



API de Evaluación de Empleados

Manual de Usuario

Versión: 0100

Fecha: 30/01/2024



1.0	<Nombre Proyecto> Manual de Usuario	Gerencia de Tecnología
-----	--	------------------------

HOJA DE CONTROL

Proyecto	API de Evaluación de Empleados		
Entregable	Manual de Usuario		
Autor	Roberto Antonio Moreno Zepeda		
Versión/Edición	0100	Fecha Versión	30/01/2024
		Nº Total de Páginas	12

REGISTRO DE CAMBIOS

Versión	Causa del Cambio	Responsable del Cambio	Fecha del Cambio
0100	Versión inicial	Roberto Antonio Moreno Zepeda	30/01/2024



1.0	<Nombre Proyecto> Manual de Usuario	Gerencia de Tecnología
-----	--	------------------------

ÍNDICE

1	DESCRIPCIÓN DEL SISTEMA	4
1.1	Objeto	4
1.2	Alcance.....	4
1.3	Funcionalidad	4
2	MAPA DEL SISTEMA	5
2.1	Diagrama Entidad Relación.....	5
2.2	Navegación	6
3	DESCRIPCIÓN DEL SISTEMA	10
4	DICCIONARIO DE DATOS	12



1.0	<Nombre Proyecto> Manual de Usuario	Gerencia de Tecnología
-----	--	------------------------

1 DESCRIPCIÓN DEL SISTEMA

1.1 Objeto

Esta API proporciona un servicio RESTful para evaluar el desempeño en conocimiento técnico de los empleados de una empresa. El sistema está construido con PHP utilizando el framework Laravel en su 9 versión, utiliza JWT o Json Web Tokens para la autenticación y autorización.

1.2 Alcance

Este sistema proporciona una API RESTful para evaluar el desempeño en conocimiento técnico de los empleados de la empresa. La API está construida con Laravel 9 y utiliza tokens de acceso para la autorización.

1.3 Funcionalidad

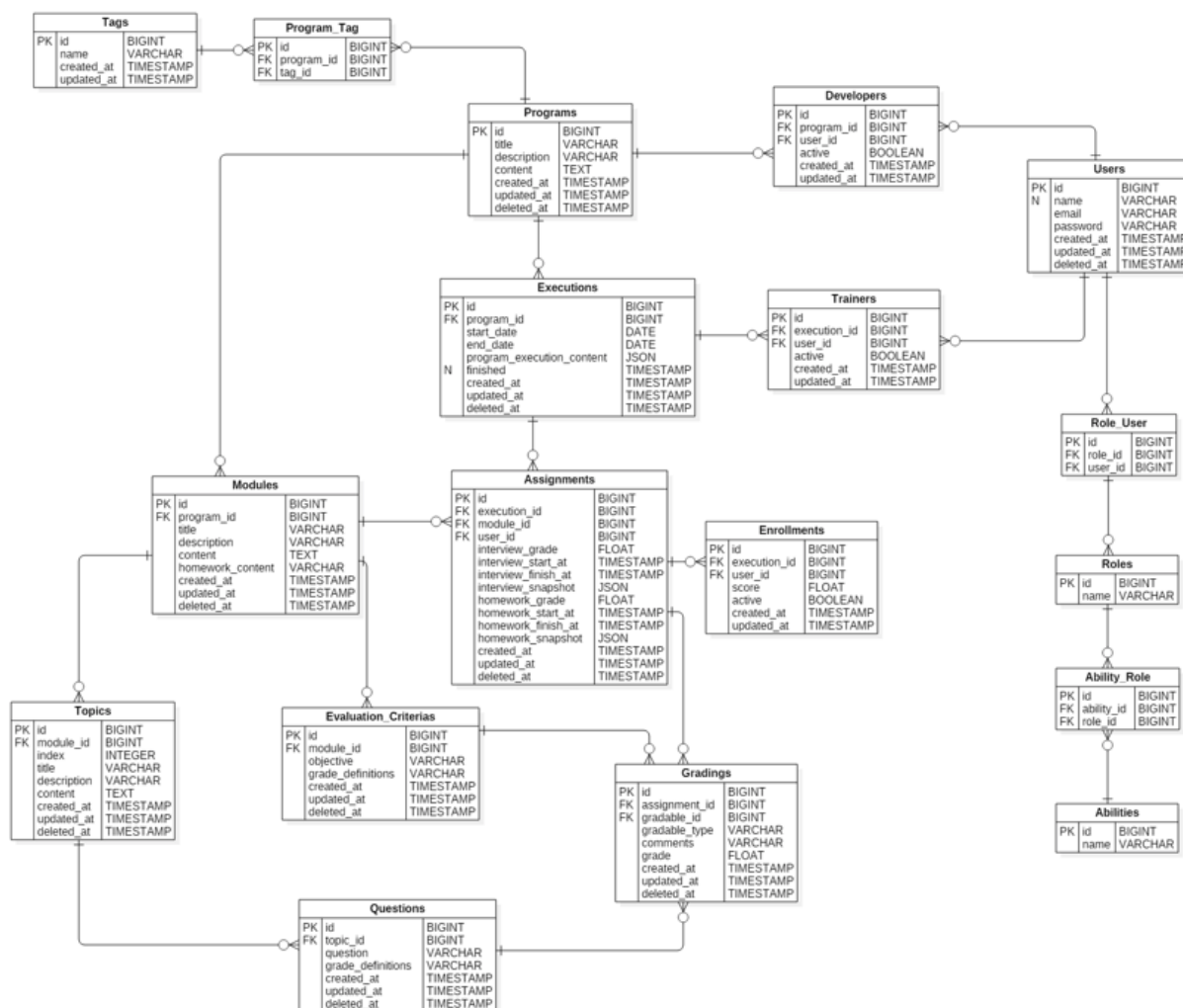
Evaluación de conocimiento de personal de la empresa, permite registrar y gestionar usuarios, realizar evaluaciones de desempeño, almacenar y analizar datos, y visualizar el progreso individual y colectivo. La API utiliza tokens de acceso para la autenticación y autorización e implementa pruebas unitarias para asegurar la calidad del código.

1.0	<Nombre Proyecto> Manual de Usuario	Gerencia de Tecnología
-----	--	------------------------

2 MAPA DEL SISTEMA

2.1 Diagrama Entidad Relación

API de evaluación de empleados de Pliret.





1.0	<Nombre Proyecto> Manual de Usuario	Gerencia de Tecnología
-----	--	------------------------

2.2 Navegación

El concepto fundamental en cualquier API REST es el recurso o “resource”, por lo que se ha optado por seguir un Diseño Orientado a Recursos (Resource Oriented Design).

Un recurso es un objeto con un tipo, datos asociados, relaciones con otros recursos y un conjunto de métodos HTTP que operan sobre él. Es similar a una instancia de un objeto en un lenguaje de programación orientado a objetos, con la importante diferencia de que solo se definen unos pocos métodos estándar para el recurso (correspondientes a los métodos HTTP estándar GET, POST, PUT, PATCH y DELETE), mientras que una instancia de objeto tiene muchos más métodos.

Los recursos internamente se han agrupado en colecciones (collections). Cada colección es homogénea de modo que contiene un solo tipo de recurso y cada acción depende del método HTTP en header de la petición. Los recursos también pueden existir fuera de las colecciones, pero en este caso los recursos aislados los hemos enfocado únicamente al registro de usuarios, a la toma de evaluación y exportación CSV.

Cada respuesta devolverá un payload con su respectivo código de estado con su respuesta json.

Los métodos HTTP corresponden a las siguientes acciones:

GET /resource

Método HTTP GET y sin parámetro en URL

Index: Mostrar la lista del recurso seleccionado.

GET /resource/{id}

Método HTTP GET y parámetro por URL

Show: Mostrar el detalle del recurso seleccionado según el id en la petición.

POST /resource

Método HTTP POST y form request (inputs)

Store: Registrar un nuevo recurso.



1.0	<Nombre Proyecto> Manual de Usuario	Gerencia de Tecnología
-----	--	------------------------

PATCH /resource/{id}

Método HTTP PUT, form request y parámetro id por URL.

Update: Actualización de datos según el recurso seleccionado por id.

DELETE /resource/{id}

Método HTTP DELETE y parámetro por URL

Destroy: Eliminación de un registro asociado a un recurso según el id en la petición.

A continuación, se muestran las colecciones y recursos:

Manager.

POST /login -> Login

GET /executions/{?} -> Index, Show

POST PATCH DELETE /executions/{?} -> Store, Update, Destroy

GET /programs/{?} -> Index, Show

POST PATCH DELETE /programs/{?} -> Store, Update, Destroy

POST /users/create-trainee-account -> Store

POST /users/create-employee-account -> Store

GET POST PATCH DELETE /users/{?} -> Index, Show, Update, Destroy

GET /executions/{execution}/enroll-trainee/{trainee} -> Asignar usuario trainee a una ejecución de un programa

POST /executions/assign-trainee-module -> Asignar modulo a trainee

GET /programs/{program}/assign-developer/{developer} -> Asignar developer a un programa

GET /executions/{execution}/assign-trainer/{trainer} -> Asignar trainer a una ejecución.



1.0	<Nombre Proyecto> Manual de Usuario	Gerencia de Tecnología
-----	--	------------------------

Developer.

GET /programs/{?} -> Index, Show

GET /modules/{?} -> Index, Show

POST PATCH DELETE /modules/{?} -> Store, Update, Destroy

GET /topics/{?} -> Index, Show

POST PATCH DELETE /topics/{?} -> Store, Update, Destroy

GET /questions/{?} -> Index, Show

POST PATCH DELETE /questions/{?} -> Store, Update, Destroy

GET /evaluations/{?} -> Index, Show

POST PATCH DELETE /evaluations/{?} -> Store, Update, Destroy

Trainer.

GET /executions/{?} -> Index, Show

POST /assignments/{assignment}/interview-start -> Iniciar entrevista

PATCH /assignments/save-question -> Guardar respuestas de entrevista

POST /assignments/{assignment}/interview-finish -> Terminar entrevista

POST /assignments/{assignment}/homework-start -> Iniciar evaluación de código

PATCH /assignments/save-evaluation-criteria -> Guardar asignación de puntos a evaluar

POST /assignments/{assignment}/homework-finish -> Terminar evaluación de código



1.0	<Nombre Proyecto> Manual de Usuario	Gerencia de Tecnología
-----	--	------------------------

GET /report/{?} -> Generar reporte de ejecución

Trainee.

GET /executions/{?} -> Index, Show

POST /assignments/{assignment}/homework-solution -> Subir código a evaluar



1.0	<Nombre Proyecto> Manual de Usuario	Gerencia de Tecnología
-----	--	------------------------

3 DESCRIPCIÓN DEL SISTEMA

La empresa Sistemas Pliret, especializada en servicios de subcontratación de personal para empresas que requieren talento en el ámbito de desarrollo de software, cuenta con un equipo de entrenamiento y evaluación de nuevos empleados y demás personal. Esto se lleva a cabo a través de un proceso de seguimiento de una entrevista generalmente acompañada de una prueba técnica. Actualmente, este proceso genera desafíos relacionados con la falta de estandarización debido a la diversidad de requerimientos técnicos. Por lo tanto, surge la necesidad de desarrollar una herramienta que agilice y reduzca los tiempos de evaluación, al mismo tiempo que ofrezca una capacitación más efectiva.

Tomando en consideración la problemática descrita, se plantea el desarrollo de una Interfaz de Programación de Aplicaciones o API por sus siglas en inglés, que proporcione la integración con otras aplicaciones para la gestión de usuarios, entrenamiento de empleados, capacitaciones en línea y mejora de las habilidades y conocimiento tanto de los empleados actuales como del personal de nuevo ingreso.

Esta herramienta estará destinada hacia cuatro tipos de usuarios quienes podrán interactuar con la API de acuerdo con su nivel de autorización y según la gestión hacia la cual sea requerida, así esos niveles de estratificación serían:

- Manager: Quien se encarga de monitorizar y controlar y administrar la plataforma al resto de usuarios.
- Developer: Quien es responsable de la creación de los programas de entrenamiento para el personal de la empresa.
- Trainer: Usuario designado para llevar a cabo las evaluaciones del personal y nuevos empleados.
- Trainee: Todos los usuarios que ingresan al programa de capacitaciones.

La implementación de esta herramienta permitirá brindar una solución integral y eficiente en la gestión de los procesos de capacitación y evaluación del personal actual y de nuevo ingreso.



1.0	<Nombre Proyecto> Manual de Usuario	Gerencia de Tecnología
-----	--	------------------------

Con el desarrollo de la Interfaz de Programación de Aplicaciones, la empresa podrá centralizar y automatizar las diversas tareas relacionadas con la formación de sus empleados en las diferentes áreas que lo requiera.

Así mismo, la API permitirá al Manager hacer un seguimiento más preciso del progreso individual y colectivo de forma eficaz y oportuna. Por otro lado, la herramienta podrá impulsar el desarrollo profesional y técnico de los empleados que conforman Sistemas Pliret, permitiéndole ajustar los programas de estudio de acuerdo con las necesidades y objetivos de la organización.

Esta API impulsará a Sistemas Pliret para ser más competitivos y ofrecer al mercado laboral personal altamente capacitado y actualizado con las herramientas tecnológicas y de mayor tendencia en la sociedad actual.



1.0	<Nombre Proyecto> Manual de Usuario	Gerencia de Tecnología
-----	--	------------------------

4 DICCIONARIO DE DATOS

- **Endpoint:** Es la dirección URL específica de un recurso en la API Rest.
Por ejemplo, <https://api.mitienda.com/productos> sería un endpoint para obtener información de los productos en una tienda en línea.
- **Método HTTP:** Es el verbo utilizado para definir la acción que se va a realizar sobre el recurso. Los métodos HTTP más comunes son GET, POST, PUT, PATCH y DELETE.
- **URL:** Es una cadena de caracteres que identifica el recurso y se utiliza en conjunto con el método HTTP para acceder a él. La URI se compone del endpoint y los parámetros.
- **Parámetros:** Son valores adicionales que se envían con la solicitud HTTP para ayudar a definir la acción que se desea realizar. Los parámetros pueden ser incluidos en la URI o en el cuerpo de la solicitud.
- **Payload:** Es la información adicional que se envía en la solicitud HTTP. El payload puede estar en formato JSON o XML y puede contener datos adicionales para la creación o actualización de un recurso.
- **Autenticación:** Es el proceso de verificación de la identidad de un usuario o aplicación para permitir el acceso a los recursos protegidos. La autenticación se realiza mediante la inclusión de credenciales de usuario o de aplicación en la solicitud HTTP.
- **Respuesta HTTP:** Es la respuesta que devuelve la API Rest después de procesar la solicitud HTTP. La respuesta puede estar en formato JSON o XML y puede incluir datos adicionales o mensajes de error.
- **Códigos de estado HTTP:** Son códigos numéricos que indican el resultado de una solicitud HTTP. Los códigos de estado más comunes son 200 OK (éxito), 400 Bad Request (solicitud incorrecta), 401 Unauthorized (no autorizado), 404 Not Found (recurso no encontrado) y 500 Internal Server Error (error interno del servidor).
- **Framework:** Un framework es una estructura o conjunto de herramientas, bibliotecas de código.
- **JSON:** El formato de intercambio de datos utilizado en la API REST.