

**Anteproyecto**  
FACULTAD DE INGENIERÍA Y CIENCIAS BÁSICAS  
**Institución Universitaria Politécnico Grancolombiano**

Versión formato:  
2019-02-20

**Aplicación web para el registro y control de tiempos de los empleados en una empresa de desarrollo de software**

**Richard Daniel Hernández Arias**

**Institución Universitaria Politécnico Grancolombiano**

**Facultad de Ingeniería, Diseño e Innovación**

**Ingeniería de Sistemas**

**Bogotá**

**2024**

**Anteproyecto**  
FACULTAD DE INGENIERÍA Y CIENCIAS BÁSICAS  
**Institución Universitaria Politécnico Gracolombiano**

Versión formato:  
2019-02-20

## Contenido

1.	DATOS GENERALES - PRÁCTICA EMPRESARIAL .....	4
2.	INTRODUCCIÓN .....	5
2.1.	DESCRIPCIÓN DE LA EMPRESA.....	5
2.2.	FORMULACIÓN DEL PROBLEMA .....	5
3.	PROPUESTA DE PRÁCTICA .....	6
3.1.	JUSTIFICACIÓN .....	6
3.2.	ALCANCE .....	6
3.3.	OBJETIVO PRINCIPAL.....	7
3.4.	OBJETIVOS SECUNDARIOS .....	7
4.	REVISIÓN DE LITERATURA .....	7
5.	METODOLOGÍA.....	10
5.1.	METODOLOGÍA AGÍL.....	10
5.2.	DISEÑO y CONSTRUCCIÓN DEL SISTEMA.....	11
5.2.1	HISTORIAS DE USUARIO.....	13
5.2.2	LISTADO DE REQUERIMIENTOS Y FLUJO.....	16
5.2.3	BASE DE DATOS.....	20
5.3.	CRONOGRAMA DE ACTIVIDADES.....	23
5.4.	TECNOLOGÍAS .....	25
5.5.	ENTREGABLES .....	27
6.	RESULTADOS.....	27
7.	DISCUSIÓN Y CONCLUSIONES.....	31
	BIBLIOGRAFÍA.....	32
	ANEXOS .....	33

**Anteproyecto**  
**FACULTAD DE INGENIERÍA Y CIENCIAS BÁSICAS**  
**Institución Universitaria Politécnico Gran Colombiano**

Versión formato:  
2019-02-20

## 1. DATOS GENERALES - PRÁCTICA EMPRESARIAL

Facultad:	Facultad de Ingeniería, Diseño e Innovación
Departamento Académico:	Ingeniería de sistemas – Escuela TIC
Nombre completo practicante:	Richard Daniel Hernández Arias
Código estudiante:	100317108
Cédula No.:	1.002.468.594
Correo electrónico:	rdanhernandez@poligran.edu.co
Cargo asignado:	Analista Desarrollador Java
Empresa:	Nexos Software
Fecha inicio:	07/02/2024
Fecha finalización:	03/06/2024
Jefe inmediato:	Caterin Vazquez
Cargo jefe inmediato:	Gerente de proyectos
Asesor práctica empresarial:	José Manuel Chauta Torres
Título del anteproyecto:	Sistema control de tiempos
Área de conocimiento:	Software.
Palabras clave:	Control de tiempos, desarrollo web, disponibilidad de datos, gestión administrativa, metodología ágil.

## 2. INTRODUCCIÓN

### 2.1. DESCRIPCIÓN DE LA EMPRESA

Nexos Software es una organización fundada en la ciudad de Medellín con más de 23 años enfocada en el desarrollo de soluciones tecnológicas. Ofrece un catálogo amplio de servicios que buscan potenciar continuamente necesidades de negocio de clientes y aliados, velando por la calidad, transformación digital, innovación, el agilismo y la entrega oportuna.

La empresa tiene como visión consolidarse nacional e internacionalmente en la gestión de proyectos digitales, haciendo de lo complejo algo simple, solucionando problemas y retos de sus clientes, a través de soluciones tecnológicas que generen valor, con el propósito de ser sus aliados preferidos.

Actualmente cuenta con más de 4000 proyectos desarrollados con éxito, destacando en el sector financiero y económico, con más de 136 profesionales que conforman la organización y con presencia en más de 11 países (*Nexos Software – ¡Siempre Conectados!*, s. f.).

### 2.2. FORMULACIÓN DEL PROBLEMA

Para dar un seguimiento más preciso y controlado de los colaboradores que hacen parte de la organización, se requiere el registro diario de las actividades que se encuentran realizando. Esto es información que será tratada por distintas áreas de la organización como facturación, recursos humanos, gerencia, entre otras.

Como se explica en el artículo (*¿Qué es el control del tiempo? | Manatal Contratación*, s. f.), el seguimiento del tiempo proporciona visibilidad en tiempo real de los proyectos y destaca importantes métricas de rendimiento que de otro modo serían invisibles. La evaluación comparativa, la elaboración de presupuestos, la previsión y el cumplimiento requieren un seguimiento del tiempo. Mejora la eficiencia de los equipos, la productividad de las organizaciones y la rentabilidad de las empresas. Además, el registro de tiempos no sólo se utiliza para la facturación de nóminas, sino que también es una forma segura de que se controle a nivel general a dónde va el tiempo de los empleados y cuánto tiempo se tarda en realizar una determinada actividad. Está claro que algunas tareas consumen más tiempo, es posible reflexionar sobre cómo se utiliza ese tiempo adecuadamente.

El registro se realiza a través de una web donde se especifican datos como: cliente al que se le presta el servicio, jefe, fecha y horas trabajadas. Esta web no ha sido actualizada desde su implementación hace un poco más de 10 años, lo que afecta a la experiencia de los colaboradores al momento de realizar el registro, esto debido a su interfaz desactualizada, falta de funciones, lentitud y otras limitaciones, generando desinterés a los usuarios, los cuales esperan que este proceso sea intuitivo, sencillo y ágil.

## 3. PROPUESTA DE PRÁCTICA

### 3.1. JUSTIFICACIÓN

Actualmente, el sistema de registro de tiempos presenta diversas deficiencias que afectan la operatividad y la precisión de los datos en la organización. Estas deficiencias han llevado a que, en múltiples ocasiones, los colaboradores no registren adecuadamente sus días, lo cual genera una serie de inconvenientes tanto para ellos como para la administración. Entre las causas de esta problemática se encuentran la complejidad del sistema, la falta de intuitividad en la interfaz y el tiempo excesivo que toma completar el proceso, factores que contribuyen a la omisión involuntaria del registro.

El objetivo de este proyecto es desarrollar una plataforma web que sea eficiente, intuitiva y fácil de usar para los colaboradores de la organización. Esto permitirá que el proceso de registro de días sea más rápido y accesible, lo cual reducirá significativamente los errores y las omisiones en los datos registrados. Al contar con datos actualizados y precisos, la organización podrá optimizar la planificación y ejecución de sus actividades.

Además, con una plataforma mejorada, se reducirán los reprocesos y las llamadas de atención hacia los colaboradores que, por diversas razones, no cumplan con el proceso de registro actual.

### 3.2. ALCANCE

Para este proyecto se planea realizar las fases de análisis y diseño, donde se realizará el levantamiento de especificaciones funcionales y no funcionales, la definición de casos de uso y diagramas, la creación de prototipos de interfaz, entre otras actividades correspondientes a estas fases. Posteriormente el desarrollo del código fuente y realización de pruebas sobre la página web. Con esto se realiza todo un ciclo de desarrollo desde el análisis inicial hasta su presentación final como proyecto, se debe tomar en cuenta que esta presentación no abarca la implementación final ni despliegue de la página web dentro de la empresa.

### 3.3.OBJETIVO PRINCIPAL

Proponer un sistema de control de tiempos en la empresa siguiendo estándares y buenas prácticas de desarrollo presentando una propuesta moderna, eficiente, que mejore la experiencia de los colaboradores y que permita a la empresa disponer de información actualizada de los proyectos para su análisis, control, seguimiento de objetivos y gestión de los colaboradores.

### 3.4.OBJETIVOS SECUNDARIOS

1. Diseñar una página web con una interfaz de usuario intuitiva, adaptativa y fácil de usar
2. Permitir dentro de las funcionalidades de la página web.
  - a. El registro de horas de los empleados, con su respectiva selección de proyecto y tiempo en el cual participan
  - b. El ingreso mediante autenticación de usuarios.
  - c. Visualización de días registrados en el mes
  - d. Edición de registro de días
  - e. Reporte de días no laborados y/o permisos solicitados
3. Realizar pruebas unitarias, y de funcionalidad para garantizar la usabilidad y correcto funcionamiento de la página web.

## 4. REVISIÓN DE LITERATURA

La medición del tiempo de trabajo es una parte fundamental en el área de recursos humanos. La optima gestión y medición del tiempo y de las actividades laborales, aumenta la productividad y mejora la experiencia profesional de los trabajadores; quienes estarán menos estresados y más enfocados. “Un directivo que quiera gestionar bien su tiempo ha de emplear un método de trabajo idóneo y eficaz, y distribuirlo de la manera más productiva posible; es decir, tiene que autogestionarse adecuadamente. Para ello ha de controlar su vida y no ser un instrumento de su entorno profesional y personal; en definitiva, tiene que autocontrolarse.” (Guillén, 2013). Esto nos abre una idea a la importancia de que en el día a día se planifique de manera efectiva las actividades a realizar. En este contexto es fundamental contar con herramientas que permitan facilitar la medición de tiempos invertidos entre los diversos proyectos o actividades, lo que ayuda a acortar tiempos y organizarse adecuadamente,

**Anteproyecto**  
**FACULTAD DE INGENIERÍA Y CIENCIAS BÁSICAS**  
**Institución Universitaria Politécnico Gran Colombiano**

Versión formato:  
2019-02-20

facilitando también la supervisión y el manejo de esta información a los supervisores y gerencia.

A través de herramientas tecnológicas es posible realizar una gestión óptima del día a día de los trabajadores, se debe aclarar que, si bien la mayoría de estas herramientas presentan un enfoque más amplio al cual se aborda como tal para este proyecto en concreto, y que abarca no solo la posibilidad de registrar los tiempos y brindar información al final del día, sino que cambian la manera de autogestión del trabajo, permitiendo una planificación más elaborada y un mayor alcance.

WorkMeter es un buen ejemplo de este tipo de softwares, “El software de gestión del rendimiento de WorkMeter es una solución de gestión de personas desarrollada por WorkMeter que te permite optimizar la gestión del tiempo” (Software de Gestión del Tiempo RRHH - WorkMeter, s. f.). La manera en que trabaja y recolecciona datos WorkMeter es de forma automática. Se indica que los empleados no tienen que informar de manera manual el tiempo dedicado en sus tareas y/o asignaciones diarias. WorkMeter se instala en el ordenador de los empleados y desde su instalación se comienza a medir el tiempo de actividad, de productividad y el tiempo que se invierte en cada tarea. Este tipo de herramientas es uno de los más completos donde se prioriza la organización del horario de trabajo, la gestión del personal, la toma de decisiones y distribución de carga laboral para facilitar así el seguimiento de los equipos y el rendimiento combinando la flexibilidad de registros diarios de tiempo. Como notamos, precisa un alcance, y da un enfoque más amplio acerca de la gestión del tiempo.

También podemos encontrar como ejemplo Bizneo, “Software de control de horas trabajadas, acceso y presencia” (Software para control de horas trabajadas #1 | Bizneo HR, s. f.). Controla las horas trabajadas por tus colaboradores y cumple la ley con el software de control de horario más robusto y sencillo”. La decisión de las empresas en cuanto a la adquisición de este tipo de herramientas no solo ayudará a optimizar la productividad en el trabajo, sino también garantiza y promueve la optimización de la gestión del tiempo en el trabajo para cada trabajador.

Es interesante la cuestión de establecer o no este tipo de enfoques para el manejo del tiempo de los empleados teniendo como alternativa a estos los softwares de gestión del horario laboral que son brindan un control mucho más avanzado y que generalmente son soluciones construidas para aquellas empresas que tienen a sus empleados trabajando todo el tiempo en oficinas con dispositivos tecnológicos. Encontramos softwares como Time@Work, de la misma familia de WorkMeter donde se instala en el computador asignado a los empleados y se recolectan automáticamente las horas de entrada, horas de salida y la actividad en tiempo real. Esto permite acceder a indicadores de actividad que dan muestra de la productividad, de cómo se invierte el tiempo de cada empleado y de cómo podemos mejorarla en pro de optimizar nuestro tiempo. El equipo de recursos humanos y administración también puede

acceder a dicha información y manejar fácilmente las cargas de trabajo y la asignación de actividades.

Esta es quizá una solución menos estricta, y un poco más colaborativa y transparente, que tiene como finalidad el aportar más datos que cualquier otra herramienta de control de tiempo y/o asistencias aparte de quitar responsabilidad al empleado de hacerlo de manera manual. Así vemos que con este tipo de software se evita los procesos manuales de registro por parte del empleado, y se puede organizar el tiempo de forma más automatizada.

Como se mencionaba anteriormente el enfoque que proponen este tipo de herramientas es más diferente y amplio, no se limita únicamente al registro de tiempos, lo cual no es descartable realizar en un futuro, es decir, la proposición de un sistema más completo para la gestión del tiempo, desarrollado y mantenido de manera privada por la propia empresa.

Aparte de soluciones privadas y bajo licencia disponibles en el mercado, podemos encontrar el sistema desarrollado (Piaget Arana & Mamani Anchea, 2013) proyecto el cual tiene como objetivo según como ellos mencionan, simplificar el seguimiento de proyectos software, mediante la definición de los equipos de trabajo, el manejo de los requerimientos, y la supervisión del registro, medición y control de las actividades y los avances, este software amplía la capacidad de poder crear proyectos, definir los requerimientos, definir los roles en el equipo de trabajo, controlar los requerimientos, asignar responsabilidades, permitir el registro de estas, así como el avance y cierre de las mismas.

Además de este último, tenemos el proyecto “Desarrollo de una aplicación web para el control de tiempos y la gestión de personal en las empresas” (Burgos Martinez, 2021), otro gran ejemplo de una solución adoptada mediante software para disponer de una plataforma web que permita llevar a cabo el registro, consulta y seguimiento laboral de los empleados de una empresa.

Sumado a estos, se presenta (Tuso Guayta, 2016) el cual está también orientado al control del tiempo de los empleados además de la recepción de requerimientos, con esta solución es posible la administración de requerimientos, como su estado, y el registro del tiempo invertido por cada desarrollador en determinado requerimiento.

El siguiente proyecto (Sistema web para el control de tiempos y costos de proyectos en la empresa Soniviu, 2018) presenta una estrecha similitud con el anterior en cuanto a la solución que provee, hablamos del control de tiempos y requerimiento mediante un sistema que permita administrar el registro de requerimientos, los cambios que presenten estos, y la generación de reportes con información de costes y tiempos que permitan marcar una comparativa de los diferentes proyectos.

(Rojas Molina, 2015) presenta un enfoque más amplio y no limitado únicamente al registro y seguimiento de tiempos, sino a un control más orientado a proyectos donde construyen una solución capaz de gestionar usuarios y roles, además de guardar información crucial del proyecto, incluyendo el nombre, descripción, fecha estimada de finalización y miembros

involucrados. Carga de archivos de requerimientos, y registro de tareas específicas por cada requerimiento, fechas estimadas de finalización y la duración de Sprint, asignación de responsabilidades de cada tarea, seguimiento y modificación de estas.

Para finalizar, a lo largo del documento se han presentado distintas soluciones con enfoques parecidos y no tan parecidos al objetivo del proyecto en cuestión, encontrando distintas opciones de softwares comerciales robustas y escalables, que integran características avanzadas como análisis de productividad, compatibilidad y capacidad de automatización. Estas soluciones están diseñadas para empresas de diferentes tamaños y sectores, proporcionando una gestión eficiente del tiempo por parte del empleado y de recursos humanos.

Por otro lado, los distintos proyectos universitarios, aunque un poco más limitados en alcance y recursos, presentan innovaciones y enfoques personalizados ya que se centran en resolver problemas específicos de una organización o implementar mejoras con nuevas tecnologías y metodologías de desarrollo. Esto nos permite contextualizar de mejor manera el estado de este tipo de herramientas, las problemáticas que resuelven y la solución que se provee en cada caso.

## 5. METODOLOGÍA

### 5.1.METODOLOGÍA AGÍL

Basado en las características y requerimientos del proyecto se establece SCRUM como metodología ágil. Esta metodología ya es común en muchas empresas y no solo en el ámbito de desarrollo de software. Al permitir plantear etapas del proyecto, definir los entregables, proporcionar metas claras y específicas, ofrecer flexibilidad al momento de modificar el alcance del proyecto, identificar actividades y agilizar los tiempos de entrega. (*The Scrum Guide*, s. f.)

Para el desarrollo del proyecto se plantean las siguientes ceremonias de SCRUM, esto con el fin de adaptar esta metodología a los requerimientos, plazo y objetivos del proyecto.

#### 1. Planning

Durante esta etapa inicial, se realiza el análisis detallado de los requerimientos del proyecto, se identifican los interesados, se establecen los objetivos y se da un primer alcance de la solución a realizar. En esta etapa se pretende dar claridad a un alto nivel de las actividades que contempla la solución en general y las expectativas de los interesados desde el principio, esto ayudará a evitar desviaciones o cambios abruptos garantizando transparencia en cuanto a

objetivos y expectativas, además, dará paso a la planificación de esas actividades y entrega de resultados divididas en los futuros Sprint.

## 2. Sprint

Esta etapa se define como el período de tiempo en el que se trabaja para completar una cantidad de trabajo o actividades planificada/s. Para el proyecto se definen sprint con una duración de 10 días, lo que se traduce en dos semanas.

## 3. Sprint Planning

En esta etapa se seleccionan las historias de usuario y las actividades a incluir dentro del sprint, se realiza la estimación de tareas con la finalidad de estar alineados con las actividades que se realizarán dentro del sprint.

## 4. Sprint Refinement

El refinamiento se realizará semanalmente con el objetivo de detallar cada historia de usuario, brindar contexto de estas a todo el equipo, mapear posibles dependencias o incidentes que se puedan llegar a presentar, definir los criterios de aceptación de la historia de usuario y realizar la estimación de tiempo de desarrollo/pruebas de cada historia. Es importante que en cada refinamiento se logre conseguir alineamiento entre el PO y el equipo de desarrollo y QA en cuestión de tiempos y resultado, esto para evitar que al cierre de sprint si no se consigue terminar las actividades y no se logre realizar una entrega de valor, no darlo por reportado al cierre del sprint sino plantear con tiempo las dependencias o bloqueos que se tiene o tuvieron para correr el tiempo de entrega hacia el siguiente sprint.

## 5. Sprint Review

Al cierre de sprint, se llevará a cabo una sesión con el objetivo de revisar el trabajo realizado, demostrando de manera funcional el resultado obtenido durante el sprint. En esta fase se podrá obtener retroalimentación del cliente y validar el trabajo realizado, garantizando así la mejora continua del producto.

## 5.2. DISEÑO y CONSTRUCCIÓN DEL SISTEMA

Como condiciones previamente definidas y según el proyecto se definen las plantillas de las historias de usuario y el flujo de datos de la siguiente manera:

Historia de Usuario	
Numero: 1 ( <i>Consecutivo de historia</i> )	Nombre: Guardar registro ( <i>Nombre asignado</i> )
Usuario: ( <i>Tipo de usuario</i> )	Iteración: ( <i># Sprint</i> )
Prioridad: ( <i>Se define prioridad de HU</i> )	Riesgo en Desarrollo: Bajo

**Anteproyecto**  
FACULTAD DE INGENIERÍA Y CIENCIAS BÁSICAS  
**Institución Universitaria Politécnico Gracolombiano**

Versión formato:  
2019-02-20

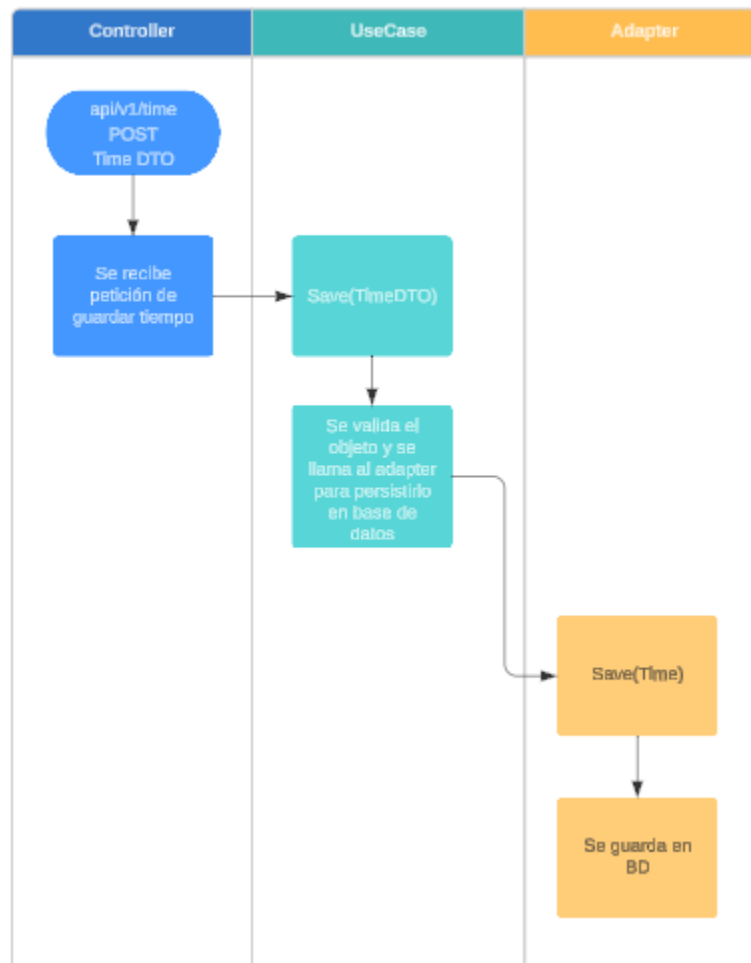
Descripción: (*Descripción detallada de la HU*)

Observaciones:

*Plantilla 1. Historia de usuario*

Representación del flujo de datos que debe seguir la API para las peticiones que se realicen, se establecen tres componentes fundamentales, el controlador, el caso de uso y el adaptador, a continuación, se describe que función cumple cada componente dentro del flujo:

1. Controlador: encargado de recibir la petición desde un cliente (navegador, Postman, etc.), invoca al caso de uso y se encarga también de brindar una respuesta HTTP al cliente.
2. Caso de uso: Almacena la lógica y reglas de negocio según la petición, es el encargado de realizar cualquier tipo de validación si así lo requiere.
3. Adaptador: es invocado desde el caso de uso, este componente es quien mantiene conexión con el motor de base de datos y se encarga de la transaccionalidad de esta.



*Plantilla 2. Flujo de datos API*

## 5.2.1 HISTORIAS DE USUARIO

### 1. Guardar registro de tiempo

Historia de Usuario	
Numero: 1	Nombre: Guardar registro de tiempo
Usuario: Empleado	Iteración:
Prioridad: Alta	Riesgo en Desarrollo: Bajo
Descripción: Como empleado quiero guardar un registro diario de tiempos, diligenciando los datos de proyecto al que pertenezco, líder inmediato, información de facturación, fecha de registro, descripción de actividades realizadas en el día, tiempo total de esas actividades	
Observaciones:	

## 2. Editar registro de tiempo

Historia de Usuario	
Numero: 2	Nombre: Editar registro de tiempo
Usuario: Empleado	Iteración:
Prioridad: Media	Riesgo en Desarrollo: Bajo
Descripción: Como empleado quiero editar un registro de tiempo en caso de querer complementar información, corregir la descripción, tiempo, proyecto, etc. del registro	
Observaciones:	

## 3. Listar registros de tiempo

Historia de Usuario	
Numero: 3	Nombre: Listar registros de tiempo
Usuario: Empleado	Iteración:
Prioridad: Alta	Riesgo en Desarrollo: Bajo
Descripción: Como empleado quiero listar todos los días actuales registrados	
Observaciones:	

## 4. Guardar reporte de días no laborados

Historia de Usuario	
Numero: 4	Nombre: Guardar reporte de días no laborados
Usuario: Empleado	Iteración:
Prioridad: Media	Riesgo en Desarrollo: Bajo
Descripción: Como empleado quiero guardar un reporte de registro de un día no laborado, y en el indicar el tiempo total no laborado y el motivo	
Observaciones:	

**Anteproyecto**  
FACULTAD DE INGENIERÍA Y CIENCIAS BÁSICAS  
**Institución Universitaria Politécnico Gran Colombiano**

Versión formato:  
2019-02-20

### 5. Guardar proyecto

Historia de Usuario	
Numero: 5	Nombre: Guardar proyecto
Usuario: Administrador	Iteración:
Prioridad: Baja	Riesgo en Desarrollo: Bajo
Descripción: Como administrador quiero guardar nuevos proyectos para que estén disponibles al momento de que los empleados llenen el registro diario de tiempos	
Observaciones:	

### 6. Listar proyectos

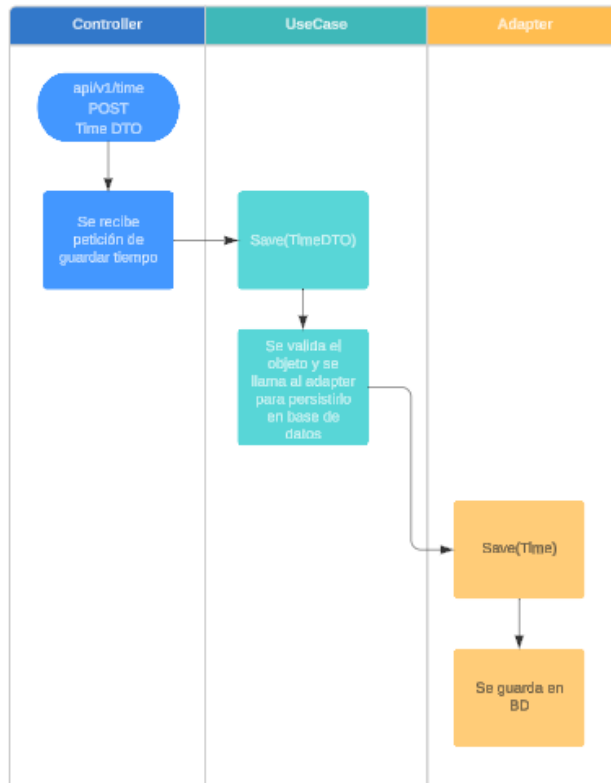
Historia de Usuario	
Numero: 6	Nombre: Listar proyectos
Usuario: Empleado	Iteración:
Prioridad: Baja	Riesgo en Desarrollo: Bajo
Descripción: Como empleado quiero listar los proyectos para seleccionar al que pertenezco al momento de realizar el registro de tiempos	
Observaciones:	

### 7. Eliminar proyectos

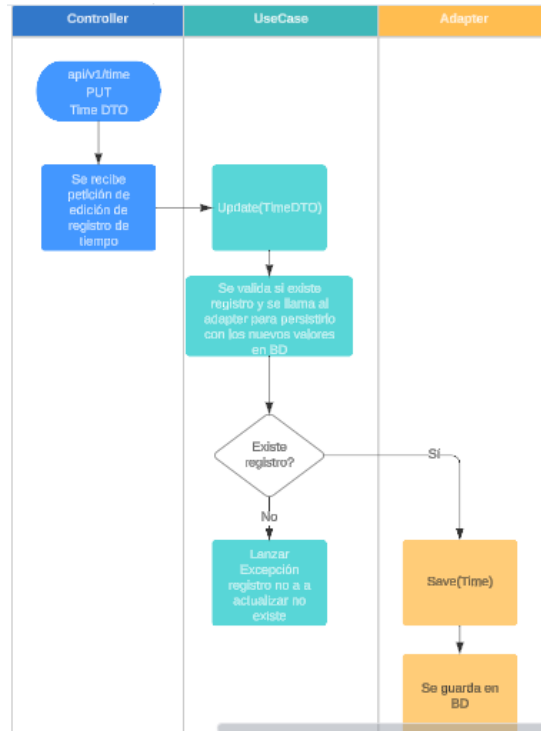
Historia de Usuario	
Numero: 7	Nombre: Eliminar proyectos
Usuario: Administrador	Iteración:
Prioridad: Baja	Riesgo en Desarrollo: Bajo
Descripción: Como administrador quiero eliminar proyectos que ya hayan finalizado para mantener actualizado el listado de proyectos de la empresa	
Observaciones: La eliminación no implica borrar permanentemente el proyecto, esto con el fin de tener en la base de datos una traza del total de proyectos en cual la empresa esta o haya participado y culminado.	

## 5.2.2 LISTADO DE REQUERIMIENTOS Y FLUJO

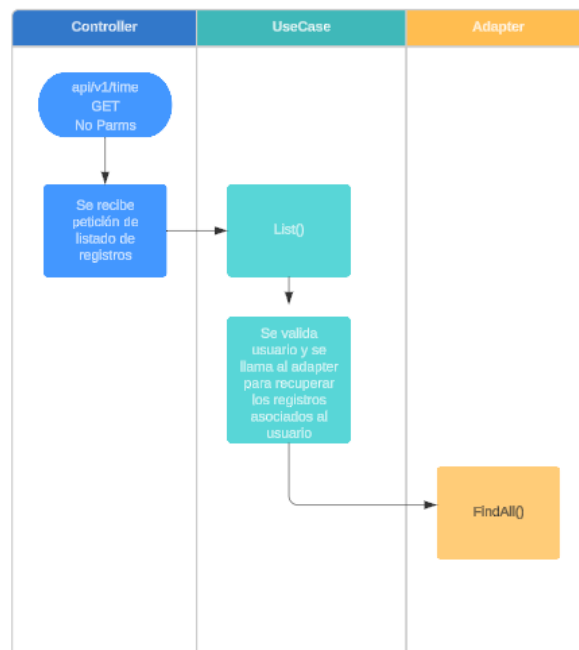
### 1. Guardar registro de tiempo



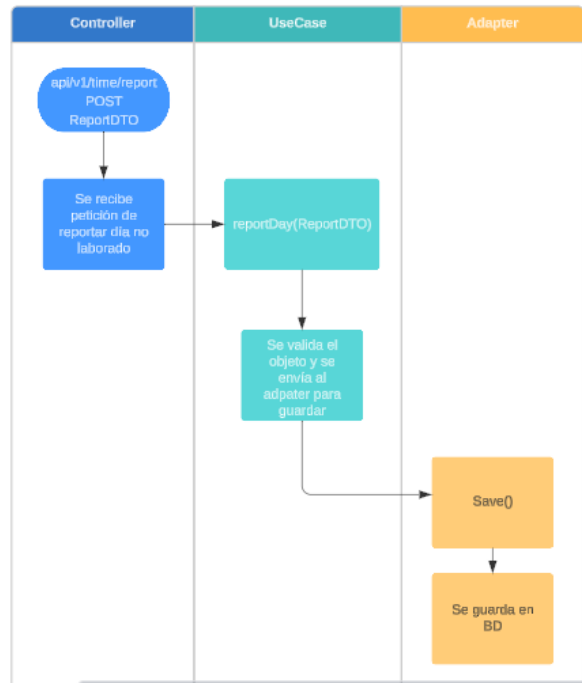
### 2. Editar registro de tiempo



### 3. Listar registros de tiempo



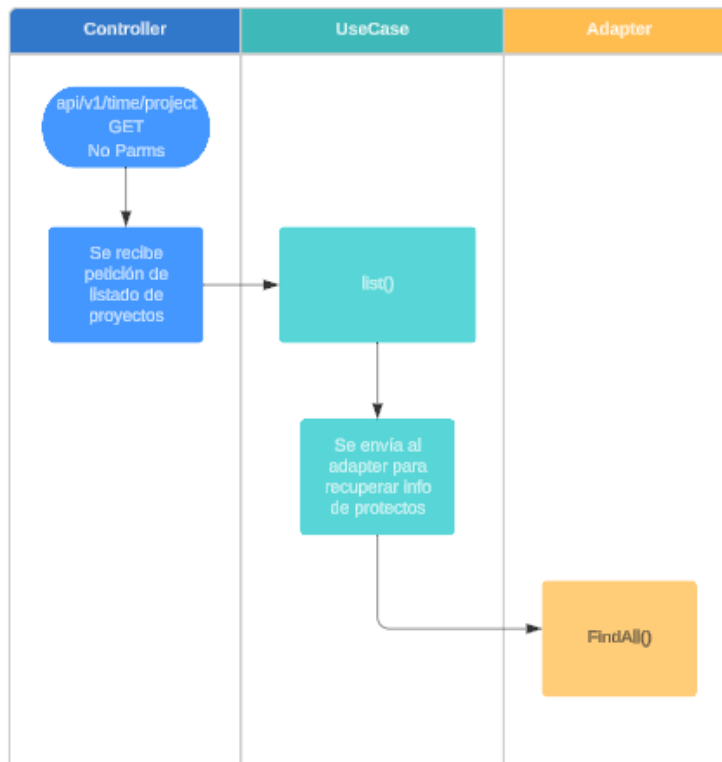
### 4. Guardar reporte de días no laborados



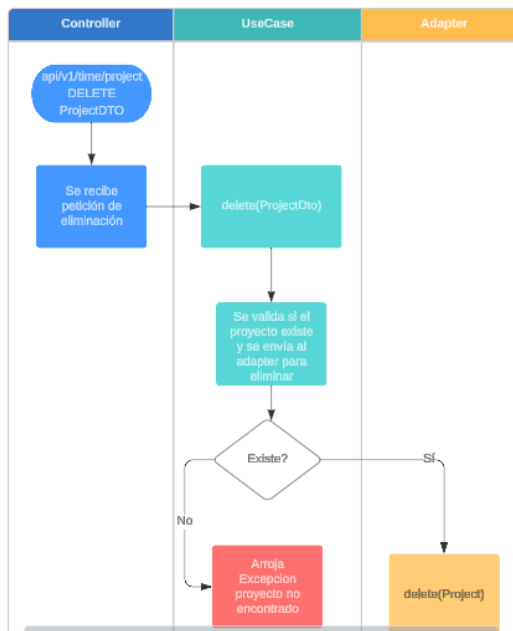
## 5. Guardar proyecto



## 6. Listar proyectos

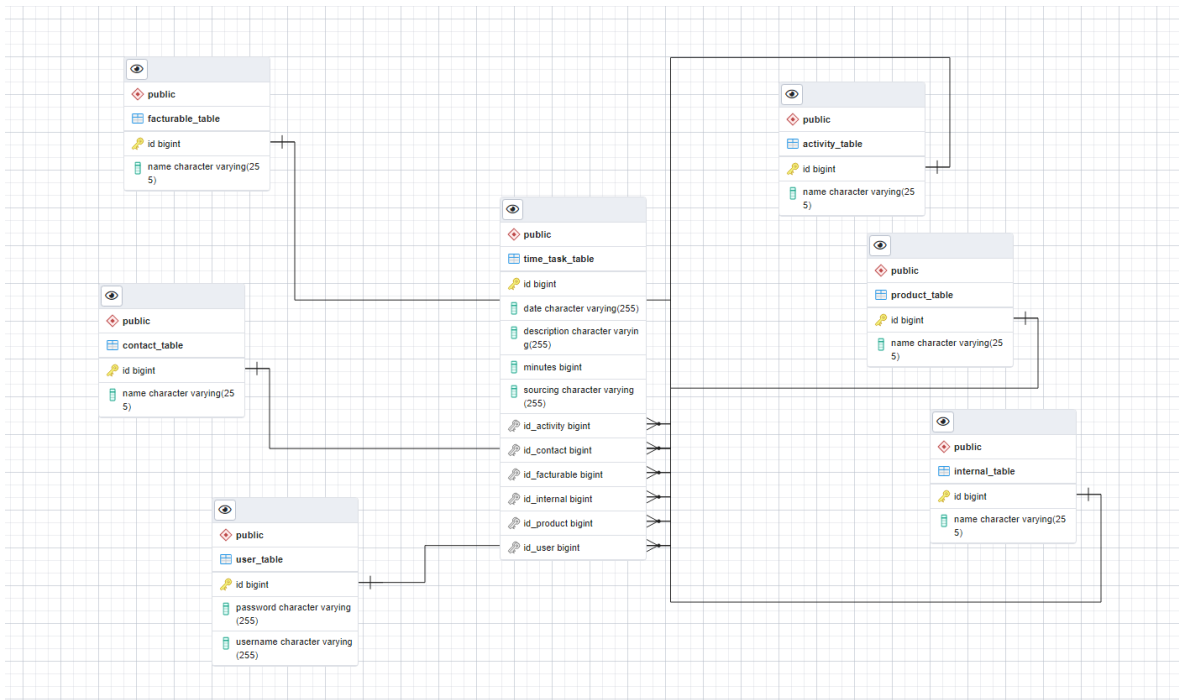


### 7. Eliminar proyectos



### 5.2.3 BASE DE DATOS

A continuación, se presenta el modelo entidad relación del sistema.



El anterior diagrama nos describe:

#### Entidades

facturable_table
id (bigint): Clave primaria. name (character varying (255)): Nombre del elemento facturable.

contact_table
id (bigint): Clave primaria. name (character varying (255)): Nombre del contacto.

**Anteproyecto**  
FACULTAD DE INGENIERÍA Y CIENCIAS BÁSICAS  
**Institución Universitaria Politécnico Gran Colombiano**

Versión formato:  
2019-02-20

user\_table

id (bigint): Clave primaria.  
password (character varying (255)): Contraseña del usuario.  
username (character varying (255)): Nombre de usuario.

activity\_table

id (bigint): Clave primaria.  
name (character varying (255)): Nombre de la actividad.

product\_table

id (bigint): Clave primaria.  
name (character varying (255)): Nombre del producto.

internal\_table

id (bigint): Clave primaria.  
name (character varying (255)): Nombre del elemento interno.

time\_task\_table

id (bigint): Clave primaria.  
date (character varying (255)): Fecha de la tarea.  
description (character varying (255)): Descripción de la tarea.  
minutes (bigint): Minutos dedicados a la tarea.  
sourcing (character varying (255)): Origen de la tarea.  
id\_activity (bigint): Clave foránea que referencia a activity\_table.  
id\_contact (bigint): Clave foránea que referencia a contact\_table.  
id\_facturable (bigint): Clave foránea que referencia a facturable\_table.  
id\_internal (bigint): Clave foránea que referencia a internal\_table.  
id\_product (bigint): Clave foránea que referencia a product\_table.  
id\_user (bigint): Clave foránea que referencia a user\_table.

## Relaciones

Relaciones de time\_task\_table con otras entidades:

- id\_activity: Relación de muchos a uno con activity\_table. Una tarea puede estar relacionada con una actividad específica.
- id\_contact: Relación de muchos a uno con contact\_table. Una tarea puede estar asociada con un contacto específico.
- id\_facturable: Relación de muchos a uno con facturable\_table. Una tarea puede ser relacionada con un elemento facturable.
- id\_internal: Relación de muchos a uno con internal\_table. Una tarea puede estar vinculada a un elemento interno específico.
- id\_product: Relación de muchos a uno con product\_table. Una tarea puede estar asociada con un producto específico.
- id\_user: Relación de muchos a uno con user\_table. Una tarea está asignada a un usuario específico.

## Descripción General

El modelo está centrado en la entidad time\_task\_table, que actúa como el núcleo donde se registran las tareas y sus detalles (fecha, descripción, minutos, y origen). Cada tarea está relacionada con varias otras entidades (activity\_table, contact\_table, facturable\_table, internal\_table, product\_table, y user\_table) a través de claves foráneas, lo que permite categorizar y asociar cada tarea con elementos específicos del sistema, como actividades, contactos, productos, elementos facturables e internos, y usuarios.

Este enfoque permite una gestión detallada y estructurada de las tareas, facilitando la trazabilidad y el análisis de tiempo y recursos dedicados a diferentes aspectos del negocio.

### 5.3. CRONOGRAMA DE ACTIVIDADES

A continuación, se describen las actividades principales y secundarias que se mapearon para el proyecto, asegurando actividades fundamentales para lograr la creación de código de manera correcta como se menciona en (Miguel & Eduardo, 2020) y en (Pressman, 2010), dentro de esto se incluye fase de requerimientos (requerimientos de usuario, funcionales y no funcionales), diseño de interfaz y arquitectura del Sistema, codificación y pruebas. A partir de estas actividades se estableció un cronograma donde se plantearon fechas estimadas en base a los objetivos de cada sprint.

#### 1. Elaboración documento técnico

- Historias de usuario: Describir de manera detallada las funcionalidades que debe tener el sistema desde la perspectiva del usuario final. Con el objetivo de asegurar que el sistema cumpla con las expectativas y necesidades del usuario.

#### 2. Informe de Requerimientos

- Requerimientos funcionales: Documentar las funciones específicas que el sistema debe realizar, garantizando que el sistema tenga todas las capacidades necesarias para cumplir con su propósito.
- Requerimientos no funcionales: Documentar los atributos de calidad que el sistema debe cumplir, como rendimiento, seguridad y usabilidad, asegurando que el sistema no solo funcione correctamente, sino que también sea eficiente, seguro y fácil de usar.

#### 3. Casos de Uso

- Diagrama de flujo de datos: Representar gráficamente cómo se mueve la información a través del sistema, entendiendo el flujo de los datos.
- Plantilla historias de usuario: Seguir con la estructura estandarizada de las historias de usuario. Con eso facilitaremos la comprensión y la gestión de las necesidades del usuario.
- Diagrama de Gantt: Cronograma planificado para controlar los tiempos y recursos del proyecto para asegurar el cumplimiento de los plazos del proyecto.

#### 4. Diseño base de datos

- Diagrama de clases: Representar las clases del sistema, campos y tipos de datos junto con las relaciones entre ellas.

#### 5. Prototipo Maquetas

- Maquetas de Navegación y Maquetas con base en los Mockups: Crear representaciones visuales del sistema antes de su desarrollo completo con el fin de presentar un prototipo visual al cliente antes de realizar desarrollos.

**6. Sistemas de control de versiones**

- Creación de repositorio: Configurar un sistema para gestionar y almacenar todas las versiones del código del proyecto.

**7. Base de Datos**

- Construcción de la Base de Datos: Crear la base de datos según el diseño previamente establecido.
- Inserción de datos en la Base de Datos: Insertar datos en la base información de prueba para validar la integridad y el funcionamiento del sistema con los datos.
- Consultas y joins en la Base de Datos: Crear y optimizar consultas SQL para obtener información de la base de datos.

**8. Prototipo no funcional de pantallas y menú**

- Validar la interfaz de usuario y la navegación del sistema sin necesidad de funcionalidad completa.

**9. Pruebas Técnicas**

- Pruebas unitarias: Verificar la funcionalidad de unidades individuales de código asegurando que cada componente del sistema funciona correctamente por sí solo.
- Pruebas del sistema: Evaluar el sistema completo en su entorno operativo para poder verificar que todas las partes del sistema funcionan juntas correctamente.
- Pruebas funcionales: Verificar que el sistema cumple con los requerimientos funcionales.

**10. Pruebas de desempeño**

- Pruebas de integridad de datos y bases de datos: Asegurar que los datos son consistentes y precisos.
- Pruebas de seguridad y control de acceso: Evaluar la capacidad del sistema para proteger datos y restringir el acceso.
- Pruebas de GUI: Verificar la interfaz de usuario.

**11. Manuales de usuario y de operación**

- Objetivo: Proveer documentación que facilite a los usuarios y operadores entender y utilizar el sistema.

Se adjunta el diagrama de Gantt con las actividades a realizar.



Diagrama de  
Gantt.xlsx

## 5.4.TECNOLOGÍAS

Para el desarrollo de este proyecto se utilizo Java en su versión 17, Angular 14, SpringBoot, PostgreSQL, el stack de tecnologías, frameworks y componentes se presenta a continuación:

### 1. Lenguajes y Frameworks Utilizados

**1.1.Java 17:** se utiliza Java 17 que es una versión LTS (Long-Term Support) la cual trae consigo mayor estabilidad y mejoras en rendimiento, lo que nos proporciona al desarrollar las ultimas características de Java, nos aporta un mayor rendimiento y soporte a largo plazo, asegurando estabilidad y seguridad.

**1.2.Spring Boot:** se utilizó SpringBoot por el hecho de que simplifica la creación de aplicaciones con Java. Nos facilita el desarrollo de aplicaciones robustas y escalables, permite desarrollar aplicaciones de manera más ágil ya que trae configuraciones predeterminadas y prácticas/patrones recomendados, esto nos ahorra tiempo en desarrollo ya que no se deben realizar tantas configuraciones de manera manual. (*Spring Boot, 2024*)

**1.3.JPA (Java Persistence API):** JPA es una especificación de Java para la gestión de datos relacionales en aplicaciones Java a través de mapeo objeto-relacional (ORM), simplifica la interacción con bases de datos relacionales, permitiendo a los desarrolladores trabajar con objetos en lugar de consultas SQL.

**1.4.JUnit:** es un marco de pruebas unitarias para el lenguaje de programación Java. Es utilizado para escribir y ejecutar pruebas repetibles, lo que nos ayuda a asegurar la calidad del código mediante la automatización de pruebas unitarias, lo que facilita la detección de errores en las primeras etapas del desarrollo. (*JUnit 5, 2024*)

### 2. Arquitectura

**2.1.Arquitectura Hexagonal:** también conocida como arquitectura de puertos y adaptadores, esta arquitectura promueve la separación de la lógica de negocio del código de infraestructura y los detalles técnicos, facilita la mantenibilidad, la prueba y la escalabilidad del software al desacoplar los componentes del sistema, permitiendo que la lógica de negocio sea independiente de la interfaz de usuario y la base de datos.

### 3. Base de Datos

**3.1. PostgreSQL:** es un sistema de gestión de bases de datos relacional de código abierto, conocido por su robustez, extensibilidad y estándares de conformidad SQL, proporciona características avanzadas como transacciones ACID, soporte para procedimientos almacenados, y capacidades avanzadas de indexación y búsqueda, garantizando la integridad y eficiencia de los datos. (Group, 2024)

#### 4. Frontend

**4.1. Angular 14:** Angular es un framework de desarrollo frontend basado en TypeScript, diseñado para crear aplicaciones web robustas y escalables, ofrece una estructura sólida para el desarrollo de aplicaciones de una sola página (SPA), mejorando la experiencia del usuario a través de interfaces dinámicas y reactivas. (*Angular - Introduction to the Angular docs*, s. f.)

#### 5. Detalles Técnicos Adicionales

##### 5.1. Gestión de Dependencias y Construcción

**5.1.1. Gradle:** Gradle es una herramienta de gestión de dependencias y automatización de la construcción de proyectos, facilita la gestión de librerías y plugins necesarios para el proyecto, asegurando un proceso de construcción consistente y reproducible. (*Gradle Build Tool*, 2024)

##### 5.2. Control de Versiones

**5.2.1. GitHub:** Sistema de control de versiones distribuido utilizado para el seguimiento de cambios en el código fuente, permite la colaboración eficiente entre los desarrolladores, el manejo de ramas y la integración continua.

**5.3. Spring Security:** este es un módulo de Spring que proporciona autenticación y autorización en aplicaciones basadas en Spring, asegura que solo usuarios autorizados puedan acceder a los recursos de la aplicación, protegiendo los datos sensibles y las funcionalidades críticas. (*Spring Security*, 2024)

**5.4. Mockito:** Herramienta utilizada para realizar pruebas de unitarias (Mockito), asegura que cada componente del sistema funcione correctamente. (*Mockito framework site*, 2024)

Este conjunto de tecnologías y prácticas asegura que el proyecto de software sea robusto, escalable, mantenible y seguro, cumpliendo con los estándares modernos de desarrollo de software.

## 5.5. ENTREGABLES

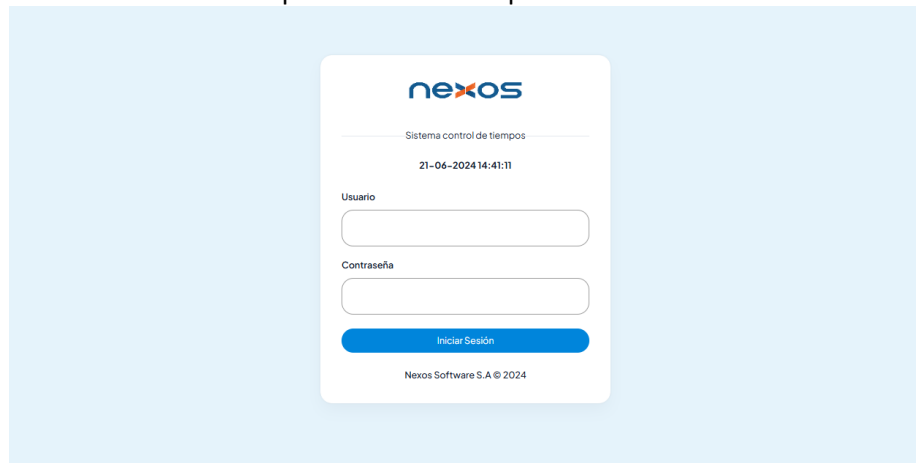
De acuerdo con duración establecida por la universidad para este proyecto de grado y basado en la metodología propuesta, al finalizar cada sprint se entregará como resultado de trabajo la documentación, pruebas funcionales, o evidencia que demuestre los resultados conseguidos al cierre de cada sprint planificado.

Entregable	Estado	Fecha de entrega
Repositorio del código fuente	Completado	15/04/2024
Colección Postman API	Completado	26/04/2024
Flujo de datos	Completado	18/03/2024
Modelo de base de datos	Completado	12/04/2024
Cobertura pruebas unitarias	Completado	01/06/2024
Documentación y manuales	Completado	01/06/2024
Integración BACK - FRONT	Completado	26/05/2024

## 6. RESULTADOS

Una vez realizado el proceso de diseño y desarrollo de la página web que hace el registro de días de los empleados, se obtuvo un sistema capaz de disponibilizar datos precisos y de realizar el registro de estos mismos mediante una página web que presenta una interfaz intuitiva y dispone un manejo rápido para el registro y consulta de los tiempos. Al realizar este proyecto siguiendo la metodología planteada se logró obtener los siguientes resultados.

- Inicio de sesión: desde esta pantalla el usuario podrá autenticarse



**Anteproyecto**  
**FACULTAD DE INGENIERÍA Y CIENCIAS BÁSICAS**  
**Institución Universitaria Politécnico Gran Colombiano**

Versión formato:  
2019-02-20

- Registro de días: desde esta pantalla el usuario podrá diligenciar el formulario de registro presentado, indicando los campos allí definidos, estos valores conformarán la petición a realizar desde el FrontEnd hasta el BackEnd.

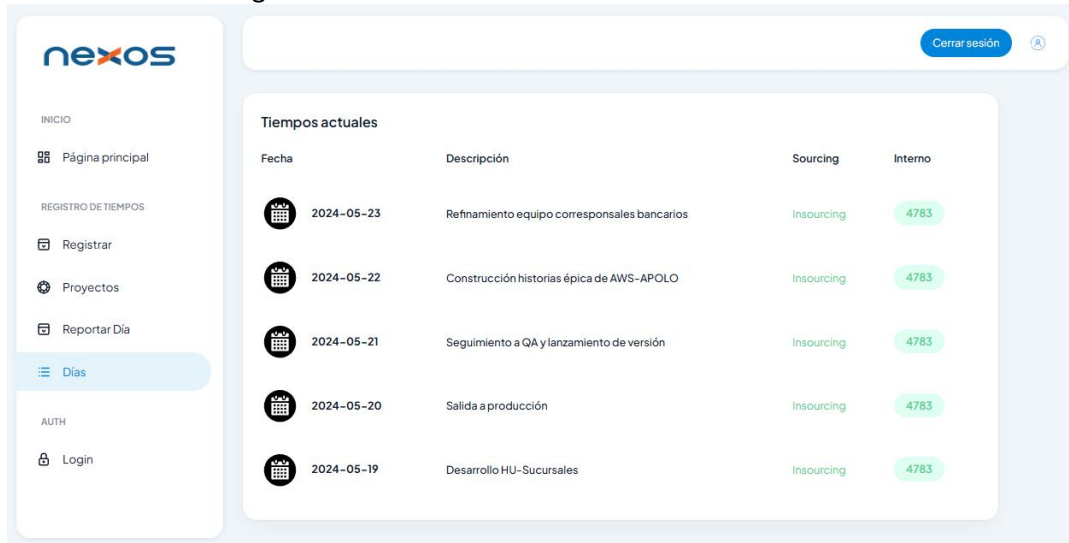
- La integración que se hace desde el FrontEnd hasta el BackEnd, esa petición de registro se traduce a una petición JSON con el siguiente modelo:

```
localhost:8080/apl/v1/time/
POST localhost:8080/apl/v1/time/
Body
[
  {
    "internal": {
      "name": "Credibanco"
    },
    "sourcing": "Insourcing",
    "contact": {
      "name": "Caterin Vasquez"
    },
    "product": {
      "name": "Stand By"
    },
    "activity": {
      "name": "Asunto interno"
    },
    "facturable": {
      "name": "Facturable"
    },
    "date": "13-05-2024",
    "minutes": 480,
    "description": "Thats a description",
    "user": {
      "username": "zohernandez"
    }
  }
]
```

**Anteproyecto**  
**FACULTAD DE INGENIERÍA Y CIENCIAS BÁSICAS**  
**Institución Universitaria Politécnico Gran Colombiano**

Versión formato:  
2019-02-20

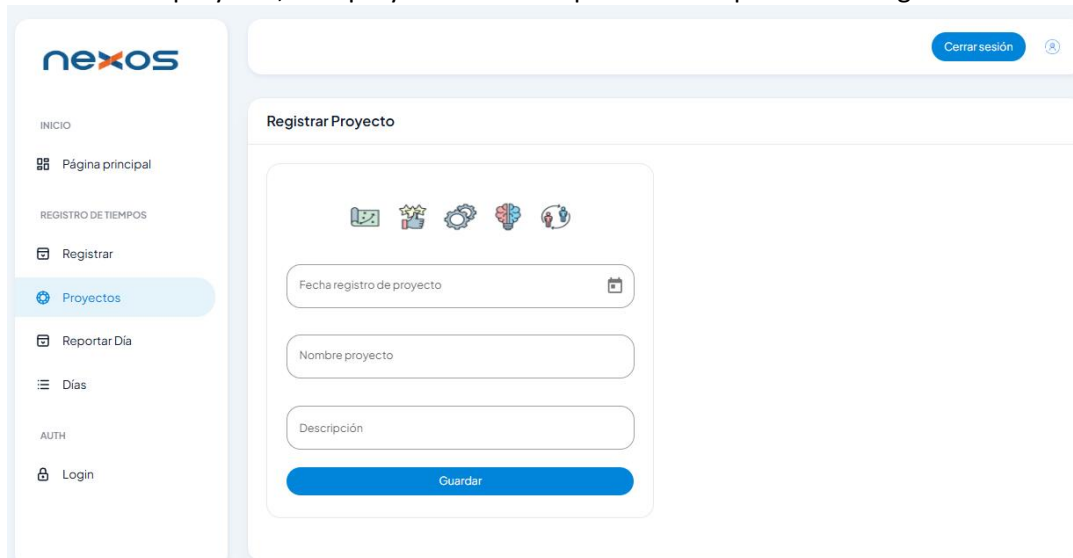
- Una vez diligenciado y enviado el formulario de registro, el usuario podrá consultar el listado de días registrados:



The screenshot shows the 'nexus' dashboard. On the left is a navigation menu with categories: INICIO (Página principal), REGISTRO DE TIEMPOS (Registrar, Proyectos, Reportar Día, Días), AUTH (Login), and a 'Cerrar sesión' button in the top right. The main content area is titled 'Tiempos actuales' and contains a table with the following data:

Fecha	Descripción	Sourcing	Interno
2024-05-23	Refinamiento equipo corresponsales bancarios	Insourcing	4783
2024-05-22	Construcción historias épica de AWS-APOLO	Insourcing	4783
2024-05-21	Seguimiento a QA y lanzamiento de versión	Insourcing	4783
2024-05-20	Salida a producción	Insourcing	4783
2024-05-19	Desarrollo HU-Sucursales	Insourcing	4783

- Registro de proyectos: el usuario administrador podrá acceder a esta pantalla para agregar un nuevo proyecto, este proyecto estará disponible en la pantalla de registro de días.



The screenshot shows the 'Registrar Proyecto' form in the 'nexus' dashboard. The navigation menu is similar to the previous screenshot, but the 'Proyectos' option is highlighted. The form contains the following fields and elements:

- Decorative icons: calendar, gift, gear, brain, and globe.
- Form field: 'Fecha registro de proyecto' with a calendar icon.
- Form field: 'Nombre proyecto'.
- Form field: 'Descripción'.
- Submit button: 'Guardar'.

**Anteproyecto**  
**FACULTAD DE INGENIERÍA Y CIENCIAS BÁSICAS**  
**Institución Universitaria Politécnico Gran Colombiano**

Versión formato:  
 2019-02-20

- Reportar día: desde esta pantalla se podrá realizar el registro de días no laborados o permisos, el usuario podrá indicar el motivo de reporte y el tiempo (en minutos).

- Cobertura de pruebas unitarias: una vez desarrollada la API que recibe las peticiones del cliente, se realizaron pruebas de cobertura en código de los métodos de cada uno de los casos de uso. Estos resultados fueron obtenidos directamente del editor de código ya que no se utilizó ninguna herramienta de Test Coverage para obtener un resultado más detallado sobre estas.

> application	0% (0/62)	0% (0/274)	0% (0/664)
▼ domain	75% (36/48)	63% (178/282)	60% (334/552)
> api	100% (0/0)	100% (0/0)	100% (0/0)
> exception	62% (10/16)	38% (10/26)	33% (10/30)
> model	90% (20/22)	68% (136/200)	51% (142/278)
> spi	100% (0/0)	100% (0/0)	100% (0/0)
> usecase	60% (6/10)	57% (32/56)	74% (182/244)
▼ infrastructure	2% (2/76)	0% (2/394)	0% (2/1112)
> configuration	0% (0/6)	0% (0/32)	0% (0/104)
> exception	100% (2/2)	100% (2/2)	100% (2/2)
> exceptionhandler	0% (0/4)	0% (0/26)	0% (0/76)
> feignconfig	0% (0/2)	0% (0/4)	0% (0/4)
> input	0% (0/6)	0% (0/36)	0% (0/50)
> output	0% (0/46)	0% (0/248)	0% (0/740)
> security	0% (0/10)	0% (0/46)	0% (0/136)

## 7. DISCUSIÓN Y CONCLUSIONES

El desarrollo de la aplicación web de control de tiempos fue un proceso enriquecedor tanto a nivel laboral como académico. El hecho de integrar los conocimientos teóricos adquiridos en la universidad con las prácticas y necesidades reales de la empresa permitió un aprendizaje integral y práctico, sumado al poder continuar trabajando en un entorno ágil con un manejo de herramienta de colaboración y de gestión de proyectos, la experiencia adquirida reforzó la relevancia de aplicar conocimientos académicos en contextos laborales reales.

Uno de los principales retos fue asegurar la precisión y confiabilidad del registro de tiempos, la colaboración con los equipos de TI y recursos humanos pudo contribuir para superar estos obstáculos, mejorando nuestras habilidades técnicas y comunicativas.

Se destaca la importancia que tiene el registro de los tiempos de trabajo, lo cual no solo ayuda a mejorar la eficiencia operativa, sino que también proporciona datos valiosos para la toma de decisiones estratégicas por parte de la gerencia. La plataforma permite a la empresa tener un acceso centralizado a datos precisos sobre el uso del tiempo, lo cual es crucial para evaluar la productividad, planificar y organizar recursos de manera más efectiva. La capacidad de medir y analizar los tiempos de trabajo proporciona insights importantes que pueden influir en la mejora continua de los procesos y en la optimización de los recursos humanos.

## BIBLIOGRAFÍA

Burgos Martinez, A. (2021). *Desarrollo de una aplicación web para el control de tiempos y la gestión de personal en las empresas* [Proyecto/Trabajo fin de carrera/grado, Universitat Politècnica de València]. <https://riunet.upv.es/handle/10251/172394>

Guillén, M. del C. M. (2013). *Administración y planificación del tiempo: La gestión empresarial*. Ediciones Díaz de Santos.

Piaget Arana, X., & Mamani Anchea, L. C. (2013). Sistema informático para el seguimiento y supervisión de proyectos de Desarrollo de software. *Universidad Peruana de Ciencias Aplicadas (UPC)*. <https://repositorioacademico.upc.edu.pe/handle/10757/273549>

Rojas Molina, L. N. (2015). *Desarrollo de una herramienta web para la planificación, ejecución y control de proyectos de desarrollo de sistemas web para la empresa HC Consultores*. [bachelorThesis, 2015.]. <http://bibdigital.epn.edu.ec/handle/15000/10920>

*Sistema web para el control de tiempos y costos de proyectos en la empresa Soniviu*. (2018). <https://repositorio.utp.edu.pe/handle/20.500.12867/1775>

Tuso Guayta, A. D. (2016). *Desarrollo de un sistema web para la recepción de requerimientos y control de tiempo empleado por los desarrolladores en la ejecución de proyectos en la empresa Gamedasoft* [bachelorThesis, Quito, 2016.]. <http://bibdigital.epn.edu.ec/handle/15000/16762>

Miguel, H. B., & Eduardo, B. R., Luis. (2020). *Ciclo de vida de desarrollo ágil de software seguro*. Editorial Los Libertadores.

Pressman, R. S. (2010). *Ingeniería del Software. Un Enfoque Practico*.

### Otros enlaces de interés

*Angular—Introduction to the Angular docs*. (s. f.). Recuperado 24 de junio de 2024, de <https://v17.angular.io/docs>

*Gradle Build Tool*. (2024). Gradle. <https://gradle.org/>

Group, P. G. D. (2024, junio 24). *PostgreSQL*. PostgreSQL. <https://www.postgresql.org/>

*JUnit 5*. (2024). <https://junit.org/junit5/>

*Mockito framework site*. (2024). <https://site.mockito.org/>

*Nexos Software – ¡Siempre Conectados!* (s. f.). Recuperado 12 de marzo de 2024, de <https://nxs.com.co/>

**Anteproyecto**  
**FACULTAD DE INGENIERÍA Y CIENCIAS BÁSICAS**  
**Institución Universitaria Politécnico Gracolombiano**

Versión formato:  
2019-02-20

¿Qué es el control del tiempo? | Manatal Contratación. (s. f.). Recuperado 12 de marzo de 2024, de <https://www.manatal.com/es/glossary/time-tracking>

Sistema web para el control de tiempos y costos de proyectos en la empresa Soniviu. (2018). <https://repositorio.utp.edu.pe/handle/20.500.12867/1775>

Software de Gestión del Tiempo RRHH - WorkMeter. (s. f.). Recuperado 19 de abril de 2024, de <https://www.workmeter.com/software-gestion-del-tiempo/>

Software para control de horas trabajadas #1 | Bizneo HR. (s. f.). Recuperado 19 de abril de 2024, de [https://www.bizneo.com/es-co/control-horario/?utm\\_campaign=ADS-GO-ContHorario-LATAM&utm\\_term=%2Bsoftware%20%2Bjornada%20%2Blaboral&utm\\_content=499639175897&utm\\_medium=Advertisement&utm\\_source=GoogleAds&gad\\_source=1&gclid=EAlaIqobChMI08L24IfPhQMVobhaBR0OVgQYEAAYASAAEgLqBvD\\_BwE](https://www.bizneo.com/es-co/control-horario/?utm_campaign=ADS-GO-ContHorario-LATAM&utm_term=%2Bsoftware%20%2Bjornada%20%2Blaboral&utm_content=499639175897&utm_medium=Advertisement&utm_source=GoogleAds&gad_source=1&gclid=EAlaIqobChMI08L24IfPhQMVobhaBR0OVgQYEAAYASAAEgLqBvD_BwE)

Spring Boot. (2024). Spring Boot. <https://spring.io/projects/spring-boot>

Spring Security. (2024). Spring Security. <https://spring.io/projects/spring-security>

The Scrum Guide. (s. f.). Recuperado 5 de abril de 2024, de <https://www.scrum.org/resources/scrum-guide>

## ANEXOS

### ANEXO 1. CARTA DE ACEPTACIÓN DE LA EMPRESA