CSRF y/o Clickjacking mediante la implementación del Django.
* Brindar una aplicación flexible, escalable y fácil de mantener mediante el uso de componentes generados con React.
* Brindar una aplicación que permita reusar sus componentes..
* Obtener una estructura clara de todas las entidades relacionadas con el fin de poder realizar validaciones consistentes.

## **Restricciones**

La arquitectura del sistema de Tool de Identificación de Requerimientos de Software tiene las siguientes restricciones:

* El sistema está basado en una arquitectura web de cliente - servidor.
* Uso de Postgres como servidor de base de datos.
* Al ser un sistema web las funcionalidad se verán sólo mediante una conexión a internet a través de un navegador.
* Uso de HTML, CSS, JS, Node.js y la biblioteca React para implementar la sección del cliente.
* El cliente y el servidor se comunicarán mediante peticiones HTTP.
* Uso del Windows como sistema operativo de desarrollo.

## **Vista de datos**

## **Modelo Relacional**

