 Template

2018-12-20

**About arc42**

arc42, the Template for documentation of software and system architecture.

By Dr. Gernot Starke, Dr. Peter Hruschka and contributors.

Template Revision: 7.0 EN (based on asciidoc), January 2017

© We acknowledge that this document uses material from the arc 42 architecture template, <http://www.arc42.de>. Created by Dr. Peter Hruschka & Dr. Gernot Starke.

# Introduction and Goals

Neuron Software es una herramienta encargada de centralizar la información de los diferentes proyectos que trabajan los frentes de trabajo de la organización, presentandola en un front y bajo ciertos provilegios y condiciones dependiendo del grupo de quien la necesite.

**Objetivos del Negocio**

* Aumentar en un 100% el indice de confianza en las cifras presentadas sobre el estado de las implementaciones en cada de sus etapas y a nivel general.
* Generar informes por área cada 30 días,para cada uno de los jefes.
* Consultar la información de las implementaciones en tiempo real.
* Generar un informe general cada mes,para el gerente.
* Generar un informe cada 6 meses para el área de auditoria
* Implementar durante los próximos 3 meses las vistas para cada una de las áreas;una vista por mes.

**Requisitos Funcionales**

* -El Sistema permitirá registrar diseño y/o mantenimiento nuevo.
* -El Sistema permitirá modificar diseño y/o mantenimiento.
* -El Sistema permitirá consultar en tiempo real, información de diseño y/o mantenimientos.
* -El Sistema permitirá consultar en tiempo real información de implementaciones en etapa de desarrollo y el estado de estas.
* -El Sistema permitirá modificar información correspondiente a desarrollos de implementaciones y/o mantenimientos registrados.
* -El Sistema permitirá consultar información de las implementaciones y/o mantenimientos.
* -El Sistema permitirá consultar información de las implementaciones y/o mantenimientos de sus equipos de trabajo.
* -El Sistema permitirá exportar información a PDF o EXCEL.
* El Sistema permitirá administrar la información de todas las implementaciones.
* El Sistema permitirá Generar vistas por equipo.
* El Sistema permitirá Generar vista global de todas las etapas para cada implementación y/o mantenimiento.
* El Sistema permitirá Consultar información detallada de las etapas de las implementaciones.

## Requirements Overview

## Quality Goals

|  |  |  |
| --- | --- | --- |
| Prioridad | Objetivo de Calidad | Argumento |
| Seguridad | El sistema deberá restringir el acceso ante alguna operación invalida, mediante la fragmentación de permisos y roles |  |
| Usabilidad |  |  |
| Mantenibilidad |  |  |
| Portabilidad |  |  |
| Eficiencia |  |  |
| Interoperabilidad |  |  |
| Fiabilidad |  |  |
| Escalabilidad |  |  |

## Stakeholders

|  |  |  |
| --- | --- | --- |
| Role/Name | Contact | Expectations |
| Auditoría | Realizar un control de todos los componentes del Software | *<Expectation-1>* |
| Equipo de Soporte  Equipo de Diseño  Equipo de Desarrollo  Gerente  Administrador del Sitio  Jefes de cada equipo | Proporciona asistencia a los usuarios cuando se tiene un problema.  Diseña interfaces mas intuitivas y amigables para el usuario  Elabora e Implementa sistemas informáticos  Tiene la responsabilidad de administrar los elementos de ingresos y costos  Asegurar de que la información del sitio web es correcta.  Planificar,ejecutar y controlar un proyecto | *R* |

# Architecture Constraints

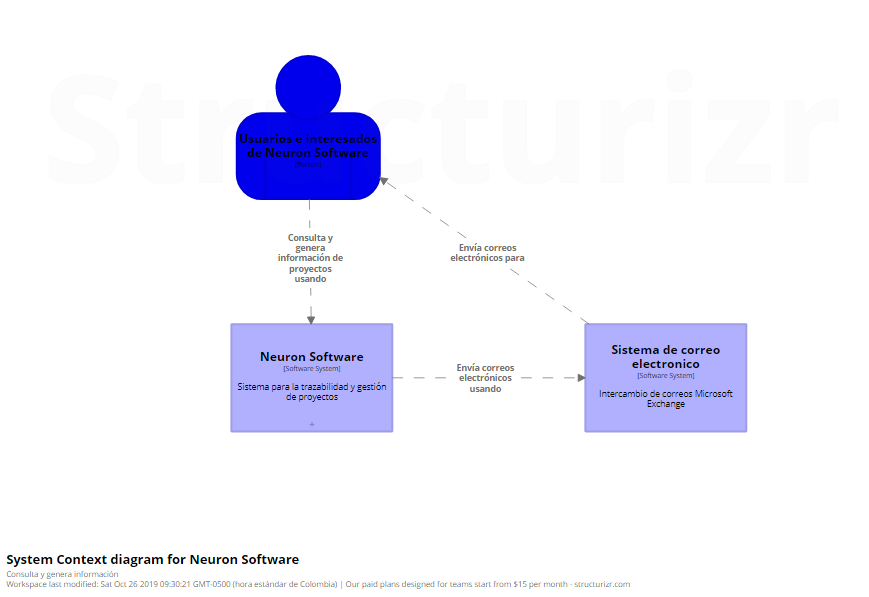
Neuron Software será:

* Independiente de la plataforma y debe ejecutarse en los principales sistemas operativos (Windows ,Linux y Mac-OS).

# System Scope and Context

Este numeral permite describir el entorno y el contexto en que el sistema va a trabajar, así mismo permitirá reconocer los componentes externos que interactuaran con el sistema.

## Business Context

****

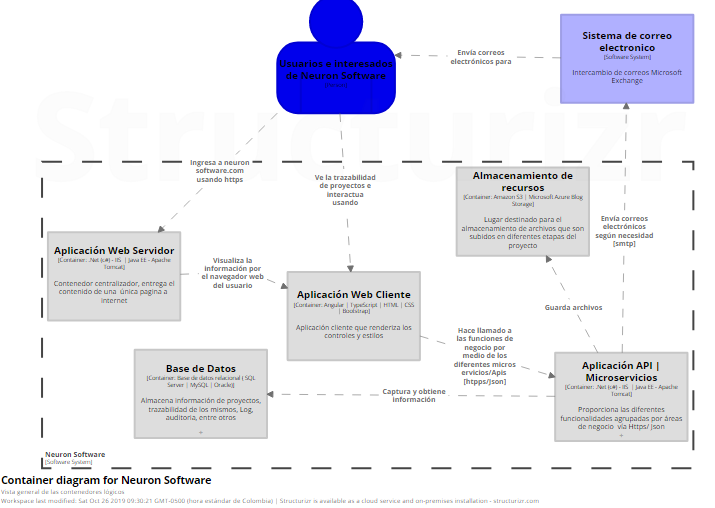
**Usuarios e interesados de Neuron Software:** los usuarios que interactuarán con el sistema se clasifican en 8 grupos, los cuales dependiendo de sus características podrán o no realizar ciertas actividades.

1. **Auditoria**: Verifican la correcta trazabilidad de la información, niveles de permisos según estándares corporativos y de entidades gubernamentales.   
   Gerente:
2. **jefe de área:** Son los encargados de planificar, gestionar y controlar las diferentes etapas del proyecto, así mismo como realizar algunas aprobaciones.
3. **Cliente:** Son los que tienen la necesidad o problema y requieren la automatización de algún proceso.
4. **Equipo de diseño:** Son las personas encargadas de tomar las necesidades, modelar y de realizar las parametrizaciones iniciales en las iniciativas y proyectos próximos a comenzar.
5. **Equipo de Implementación:** Es el equipo encargado de realizar las implementaciones y configuraciones correspondientes, partiendo de las parametrizaciones y modelado que se realice desde el diseño.
6. **Equipo de soporte:** Son las personas encargadas de proporcionar asistencia técnica cuando los usuarios presenten incidentes o algún tipo de novedad con la implementación.
7. **Gerente:** Responsable de administrar los costos del proyecto, realizar parametrizaciones generales y aprobaciones de cambio de data en las diferentes etapas.
8. **Administradores:** Son los encargados de realizar cualquier tipo de modificación ya sea a nivel de proyecto o de plataforma

**Neuron Software:** Proporciona todas las funcionalidades necesarias para satisfacer los requisitos de los diferentes usuarios e interesados como lo son parametrización de proyectos, consulta de proyectos, trazabilidad, guardar información por etapas, adjuntar archivos entre otras, que todas juntas ayudan a consolidar un gestor de proyectos.

**Sistema de correos electrónico:** Componente que proporciona todos los protocolos necesarios SMTP para el envío y recepción de correos.

## Technical Context



Neuron Software se divide en 5 contenedores o componentes principales

**Aplicación Web Servidor:** Representa el servidor de aplicaciones y todas sus características, Http, Https y sus demás protocolos donde correrá una o varias aplicaciones y APIs del sistema Neuron, de igual forma

**Aplicación Web Cliente:** Aplicación web que se ejecuta en el navegador de una maquina local, la cual renderiza todos los controles gráficos y estilos, en esta oportunidaduna aplicación web cliente centralizadora de todas las peticiones. Esta aplicación se realizará mediante el estilo SPA (Single Page Application).  
El spa se ejecuta en cualquier navegador web moderno y se comunica a través de http con el api

**Bases de Datos:** Este contenedor representa los servidores de base de datos que se utilizarán, los cuales pueden llegar a ser uno o varios, que a la vez tendrán toda la configuración y seguridad necesariapara un correcto funcionamiento. Se tendrán 5 bases de datos distribuidas en los diferentes servidores, apuntando cada una a su correspondiente api.

**Aplicación Api | Microservicios:** Este contenedor dedicado al funcionamiento de las Apis en su interior se dividen en 5 componentes, donde cada uno tiene una función y una responsabilidad especifica [ApiProyecto, ApiLoginPermisos, ApiEmail, ApiCliente, ApiAuditoria]. Estas Apis se ejecutan en servidores de aplicaciones compatibles y multiplataformas, utilizando contenedores integrados para una escalabilidad mucho más optima. Estas Apis expondrían métodos públicos que se consumirían desde un FrontEnd centralizador, algunas de estas Apis se comunicarán entre ellas, como es el caso del ApiAuditoria, que será instanciada desde las demás.

**Almacenamiento de Recursos:**Sera un contenedor de almacenamientode archivos, donde la aplicación podrá guardar físicamente todos los adjuntos cargados por los usuarios, contando con una segregación por proyectos, los nombres de los archivos serán guardados en una tabla junto con el nombre de la subcarpeta del proyecto, para armar lógicamente la ruta de cada archivo.  
la manipulación del almacenamiento será por medio del Api expuesto por el proveedor de nube para este servicio

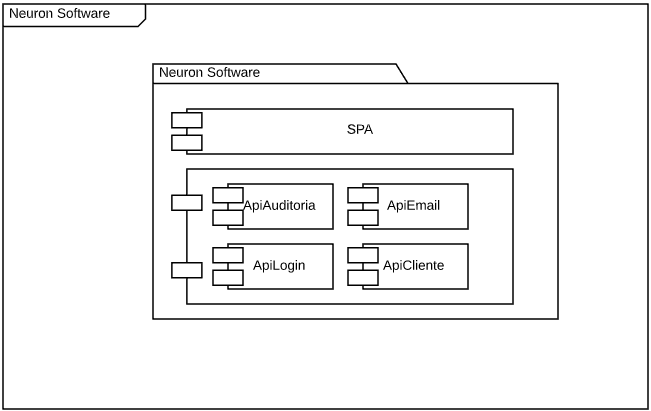
Algunas de las interfaces principales y el canal de comunicación las listamos a continuación

|  |  |
| --- | --- |
| Interfaces de negocio | Canal |
| Login | ApiLoginPermisos (Http) |
| Registro de clientes | ApiClientes(Http) |
| Registro de Proyecto | ApiProyecto(Http) |
| Registrar Diseño | ApiProyecto(Http) |
| Registrar Implementación | ApiProyecto(Http) |
| Envió de correos | ApiEmail(Http) |
| Guardar Errores | ApiAuditoria(Http) |
|  |  |

# Solution Strategy

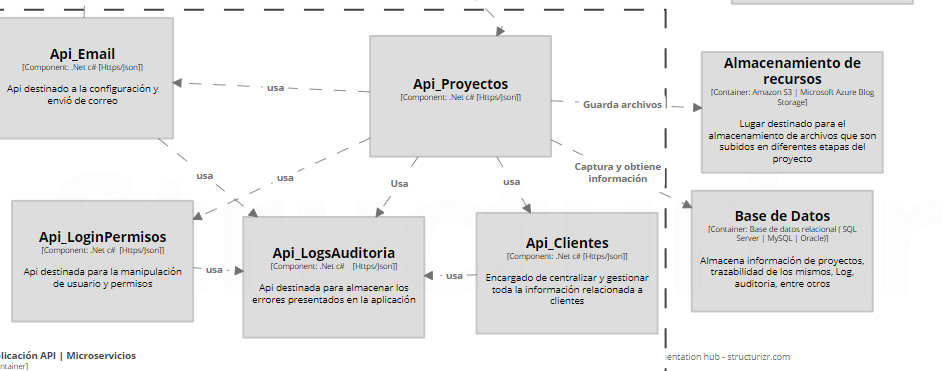
# Building Block View

La aplicación Neuron Software tiene varios empaquetados, que contienen sus principales componentes



De estos elementos se hará mayor énfasis a los componentes del Api detallando un poco sus elementos internos

## Whitebox Overall System



Partiendo del diagrama de contenedores o de contexto técnico, se realiza un zoom al contenedor de Apis, donde en su interior vemos cinco Apis totalmente autónomas e independientes que tienen diferentes responsabilidades. Muchas Apis se comunican entre ellas o requieren compartir objetos, para no tener una dependencia entre las Apis y tener un bajo acoplamiento, la comunicación entre ellas se realizará por medio de eventos que disparará la acción en el respectivo.

### White Box *Neuron Software: ApiProyecto*

El siguiente diagrama muestra los principales bloques de construcción del ApiProyecto y como estos se relacionan.



Los seis componentes principales que conforman el ApiProyecto tienen responsabilidades diferentes que se encapsulan dentro del microservicio

|  |  |
| --- | --- |
| Componente | Descripción |
| Servicio | Este componente interno sera el encargado de centralizar todas las funciones y reglas de negocio que se utilicen en el Api, adicionalmente tendra segregación por opción. |
| IServicio | Componente interno encargado de exponer las firma o contratos de los metodos y funciones realizados en la capa servicios a las otras capas |
| Repositorio | Capa encargada de la manipulación y persistencia de datos |
| IRepositorio | Componente interno encargado de exponer las firma o contratos de los metodos y funciones realizados en la capa DAO – Repositorio |
| Registrar Implementación | ApiProyecto(Http) |
| Envió de correos | ApiEmail(Http) |
| Guardar Errores | ApiAuditoria(Http) |
|  |  |

*<Purpose/Responsibility>*

*<Interface(s)>*

*<(Optional) Quality/Performance Characteristics>*

*<(Optional) Directory/File Location>*

*<(Optional) Fulfilled Requirements>*

*<(optional) Open Issues/Problems/Risks>*

### <Name black box 2>

*<black box template>*

### <Name black box n>

*<black box template>*

### <Name interface 1>

…

### <Name interface m>

## Level 2

### White Box *<building block 1>*

*<white box template>*

### White Box *<building block 2>*

*<white box template>*

…

### White Box *<building block m>*

*<white box template>*

## Level 3

### White Box <\_building block x.1\_>

*<white box template>*

### White Box <\_building block x.2\_>

*<white box template>*

### White Box <\_building block y.1\_>

*<white box template>*

# Runtime View

## <Runtime Scenario 1>

* *<insert runtime diagram or textual description of the scenario>*
* *<insert description of the notable aspects of the interactions between the building block instances depicted in this diagram.>*

## <Runtime Scenario 2>

## …

## <Runtime Scenario n>

# Deployment View

## Infrastructure Level 1

***<Overview Diagram>***

Motivation

*<explanation in text form>*

Quality and/or Performance Features

*<explanation in text form>*

Mapping of Building Blocks to Infrastructure

*<description of the mapping>*

## Infrastructure Level 2

### *<Infrastructure Element 1>*

*<diagram + explanation>*

### *<Infrastructure Element 2>*

*<diagram + explanation>*

…

### *<Infrastructure Element n>*

*<diagram + explanation>*

# Cross-cutting Concepts

## *<Concept 1>*

*<explanation>*

## *<Concept 2>*

*<explanation>*

…

## *<Concept n>*

*<explanation>*

# Design Decisions

# Quality Requirements

## Quality Tree

## Quality Scenarios

# Risks and Technical Debts

# Glossary

|  |  |
| --- | --- |
| Term | Definition |
| <Term-1> | <definition-1> |
| <Term-2> | <definition-2> |