**CRONOGRAMA Y ACTIVIDADES DEL PERIODO**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | Diciembre | | | |
| **ID** | **DESCRIPCION DE LA ACTIVIDAD** | S1 | S2 | S3 | S4 |
| 1 | OBJETIVO GENERAL | $10.00 |  |  | $598.00 |
| 1.2 | Objetivo Especifico 3 |  |  |  |  |
| 1.2.2 | Fase 3: Desarrollo de funcionalidades |  |  |  |  |
| 1.2.2.3 | Desarrollo del módulo de trainer |  |  |  |  |
| 1.2.2.4 | Desarrollo del módulo de trainee |  |  |  |  |
| 1.2.2.5 | Integración de módulos |  |  |  |  |
| Otros: | Salario |  |  |  | $500.00 |
|  | Internet |  |  |  | $33.00 |
|  | Energía eléctrica |  |  |  | $65.00 |

**INFORME DE AVANCES DE EJECUCION DE PROYECTO DESARROLLO DE API-REST PARA LA GESTION DE USUARIOS Y EVALUACIÓN DE DESEMPEÑO DE PERSONAL PARA EL PERIODO 27/11/2023 AL 30/12/2022**

|  |  |
| --- | --- |
| 1 | **OBJETIVO GENERAL** |
| 1.2 | Objetivo Especifico 2 |
| 1.2.2 | Fase 3: Desarrollo de funcionalidades |
| 1.2.2.3 | Desarrollo del módulo de trainer |
| 1.2.2.4 | Desarrollo del módulo de trainee |
| 1.2.2.5 | Integración de módulos |

OBJETIVO GENERAL

Desarrollar una interfaz de programación de aplicaciones (API), para la evaluación y capacitación de empleados de la empresa Sistemas Pliret, S.A. de C.V., que le permita optimizar y simplificar el proceso de registro, análisis y seguimiento de datos para todo su personal.

Nivel de avance 78% del total del proyecto alcanzado en el periodo del 28/08/2023 al 30/12/2023.

# 1.2 Objetivo Especifico 2

Garantizar la seguridad de los datos de evaluación por medio de la encriptación, requerimiento de Json Web Tokens y autentificación por roles a los distintos *endpoints* de la API.

Fecha esperada de finalización: 29/12/2023.

Nivel de avance: 100% del total del objetivo.

## 1.2.2 Fase 3: Desarrollo de funcionalidades.

Se programaron los *endpoints* restantes para el *trainer* que habían quedado pendientes por codificar en el sprint anterior además de lograrse completar en su totalidad el apartado de funcionalidad del usuario *trainee* además de lograrse la integración de los cuatro módulos de usuario para una interconexión según los procesos a realizar que involucren a más de un tipo de usuario. Con esto se da por terminado el desarrollo de la fase 3 para pasar con la ultima fase en el siguiente sprint.

## 1.2.2.3 Desarrollo del módulo de trainer.

Los endpoints programados para este usuario corresponden a la toma de evaluación por medio del apartado de las entrevistas y por la revisión de código además de la creación de eventos y clases servicio para realizar en segundo plano la captura de contenido del programa a la hora de evaluar a un empleado junto con las acciones que toman las puntuaciones del usuario asignado y promedian la nota de la ejecución a la que pertenece.

Este promedio puede ser accedido por un endpoint especifico para crear un reporte .csv con los promedios e información relevante del empleado luego de la finalización de la ejecución de un programa (véase anexo 1).

## 1.2.2.4 Desarrollo del módulo de trainee.

La mayor complejidad de programar este apartado fue la subida de la tarea o código a evaluar para la *code* *review* de la metodología *scrum* que se realiza cada día por las *daily* *meetings* en dicha modalidad. Principalmente fue corregir errores propios del trainer a la hora de tomar las observaciones y notas de los puntos a evaluar del código del empleado y su posterior promedio por código en los endpoints de tomar evaluación del trainer e ir preparando la interconectividad entre estos apartados para la siguiente parte de esta fase con services clases para delegar lógica de los procesos internos a realizar entre ambos (véase anexo 2).

## 1.2.2.5 Integración de módulos.

Se hizo uso de middlewares para separar los endpoints de cada tipo de usuario según el permiso correspondiente a su *json* *web* *token* (jwt) y a la creación de archivos *resource* para el despliegue de datos en común para los cuatro tipos de usuarios. Paralelamente se configuro el archivo *app* *provider* del proyecto para habilitar el uso de *listeners* y *events* para delegar lógica de procesos en clases *services*. Se crearon archivos *factory* para utilizar datos de prueba durante la interconectividad e ir comprobando un optimo funcionamiento, estos archivos generan datos del paquete *faker*, un paquete que nos permite rellenar registros de prueba en la base de datos o durante el tiempo de ejecución que fue instalado con *composer*, el gestor de paquetes de PHP (véase anexo 3 y 4).

## Resultados obtenidos entre las fechas 27/11/2022 al 30/12/2023

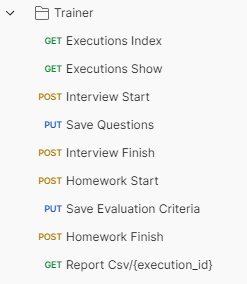
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | Estado | Conc | Pend |
| 1 | OBJETIVO GENERAL | Avance Parcial | 78% | 22% |
| 1.2 | Objetivo Especifico 2 | Finalizada | 100% | 0% |
| 1.2.2 | Fase 3: Desarrollo de funcionalidades | Finalizada | 100% | 0% |
| 1.2.2.3 | Desarrollo del módulo de trainer | Finalizada | 100% | 0% |
| 1.2.2.4 | Desarrollo del módulo de trainee | Finalizada | 100% | 0% |
| 1.2.2.5 | Integración de módulos | Finalizada | 100% | 0% |

## Resultados por actividad

|  |  |  |  |
| --- | --- | --- | --- |
| 1.2 | Objetivo Especifico 2 | Terminado |  |
| 1.2.2 | Fase 3: Desarrollo de funcionalidades | Funcionalidades terminadas |  |
| 1.2.2.3 | Desarrollo del módulo de trainer | Completado | Anexo 1 |
| 1.2.2.4 | Desarrollo del módulo de trainee | Completado | Anexo 2 |
| 1.2.2.5 | Integración de módulos | Integración y correcciones | Anexo 3 y 4 |

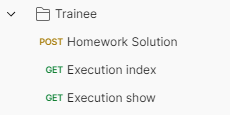
# Anexo de resultados:

## Anexo 1

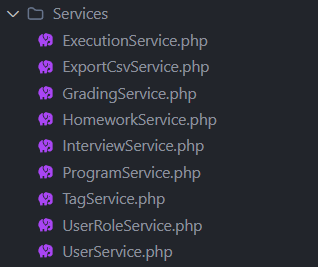
****

*Endpoints planificados y programados para el rol de trainer.*

Anexo 2

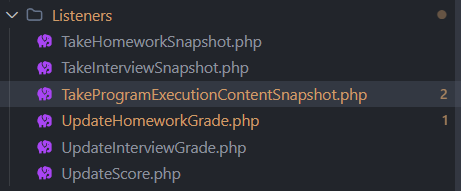
****

*Endpoints planificados y programados para el rol de trainee.*

****Anexo 3

*Clases services que delegan lógica de programación que se elimina de los controladores.*

## Anexo 4



*Listeners enlazados a eventos para una delegación de procesos en segundo plano que afecta a la funcionalidad de módulos cuando otro modulo relacionado interactúa con ellos.*