

# Arquitectura de la Base de Datos Central: El Corazón del Sistema

Este documento describe la estructura y el funcionamiento de la base de datos central que unificará el "Panel de Administración" y la "Aplicación del Empleado". Esta base de datos es el componente que permite que los cambios realizados por la gerencia se reflejen en tiempo real en los teléfonos de los empleados y que los registros de actividad de los empleados estén disponibles instantáneamente para su análisis.

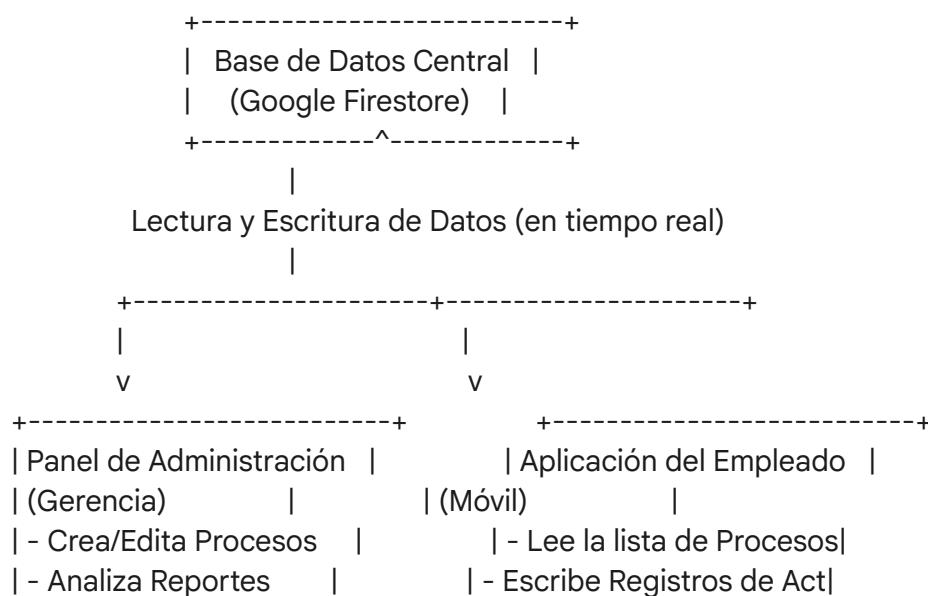
## La Tecnología Elegida: Google Firestore

Para nuestra base de datos, utilizaremos **Google Firestore**. Es una base de datos en la nube moderna, segura y en tiempo real.

- **En Tiempo Real:** Cuando un gerente crea una nueva tarea, aparece instantáneamente en la app de los empleados sin que tengan que recargarla.
- **Escalable:** Crece junto con tu negocio sin que tengas que preocuparte por la infraestructura.
- **Segura:** Permite definir reglas de acceso para que los empleados solo puedan ver y registrar su propia información.
- **Sin Servidor (Serverless):** Reduce la complejidad y los costos de mantenimiento.

## Diagrama de Flujo de Datos

El concepto es simple: las dos aplicaciones que hemos creado dejarán de usar el almacenamiento local (`localStorage`) y pasarán a leer y escribir directamente en Firestore.



+-----+	+-----+
---------	---------

## Estructura de la Base de Datos (Las "Colecciones")

En Firestore, los datos se organizan en "colecciones" de "documentos". Aquí está la estructura que usaremos:

### 1. Colección: employees

Almacena la información de cada empleado.

- **Ejemplo de un documento:**

```
{
  "id": "1",
  "name": "JUAN LAI",
  "cedula": "V-18059138-5",
  "is_active": true
}
```

### 2. Colección: processes

Contiene la definición de todos los procesos principales que se configuran en el Panel de Administración.

- **Ejemplo de un documento:**

```
{
  "id": "proc_001",
  "name": "Reparación de Impresora",
  "is_composite": true,
  "measurement_type": "Unitario",
  "standard_time_per_unit_seconds": 0
}
```

### 3. Colección: subprocesses

Contiene la definición de todos los subprocesos.

- **Ejemplo de un documento:**

```
{
  "id": "sub_201",
  "name": "Perforación de Codos",
  "measurement_type": "Cantidad",
  "standard_time_per_unit_seconds": 45
}
```

*Note: La relación entre processes y subprocesses se guardará dentro del documento del proceso compuesto.*

## 4. Colección: activity\_logs

Esta es la colección más importante. Aquí se guardará cada registro de trabajo de cada empleado. Crecerá constantemente.

- Ejemplo de un documento:

```
{  
  "log_id": "log_xyz789",  
  "employee_id": "1",  
  "process_id": "proc_001",  
  "start_time": "2025-10-15T14:30:00Z", // Formato estándar de fecha  
  "end_time": "2025-10-15T15:15:45Z",  
  "duration_seconds": 2745,  
  "reference_note": "P-2025-142",  
  "completed_subprocesses": [  
    { "subprocess_id": "sub_202", "quantity": 1 }, // Cambio de Cabezal  
    { "subprocess_id": "sub_302", "quantity": 4 } // Perforó 4 codos  
  ]  
}
```

## 5. Colección: config

Una colección especial con un único documento para guardar la configuración general del sistema.

- El único documento (system\_settings):

```
{  
  "bonus_good_threshold": 36000, // Puntos para el bono "Bueno"  
  "bonus_good_amount_usd": 5,  
  "bonus_excellent_threshold": 48000, // Puntos para el bono "Sobresaliente"  
  "bonus_excellent_amount_usd": 10  
}
```

## Próximos Pasos

El siguiente paso técnico es modificar el código de las dos aplicaciones (sistema\_productividad.html y admin\_panel.html) para:

1. Integrar el SDK de Firebase (la herramienta para conectar con Firestore).
2. Reemplazar toda la lógica que usa localStorage para que, en su lugar, lea y escriba en

estas colecciones de Firestore.

Una vez hecho esto, tu sistema será una aplicación web real, centralizada y lista para ser usada por todo tu equipo simultáneamente.