

# Crear Marcación (1) - [crear]

## ? Descripción

Registra una **marcación de entrada o salida** para un empleado, validando previamente:

- Datos obligatorios del request.
- Duplicidad de marcaciones.
- Estado del empleado (Activo / PreActivo).
- Horario asignado (tardanza y horas extra).
- Reglas especiales según modalidad (offline / online).
- Registro simultáneo en:
  - **ERP (Employee Checkin)**
  - **Base de datos interna (tabla marcaciones)**

Es uno de los servicios centrales del módulo de asistencia.

---

## ? Endpoint

**POST** /marcaciones/store

Este servicio **sí recibe parámetros en el request**.

---

## ? Request Body (Campos usados)

```
{
  "empleado": "EMP-0001",
  "image": "https://.../foto.jpg",
  "fecha_asistencia": "2025-02-03",
  "type": "entrada | salida",
  "passport_number": "12345678",
  "coordinates": "-12.10, -77.05",
  "isOffline": 0
}
```

## Campos relevantes

Campo	Tipo	Descripción
empleado	string	ID del empleado
image	string	URL base64 o imagen enviada
fecha_asistencia	date	Fecha de la marcación
type	string	“entrada” o “salida”
passport_number	string	Documento del empleado
coordinates	string	Lat/Lon opcional
isOffline	int	<input type="checkbox"/> si se marcó offline

## ? Seguridad

El servicio requiere:

- Token válido via middleware Laravel.
- Acceso interno al ERP mediante `general->insertERPNew()` y `general->dbErp()`.

## ? Flujo del Servicio (resumen real)

### 1. Validación de campos

Se ejecuta `verifyFields($request)` y si falla, retorna de inmediato.

## 2. Prevención de fechas inválidas

Si la marcación es online y la fecha es mayor al día actual → rechaza.

## 3. Validación de duplicidad

Se construye:

```
[  
  "employee" => empleado,  
  "log_type" => type_markings[type],  
  "fecha_de_consolidado" => fecha_asistencia  
]
```

Luego: `validateDuplicateDialing()`

Si ya existe marcación → error.

---

## 4. Obtención del turno del empleado

Consulta ERP:

- Hora de inicio (`start_time`)
- Hora de salida (`end_time`)
- Datos básicos del empleado

Si es “entrada” → valida que pase de PreActivo → Activo.

---

## 5. Cálculo de tardanza u horas extra

- Si el empleado marca **entrada** más de 5 min tarde → genera tardanza `"HH:MM"`.
  - Si marca **salida** con +60 minutos → horas extra.
- 

## 6. Registro en ERP

Crea documento:

**Employee Checkin**

```
[
  "employee" => empleado,
  "time" => fecha,
  "coordenadas" => coordinates,
  "log_type" => Entrada/Salida,
  "urlimagen2" => image,
  "offline" => isOffline,
  "late_time" => tardanza calculada,
  "overtime_time" => horasExtras,
]
```

---

## 7. Registro en base de datos interna

Inserta en tabla `marcaciones`:

```
employee, time, log_type, fecha_de_consolidado, urlimagen2, document...
```

---

## 8. Generación de asistencia

Si la marcación es **salida**, ejecuta: `storeAttendance($request)`

y devuelve datos adicionales.

---

## ? Response 200 – Ejemplo

? Entrada o salida registrada correctamente

```
{
  "valor": true,
  "msn": "Marcación registrada",
  "data": {
    "employee": "EMP-0001",
    "time": "2025-02-03 08:03:00",
    "late_time": "00:03",
    "overtime_time": "",
    "offline": 0
  }
}
```

? Al registrar salida (incluye asistencia generada)

```
{
  "valor": true,
  "msn": "Marcación registrada",
  "data": { ...checkin ERP... },
  "data2": { ...asistencia... }
}
```

# ? Posibles Errores

## 1. Campos inválidos

```
{
  "valor": false,
  "msn": "Falta completar campos obligatorios",
  "data": []
}
```

## 2. Marcación duplicada

```
{
  "valor": false,
  "msn": "Error al intentar realizar una marcación duplicada"
}
```

## 3. Fecha mayor al día actual

```
{
  "valor": false,
  "msn": "La marcacion no puede ser mayor al día actual."
}
```

## 4. Error creando Employee Checkin

```
{
  "valor": false,
  "msn": "Surgió un error al insertar marcación de salida.",
  "data": {}
}
```

## 5. Error servidor

```
{
  "valor": false,
  "msn": "Error del servidor: <mensaje>"
}
```

# ? Schemas utilizados

## Employee

```
{
  "name": string,
  "default_shift": string,
  "first_name": string,
  "first_last_name": string,
  "second_last_name": string,
  "passport_number": string,
  "id_sucursal": string
}
```

## Shift Type

```
{
  "name": string,
  "start_time": "HH:MM:SS",
  "end_time": "HH:MM:SS"
}
```

## Employee Checkin (ERP)

```
{
  "employee": string,
  "time": datetime,
  "log_type": string,
  "late_time": string,
  "overtime_time": string,
  "offline": int
}
```

# ? Lógica en Pseudo-código

```
verify = verifyFields(request)
if !verify.valor → return error

if fecha_asistencia > hoy AND online → return error

validateDuplicateDialing()
if duplicado → return error

turno = GET turno del empleado en ERP

if type == entrada:
    validarPreActive()

calcular tardanza / horas extra

bodyCheckin = {...}
insertERPNew("Employee Checkin", bodyCheckin)

insert DB marcaciones

if type == salida:
    storeAttendance()
    return respuesta con asistencia

return respuesta normal
```

Revisión #1

Creado 2025-11-25 15:51:55 -05 por Elian

Actualizado 2025-11-25 15:57:32 -05 por Elian