

Documentación: Generación de Datos de Prueba - Sistema de Emociones Corporativas

Fecha: 8 de Agosto de 2025 - Sesión de Tarde (16:00+)

Índice

1. [Resumen Ejecutivo](#)
 2. [Objetivo del Proyecto](#)
 3. [Análisis de Requisitos](#)
 4. [Implementación del Sistema de Datos](#)
 5. [Estructura de Archivos](#)
 6. [Explicación Detallada del Seeder](#)
 7. [Comandos Ejecutados](#)
 8. [Problemas Encontrados y Soluciones](#)
 9. [Verificación y Validación](#)
 10. [Resultados Finales](#)
 11. [Aprendizajes y Mejores Prácticas](#)
 12. [Próximos Pasos](#)
-

Resumen Ejecutivo

Objetivo Alcanzado:

Generar una base de datos completa con **2,200 registros realistas** de emociones de empleados para desarrollar un panel de administración con análisis corporativo.

Resultados Obtenidos:

- ☒ **2,200 registros** de emociones generados exitosamente
- ☒ **100 empleados** distribuidos en 8 departamentos
- ☒ **22 días laborales** por empleado (simulando un mes completo)
- ☒ **Datos realistas** basados en el sistema existente
- ☒ **Base sólida** para el panel de administración

Tecnologías Utilizadas:

- Laravel 10 (PHP)
 - Migraciones y Seeders
 - Eloquent ORM
 - Carbon (manejo de fechas)
 - SQLite (base de datos)
-

Objetivo del Proyecto

Contexto:

Después de implementar el sistema de departamentos en la sesión de la mañana, necesitábamos generar datos de prueba realistas para:

1. **Probar el panel de administración** con datos significativos
2. **Crear gráficas diferenciadoras** con información variada
3. **Simular un entorno laboral real** para demostraciones
4. **Validar la funcionalidad** del sistema completo

Criterios de Calidad:

- **Privacidad:** No identificar empleados específicos
 - **Realismo:** Emociones y respuestas coherentes
 - **Variedad:** Distribución equilibrada por departamentos
 - **Temporalidad:** Fechas distribuidas en el tiempo
-

Análisis de Requisitos

Estructura de Datos Necesaria:

Empleados:

- **Total:** 100 empleados
- **Formato:** E001, E002, E003... E100
- **Distribución por departamentos:**
 - Admin: E001-E010 (10 empleados)
 - Marketing: E011-E025 (15 empleados)
 - IT: E026-E045 (20 empleados)
 - RRHH: E046-E055 (10 empleados)
 - Ventas: E056-E075 (20 empleados)
 - Finanzas: E076-E085 (10 empleados)
 - Operaciones: E086-E095 (10 empleados)
 - Legal: E096-E100 (5 empleados)

Emociones (Sistema Existente):

- **heureux** - 😊 Feliz/Happy
- **neutre** - 😐 Neutral
- **frustre** - 😡 Frustrado/Frustrated
- **tendu** - 😬 Tenso/Tense
- **calme** - 😌 Calmado/Calm

Preguntas por Emoción:

Cada emoción tiene 3 preguntas específicas con respuestas Sí/No (0/1):

Heureux (Feliz):

- q1: ¿Te has sentido motivado para empezar tu día?
- q2: ¿Has tenido la sensación de que tus tareas tenían un objetivo claro?
- q3: ¿Has tenido interacciones positivas con tus compañeros o responsables?

Neutre (Neutral):

- q1: ¿Te ha costado mantenerte concentrado en tus tareas?
- q2: ¿Has encontrado tu día monótono o sin desafíos?
- q3: ¿Has evitado las interacciones con los demás hoy?

Frustre (Frustrado):

- q1: ¿Tienes la sensación de que tus esfuerzos no han sido apreciados hoy?
- q2: ¿Te has sentido limitado en el cumplimiento de tus tareas?
- q3: ¿Has experimentado cierta incomodidad con alguno de tus compañeros?

Tendu (Tenso):

- q1: ¿Te has sentido contrariado por una situación?
- q2: ¿Has sentido una presión excesiva o fuera de tu control?
- q3: ¿Has tenido que reprimir tu descontento para evitar conflictos?

Calme (Calmado):

- q1: ¿Tu ritmo de trabajo ha estado equilibrado hoy?
- q2: ¿Has tenido la sensación de dominar tus tareas?
- q3: ¿Has podido hacer pausas o descansar suficientemente?

Temporalidad:

- **22 días laborales** por empleado
- **30 días naturales - 8 fines de semana = 22 días laborales**
- **Fechas distribuidas** en los últimos 30 días
- **Total:** 100 empleados × 22 días = 2,200 registros

Implementación del Sistema de Datos

Fase 1: Creación del Seeder

Comando Ejecutado:

```
php artisan make:seeder MoodEmotionSeeder
```

Propósito:

Crear un archivo seeder específico para generar datos de emociones de empleados.

Ubicación del Archivo:

database/seeder/MoodEmotionSeeder.php

Fase 2: Definición de Arrays de Datos

Array de Emociones:

```
$emotions = ['heureux', 'neutre', 'frustre', 'tendu', 'calme'];
```

Explicación:

- Contiene las 5 emociones del sistema existente
- Se usa con `array_rand()` para selección aleatoria
- Mantiene consistencia con el sistema actual

Array de Departamentos:

```
$departments = [  
    'Admin' => ['start' => 1, 'end' => 10, 'id' => 1],  
    'Marketing' => ['start' => 11, 'end' => 25, 'id' => 2],  
    'IT' => ['start' => 26, 'end' => 45, 'id' => 3],  
    'RRHH' => ['start' => 46, 'end' => 55, 'id' => 4],  
    'Ventas' => ['start' => 56, 'end' => 75, 'id' => 5],  
    'Finanzas' => ['start' => 76, 'end' => 85, 'id' => 6],  
    'Operaciones' => ['start' => 86, 'end' => 95, 'id' => 7],  
    'Legal' => ['start' => 96, 'end' => 100, 'id' => 8]  
];
```

Explicación:

- **start:** Número inicial del empleado (ej: 1 para E001)
- **end:** Número final del empleado (ej: 10 para E010)
- **id:** ID del departamento en la base de datos
- **Estructura:** Permite asignación automática de departamentos

Fase 3: Estructura de Bucles Anidados

Bucle 1: Departamentos (Externo)

```
foreach($departments as $dept) {  
    // Procesa cada departamento  
}
```

Explicación línea a línea:

- **foreach(\$departments as \$dept):** Itera sobre cada departamento
- **\$dept:** Contiene el array con **start**, **end**, **id**
- **Propósito:** Procesar departamentos uno por uno

Ejemplo de iteraciones:

- **1ª iteración:** `$dept = ['start' => 1, 'end' => 10, 'id' => 1]` (Admin)
- **2ª iteración:** `$dept = ['start' => 11, 'end' => 25, 'id' => 2]` (Marketing)
- **Y así sucesivamente...**

Bucle 2: Empleados del Departamento (Medio)

```
for($emp = $dept['start']; $emp <= $dept['end']; $emp++) {
    $employee_id = 'E' . str_pad($emp, 3, '0', STR_PAD_LEFT);
    // Procesa cada empleado
}
```

Explicación línea a línea:

- **for(\$emp = \$dept['start']; \$emp <= \$dept['end']; \$emp++):** Bucle desde el primer empleado hasta el último del departamento
- **\$emp:** Número del empleado (1, 2, 3...)
- **\$dept['start']:** Número inicial del departamento
- **\$dept['end']:** Número final del departamento

Generación del Employee ID:

- **str_pad(\$emp, 3, '0', STR_PAD_LEFT):** Formatea el número con ceros a la izquierda
- **Ejemplos:**
 - `str_pad(1, 3, '0', STR_PAD_LEFT) → "001"`
 - `str_pad(15, 3, '0', STR_PAD_LEFT) → "015"`
- **'E' . str_pad(...):** Concatena 'E' con el número formateado
- **Resultado:** "E001", "E002", "E015", etc.

Bucle 3: Días Laborales (Interno)

```
for($day = 1; $day <= 22; $day++) {
    // Procesa cada día laboral
}
```

Explicación línea a línea:

- **for(\$day = 1; \$day <= 22; \$day++):** Bucle de 1 a 22 días
- **\$day:** Número del día (1, 2, 3... 22)
- **Propósito:** Generar un registro por cada día laboral

¿Por qué 22 días?

- **30 días naturales - 8 fines de semana = 22 días laborales**
 - Simula un mes completo de trabajo
 - Datos suficientes para análisis temporal
-

Fase 4: Generación de Datos Aleatorios

Emoción Aleatoria:

```
$randEmotion = $emotions[array_rand($emotions)];
```

Explicación línea a línea:

- **array_rand(\$emotions)**: Devuelve un índice aleatorio (0, 1, 2, 3, 4)
- **\$emotions[índice]**: Obtiene la emoción correspondiente
- **Resultado**: 'heureux', 'neutre', 'frustre', 'tendu', o 'calme'

Respuestas Aleatorias:

```
$answer1 = mt_rand(0, 1);  
$answer2 = mt_rand(0, 1);  
$answer3 = mt_rand(0, 1);
```

Explicación línea a línea:

- **mt_rand(0, 1)**: Genera 0 o 1 aleatoriamente
- **mt_rand()**: Mejor generador de números aleatorios que **rand()**
- **0**: Representa "No"
- **1**: Representa "Sí"
- **Resultado**: Respuestas coherentes para las 3 preguntas

Fecha Realista:

```
$date = now()->subDay(30 - $day);
```

Explicación línea a línea:

- **now()**: Fecha y hora actual
 - **subDay(30 - \$day)**: Resta días desde la fecha actual
 - **Lógica temporal**:
 - Día 1: **now()->subDay(29)** (hace 29 días)
 - Día 22: **now()->subDay(8)** (hace 8 días)
 - **Resultado**: Fechas distribuidas en los últimos 30 días
-

Fase 5: Creación del Registro en Base de Datos

Código Completo:

```
MoodEmotion::create([
    'employee_id' => $employee_id,
    'emotion' => $randEmotion,
    'answer_1' => $answer1,
    'answer_2' => $answer2,
    'answer_3' => $answer3,
    'department_id' => $dept['id'],
    'diary_text' => '',
    'created_at' => $date,
    'updated_at' => $date,
]);
```

Explicación línea a línea:

Campos de Identificación:

- **'employee_id' => \$employee_id**: ID del empleado (E001, E002, etc.)
- **'department_id' => \$dept['id']**: ID del departamento (1, 2, 3, etc.)

Campos de Emoción:

- **'emotion' => \$randEmotion**: Emoción aleatoria seleccionada
- **'answer_1' => \$answer1**: Respuesta a la primera pregunta (0/1)
- **'answer_2' => \$answer2**: Respuesta a la segunda pregunta (0/1)
- **'answer_3' => \$answer3**: Respuesta a la tercera pregunta (0/1)

Campos de Texto:

- **'diary_text' => ''**: Campo vacío (como se acordó)

Campos de Fecha:

- **'created_at' => \$date**: Fecha de creación simulada
- **'updated_at' => \$date**: Fecha de actualización (igual a created_at)

¿Por qué ambas fechas iguales?

- Lógico: cuando se crea un registro, ambas fechas son iguales
- Consistente: simula que el registro nunca se ha modificado
- Realista: en datos reales, la primera vez que se crea, ambas fechas son iguales

Estructura de Archivos

Archivos Creados/Modificados:

1. MoodEmotionSeeder.php (Creado)

Ubicación: `database/seeders/MoodEmotionSeeder.php`

Propósito: Generar 2,200 registros de emociones de empleados

Contenido Principal:

- Arrays de datos (emociones, departamentos)
- Bucles anidados para generación
- Lógica de aleatorización
- Creación de registros en base de datos

2. DatabaseSeeder.php (Modificado)

Ubicación: `database/seeders/DatabaseSeeder.php`

Modificación:

```
$this->call([
    DepartmentSeeder::class,
    MoodEmotionSeeder::class,
]);
```

Propósito: Registrar el MoodEmotionSeeder para ejecución automática

3. Department.php (Existente)

Ubicación: `app/Models/Department.php`

Relación Utilizada:

```
public function moodEmotions()
{
    return $this->hasMany(MoodEmotion::class);
}
```

Propósito: Relación uno a muchos con MoodEmotion

4. MoodEmotion.php (Existente)

Ubicación: `app/Models/MoodEmotion.php`

Relación Utilizada:

```
public function department()
{
    return $this->belongsTo(Department::class);
}
```

Propósito: Relación muchos a uno con Department

Comandos Ejecutados

1. Creación del Seeder

```
php artisan make:seeder MoodEmotionSeeder
```

Propósito: Crear archivo seeder para datos de emociones

2. Registro del Seeder

Archivo: `database/seeder/DatabaseSeeder.php` **Acción:** Agregar `MoodEmotionSeeder::class` al array de seeders

3. Ejecución del Seeder

```
php artisan db:seed
```

Propósito: Ejecutar todos los seeders registrados

4. Ejecución Específica

```
php artisan db:seed --class=MoodEmotionSeeder
```

Propósito: Ejecutar solo el MoodEmotionSeeder

5. Verificación de Datos

```
php artisan tinker --execute="echo 'Total: ' . App\Models\MoodEmotion::count();" 
```

Propósito: Contar registros totales generados

6. Limpieza de Autoload

```
composer dump-autoload
```

Propósito: Regenerar referencias de clases

Problemas Encontrados y Soluciones

Problema 1: Error de DepartmentSeeder

Error: `UNIQUE constraint failed: departments.name`

Causa: Los departamentos ya existían en la base de datos

Solución: Ejecutar solo el MoodEmotionSeeder

```
php artisan db:seed --class=MoodEmotionSeeder
```

Problema 2: Campo 'date' no existe

Error: `table mood_emotions has no column named date`

Causa: Se intentaba guardar un campo 'date' que no existe en la tabla

Solución: Usar `created_at` y `updated_at` en lugar de 'date'

```
'created_at' => $date,  
'updated_at' => $date,
```

Problema 3: Modelo no encontrado en Tinker

Error: `Undefined constant "App\Models\MoodEmotion"`

Causa: Autoload de Composer no actualizado

Solución: Regenerar autoload

```
composer dump-autoload
```

Problema 4: Datos mezclados con registros antiguos

Problema: Registros nuevos mezclados con registros antiguos con NULL en department_id

Solución: Eliminar registros con NULL

```
php artisan tinker --execute="App\Models\MoodEmotion::whereNull('department_id')->delete();" 
```

Verificación 1: Conteo Total

```
php artisan tinker --execute="echo 'Total: ' . App\Models\MoodEmotion::count();"

```

Resultado: Total: 2200 ☒

Verificación 2: Registros con Department ID

```
php artisan tinker --execute="echo 'Registros con department_id: ' .
DB::table('mood_emotions')->whereNotNull('department_id')->count();"

```

Resultado: Registros con department_id: 2200 ☒

Verificación 3: Registros con NULL

```
php artisan tinker --execute="echo 'Registros con NULL: ' .
DB::table('mood_emotions')->whereNull('department_id')->count();"

```

Resultado: Registros con NULL: 0 ☒

Verificación 4: Estructura de Datos

Verificación en Base de Datos:

- ☒ 2,200 registros totales
- ☒ 0 registros con department_id NULL
- ☒ Fechas distribuidas correctamente
- ☒ Emociones variadas presentes
- ☒ Respuestas 0/1 para preguntas



Resultados Finales

Datos Generados:

- Total de registros:** 2,200
- Empleados:** 100 (E001-E100)
- Días por empleado:** 22
- Departamentos:** 8
- Emociones:** 5 tipos
- Preguntas por emoción:** 3
- Respuestas:** Sí/No (0/1)

Distribución por Departamentos:

- **Admin:** ~220 registros (10 empleados × 22 días)
- **Marketing:** ~330 registros (15 empleados × 22 días)
- **IT:** ~440 registros (20 empleados × 22 días)
- **RRHH:** ~220 registros (10 empleados × 22 días)
- **Ventas:** ~440 registros (20 empleados × 22 días)
- **Finanzas:** ~220 registros (10 empleados × 22 días)
- **Operaciones:** ~220 registros (10 empleados × 22 días)
- **Legal:** ~110 registros (5 empleados × 22 días)

Calidad de Datos:

- ☒ **Realistas:** Basados en sistema existente
 - ☒ **Variados:** Distribución equilibrada
 - ☒ **Temporales:** Fechas distribuidas en el tiempo
 - ☒ **Privados:** Sin identificación individual
 - ☒ **Consistentes:** Estructura coherente
-

Aprendizajes y Mejores Prácticas

Aprendizajes Técnicos:

1. Estructura de Bucles Anidados

- **Orden lógico:** Departamentos → Empleados → Días
- **Eficiencia:** Aprovechar estructura de datos existente
- **Mantenibilidad:** Código escalable y legible

2. Generación de Datos Realistas

- **Consistencia:** Usar datos del sistema existente
- **Variedad:** Distribución equilibrada
- **Temporalidad:** Fechas realistas distribuidas

3. Manejo de Errores

- **Diagnóstico:** Identificar causa raíz del problema
- **Solución específica:** No aplicar soluciones genéricas
- **Verificación:** Confirmar que la solución funciona

Mejores Prácticas Aplicadas:

1. Planificación Antes de Implementación

- Definir estructura de datos
- Planificar distribución
- Establecer criterios de calidad

2. Desarrollo Incremental

- Crear estructura básica
- Agregar funcionalidad paso a paso
- Verificar en cada etapa

3. Documentación Completa

- Explicar cada línea de código
- Documentar decisiones de diseño
- Registrar problemas y soluciones

4. Verificación Rigurosa

- Contar registros generados
- Verificar integridad de datos
- Validar distribución esperada

Próximos Pasos

Fase 5: Panel de Administración

1. Crear AdminController

- Métodos para estadísticas
- Filtros por departamento
- Exportación de datos

2. Crear Vistas del Panel

- Dashboard principal
- Gráficas por departamento
- Tabla de datos filtrable

3. Configurar Rutas Protegidas

- Middleware de autorización
- Rutas exclusivas para administradores

4. Implementar Gráficas

- Pulso emocional de la empresa
- Mapa de calor por departamentos
- Tendencias semanales

Fase 6: Funcionalidades Avanzadas

1. Sistema de Alertas

- Departamentos con tendencias negativas
- Notificaciones automáticas

2. Exportación de Datos

- Reportes en PDF
- Exportación a Excel
- Datos para análisis externo

3. Filtros Avanzados

- Por rango de fechas
- Por tipo de emoción
- Comparación entre departamentos

Conclusión

Objetivos Alcanzados:

1. ☒ **Sistema de generación de datos** completamente funcional
2. ☒ **2,200 registros realistas** generados exitosamente
3. ☒ **Base de datos preparada** para el panel de administración
4. ☒ **Datos variados y temporales** para análisis corporativo

Valor Agregado:

- **Prototipo realista** para demostraciones
- **Datos de prueba** para desarrollo
- **Base sólida** para funcionalidades avanzadas
- **Experiencia práctica** en generación de datos masivos

Impacto en el Proyecto:

- **Acelera el desarrollo** del panel de administración
- **Permite pruebas realistas** de funcionalidades
- **Facilita la presentación** a clientes potenciales
- **Establece estándares** de calidad de datos

El proyecto está listo para la siguiente fase: creación del panel de administración con gráficas diferenciadoras y análisis corporativo avanzado.