*vicerrectoría Académica Dirección de ServicioS AcadémicoS Subdirección de ServicioS a EScuelaS*

TaskBlitz

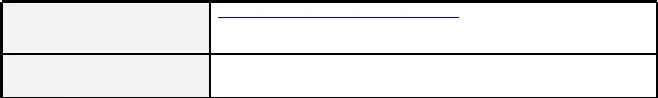
(DAS) Documento Arquitectónico del Sistema

Versión 1.0

Identificación de Documento

Jeferson Jaque

Diego Estay



Documento aprobado por

Jeferson Jaque

Fecha de última aprobación



Identificación

Proyecto Versión

DAS

TaskBlitz

1.0



Tabla de Contenidos





Contenido

[Introducción 3](#_Toc183554941)

[1 Alcance 3](#_Toc183554942)

[2 Arquitectura de Software 3](#_Toc183554943)

[3 Objetivos y Restricciones de la Arquitectura 4](#_Toc183554944)

[Objetivos de la arquitectura 4](#_Toc183554945)

[Restricciones de la Arquitectura 4](#_Toc183554946)

[4 Arquitectura Lógica 5](#_Toc183554947)

[Parte Estructural 5](#_Toc183554948)

[Parte Dinámica 6](#_Toc183554949)

[Diagrama Base De Datos 7](#_Toc183554950)

[5 Arquitectura de Procesos 8](#_Toc183554951)

[6 Arquitectura de desarrollo 11](#_Toc183554952)

[7 Arquitectura física 12](#_Toc183554953)

[8 Escenarios 13](#_Toc183554954)

[Especificación de Casos de Uso Relevantes 13](#_Toc183554955)

[Especificación de los Escenarios de Calidad Relevantes 14](#_Toc183554956)

[9 Tamaño y desempeño 15](#_Toc183554957)

# Introducción

El propósito del software TaskBlitz es proporcionar a las empresas una plataforma avanzada de gestión de proyectos que permita optimizar la asignación de tareas, mejorar la colaboración entre equipos y ofrecer una visión integral del progreso en tiempo real.

TaskBlitz está diseñado para abordar problemas comunes en la gestión empresarial, como la sobrecarga de trabajo, asignación ineficaz de recursos y falta de priorización clara. Además, el sistema fomenta la transparencia organizacional y la comunicación fluida, ofreciendo un entorno laboral más eficiente y equilibrado.

# Alcance

El software TaskBlitz está dirigido a pequeñas, medianas y grandes empresas que necesitan optimizar sus procesos de gestión de proyectos. Las principales funcionalidades incluyen:

Asignación Inteligente de Tareas: Utilización de algoritmos que consideran la carga laboral y habilidades del equipo.

Colaboración en Tiempo Real: Integración de chats y documentos compartidos para mejorar la comunicación.

Priorización de Proyectos: Sistema dinámico que ajusta prioridades según urgencia e impacto.

Reportes Visuales Detallados: Generación de gráficos y estadísticas en tiempo real sobre el avance de proyectos.

El sistema será accesible desde múltiples dispositivos (PC, laptops, smartphones y tablets) y soportará sistemas operativos modernos como Windows , iOS y Android.

# Arquitectura de Software

La arquitectura de TaskBlitz sigue el enfoque del framework 4+1, permitiendo una visión integral desde diferentes perspectivas arquitectónicas. Las vistas incluidas son:

1. **Vista de Casos de Uso y Escenarios de Calidad:**

Presenta los actores principales y los casos de uso más relevantes.

Detalla los escenarios de calidad, como desempeño, seguridad y escalabilidad.

1. **Vista de Metas y Restricciones:**

Define las restricciones tecnológicas y normativas que influyen en las decisiones arquitectónicas, como soporte multiplataforma y tiempo de desarrollo limitado.

1. **Vista Lógica:**

Divide el sistema en módulos principales, como Gestión de Tareas, Colaboración y Reportes.

Usa diagramas para representar la estructura lógica y las interacciones entre componentes.

1. **Vista de Procesos:**

Describe los flujos de trabajo clave, como la asignación de tareas y la generación de reportes.

Incluye diagramas de procesos y sincronización.

1. **Vista de Implementación:**

Muestra la infraestructura de despliegue, con servidores en la nube y aplicaciones cliente basadas en navegadores y móviles.

# Objetivos y Restricciones de la Arquitectura

## Objetivos de la arquitectura

De acuerdo a las reuniones y al análisis de los requerimientos, se listan los principales conductores iniciales de la arquitectura los cuales corresponden a las metas arquitectónicas iniciales:

1. Desempeño: Garantizar tiempos de respuesta rápidos para funcionalidades clave, como asignación de tareas y generación de reportes.
2. Seguridad: Proteger los datos de los usuarios mediante autenticación y cifrado avanzado.
3. Operatividad: Asegurar que el sistema funcione correctamente en cualquier sistema operativo o dispositivo moderno

## Restricciones de la Arquitectura

Existen restricciones que han sido levantadas con los stakeholders, las cuales se presentan a continuación:

1. Tiempo de Desarrollo: El proyecto debe completarse en un máximo de tres meses.
2. Infraestructura: Compatibilidad asegurada con Windows, iOS y Android.
3. Presupuesto: No se permite la adquisición de software adicional con licencias externas.



# Arquitectura Lógica

A continuación, se presenta una vista lógica de la aplicación TaskBlitz, representada en dos diagramas: uno que describe la parte estructural (estática y dinámica) y otro que representa la parte dinámica (interacción de componentes y conectores).

## Parte Estructural

En el siguiente diagrama de clases, el módulo principal es Gestión de Proyectos, que conecta e integra las funcionalidades clave de TaskBlitz. Este módulo incluye:

1. **Asignación de Tareas:**

* Módulo que optimiza la distribución de tareas basándose en la carga laboral, habilidades del equipo y disponibilidad.
* Proporciona alertas en caso de sobrecarga.

1. **Colaboración:**

* Chats en tiempo real y opciones para compartir documentos.
* Gestión de actualizaciones en tareas de forma instantánea.

1. **Seguimiento y Reportes:**

* Generación de reportes automáticos en formatos como PDF o Excel.
* Visualización de gráficos que muestran el progreso por tarea y equipo.

1. **Panel de Priorización:**

* Módulo que permite ajustar las prioridades de tareas y proyectos de manera dinámica.
* Evalúa la urgencia e impacto para ordenar los elementos del backlog.

1. **Autenticación y Seguridad:**

* Integración con Django Authentication para garantizar el acceso seguro mediante roles definidos

1. **Accesibilidad:**

* Opciones para personalizar la interfaz y facilitar su uso, ajustándose a necesidades específicas de los usuarios.

Cada usuario debe autenticarse para acceder a estas funcionalidades, asegurando la protección de datos y roles específicos en la plataforma. TaskBlitz también incluye un módulo de accesibilidad que permite a los usuarios personalizar la interfaz para mejorar su experiencia.

## Parte Dinámica

El sistema **TaskBlitz** sigue una arquitectura cliente-servidor, en la que **Angular** maneja la interfaz de usuario y **Django** gestiona la lógica de negocio y las operaciones del servidor.

1. **Frontend**:

* Desarrollado en Angular, proporciona una interfaz dinámica y responsiva.
* Comunica las acciones del usuario al backend mediante servicios HTTP.
* Utiliza Angular Material para una experiencia de usuario optimizada y consistente.

1. **Backend**:
   * Gestiona la lógica de negocio, realiza validaciones y responde a las solicitudes del frontend a través de una API RESTful construida con Django REST Framework (DRF).
   * Ofrece endpoints seguros para operaciones como creación de tareas, generación de reportes y administración de usuarios.
2. **Base de Datos**:
   * Utiliza SQLite como sistema de gestión de bases de datos relacional.
   * Almacena información estructurada de usuarios, tareas, reportes y configuraciones del sistema.
3. **Comunicación API RESTful:**
   * Los servicios del backend se exponen como endpoints RESTful.
   * Angular consume estos endpoints para cargar y enviar datos.
4. **Autenticación y Seguridad**:

* Se implementa autenticación con JWT (JSON Web Tokens) para sesiones seguras.
* Incluye cifrado de contraseñas

## Diagrama Base De Datos

Diagrama

Descripción generada automáticamente



# Arquitectura de Procesos

La vista de procesos se enfoca en describir cómo se ejecutan y sincronizan las funciones dentro del sistema, asegurando que los flujos operativos sigan los principios del patrón de diseño MVC (Modelo-Vista-Controlador). Este análisis es crucial para garantizar que todos los requisitos funcionales sean abordados correctamente, considerando las limitaciones proporcionadas por los clientes y las dependencias entre los diferentes espacios del sistema.

* 1. Diagrama Actividad- Agregar tareas a los Usuarios

Diagrama

Descripción generada automáticamente



* 1. Diagrama de Actividad - Priorización de Proyectos

Diagrama

Descripción generada automáticamente



* 1. Diagrama de Actividad - Colaboración en Tiempo Real.

Diagrama, Esquemático

Descripción generada automáticamente

# Arquitectura de desarrollo

El diagrama presentado corresponde a una representación del patrón Modelo-Vista-Controlador (MVC) implementado en la aplicación. Esta arquitectura separa de forma clara la lógica de negocio, la gestión de datos y la presentación al usuario, lo que garantiza modularidad, escalabilidad y facilidad de mantenimiento.

Captura de pantalla de un celular

Descripción generada automáticamenteIlustración 4: Vista de Implementación -> Diagramas de componentes

# Arquitectura física

En esta vista se identifican los nodos que componen la arquitectura de despliegue de TaskBlitz, destacando aquellos que interactúan directamente con el flujo principal de la aplicación. Estos nodos son fundamentales para asegurar el rendimiento, la seguridad y la escalabilidad del sistema. A continuación, se describen los nodos clave que permiten la ejecución eficiente de las funcionalidades del sistema

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza mediaIlustración 6: Diagrama de Despliegue 

# Escenarios

Esta sección describe en detalle el conjunto de escenarios funcionales y no funcionales que obtuvieron la mayor prioridad en el software. Para esto se presenta el diagrama de casos de uso y los casos de uso prioritarios, así como los escenarios en que uno o más atributos de calidad se ven involucrados de manera significativa

## Especificación de Casos de Uso Relevantes

Los casos de uso más relevantes para el desarrollo de la arquitectura de TaskBlitz fueron seleccionados considerando los siguientes criterios:

* + **Implementación Multinodal**: Requieren la interacción de múltiples nodos de la vista de despliegue, incluyendo el frontend, backend y la base de datos.
  + **Alto Riesgo**: Implican funcionalidades críticas o complejas, cuya falla podría afectar la estabilidad del sistema.
  + **Complejidad Conceptual**: Integran múltiples conceptos y relaciones clave dentro del dominio de la gestión de proyectos.
  + **Escenarios Críticos de Calidad:** Involucran atributos esenciales como desempeño, seguridad y escalabilidad, fundamentales para el éxito del sistema.

## Especificación de los Escenarios de Calidad Relevantes

Después de un análisis en conjunto con los stakeholders, los escenarios de calidad se expresan a continuación:

ID: QS1

**Nombre**: Gestión Inteligente de Tareas

**Sinopsis**: Permitir a los gerentes de proyecto optimizar la asignación de tareas basándose en carga laboral.

**Entorno**: Operación normal del sistema con múltiples usuarios asignando tareas.

**Cambio en el entorn**o: Incremento en la cantidad de tareas por asignar o redistribuir.

Comportamiento esperado: El sistema asigna automáticamente tareas según parámetros definidos y emite alertas si detecta sobrecarga en un miembro del equipo.

**Medida**: La asignación automática debe completarse en menos de 5 segundos por tarea.

**Prioridad** **Arquitectónica**: Alta.

**Aplicación**: Global.

ID: QS2

**Nombre**: Priorización de Proyectos

**Sinopsis**: Habilitar herramientas que permitan organizar y ajustar prioridades en proyectos dinámicamente.

**Entorno**: Operación regular con múltiples proyectos activos.

**Cambio en el entorno:** Cambio en los criterios de urgencia o impacto de los proyectos.

**Comportamiento** **esperado**: El sistema debe permitir la actualización de prioridades en tiempo real y reflejar los cambios en el tablero de tareas.

**Medida**: La actualización de prioridades debe reflejarse en el tablero en menos de 2 segundos.

**Prioridad** **Arquitectónica**: Alta.

**Aplicación**: Global.

ID: QS3

**Nombre**: Colaboración en Tiempo Real

**Sinopsis**: Proveer herramientas de comunicación en tiempo real, como chat

**Entorno**: Interacción constante entre equipos para coordinar tareas y proyectos.

**Cambio en el entorno**: Incremento en la cantidad de mensajes

**Comportamiento esperado**: El sistema debe garantizar la entrega instantánea de mensajes

**Medida**: Los mensajes y documentos deben ser procesados en menos de 1 segundo.

**Prioridad** **Arquitectónica**: Alta.

**Aplicación**: Local y Global.



# Tamaño y desempeño

Las decisiones arquitectónicas del proyecto se tomaron considerando las restricciones y prioridades definidas por el alcance y los objetivos del sistema. En particular, se dio un peso significativo a la restricción de Tiempo de Construcción, ya que el sistema debe implementarse en un plazo ajustado. Esto nos llevó a adoptar una arquitectura conocida, basada en tecnologías consolidadas como Django y Angular, minimizando el riesgo asociado a su implementación y garantizando una curva de aprendizaje más rápida para el equipo.

**Paralelización y Modularidad**

Uno de los objetivos principales fue asegurar la separación de responsabilidades en los componentes, lo que permite su construcción de forma paralela por diferentes miembros del equipo. Este enfoque también facilita la implementación de pruebas unitarias sobre cada módulo, utilizando herramientas como Pytest para el backend en Django y frameworks de pruebas específicas en Angular para el frontend, asegurando una baja tasa de fallos y confiabilidad en el sistema.

**Restricciones de Infraestructura**

La arquitectura también se diseñó teniendo en cuenta las restricciones de infraestructura y el escenario de calidad relacionado con la Tolerancia a Fallos. Por ello, el sistema se conceptualizó como una solución híbrida:

* Una aplicación web activa-activa, que asegura que los usuarios puedan interactuar continuamente con la plataforma.
* Un backend modular y robusto, que permite la disponibilidad de servicios en un esquema activa-pasiva, donde los procesos críticos pueden ser escalados o redundados en caso de necesidad.

