

Sistema de Asignación Inteligente de Proveedores

🎯 Descripción General

El Sistema de Asignación Inteligente de Proveedores es un algoritmo avanzado que recomienda automáticamente los mejores proveedores para cada orden de trabajo basado en múltiples criterios de evaluación.

Características Principales

- **Algoritmo de Scoring Multi-Factor:** Evalúa 6 dimensiones diferentes
- **Recomendaciones en Tiempo Real:** Cálculos instantáneos basados en datos actualizados
- **Análisis de Rendimiento:** Métricas detalladas de cada proveedor
- **Dashboard de Estadísticas:** Visualización del rendimiento del sistema
- **API REST Completa:** Integración fácil con cualquier frontend

📊 Factores de Evaluación

El algoritmo evalúa cada proveedor en 6 dimensiones, con un puntaje máximo de 100 puntos:

1. Rating (0-25 puntos)

- Basado en el rating histórico del proveedor (escala 1-5)
- **Fórmula:** $(\text{rating} / 5) * 25$
- **Peso:** 25% del score total

2. Disponibilidad (0-20 puntos)

- Verifica la disponibilidad actual del proveedor
- **Estados:**
 - Disponible: 20 puntos
 - Parcialmente disponible: 12 puntos
 - No disponible: 0 puntos
- **Peso:** 20% del score total
- **Boost:** x1.5 para trabajos urgentes

3. Especialización (0-15 puntos)

- Coincidencia entre el tipo de trabajo y la especialización del proveedor
- **Puntuación:**
 - Coincidencia exacta: 15 puntos
 - Sin coincidencia: 5 puntos
- **Peso:** 15% del score total

4. Carga de Trabajo (0-15 puntos)

- Evalúa la carga actual de trabajos pendientes

- **Escala:**

- 0 trabajos: 15 puntos
- 1 trabajo: 13 puntos
- 2 trabajos: 11 puntos
- 3 trabajos: 9 puntos
- 4-5 trabajos: 6 puntos
- 6-7 trabajos: 3 puntos
- 8+ trabajos: 0 puntos
- **Peso:** 15% del score total

5. Rendimiento Histórico (0-15 puntos)

- Analiza el rendimiento pasado del proveedor

- **Componentes:**

- Tasa de finalización a tiempo: 7 puntos
- Calidad del trabajo (valoraciones): 5 puntos
- Tendencia reciente: 3 puntos
- **Peso:** 15% del score total

6. Tiempo de Respuesta (0-10 puntos)

- Velocidad promedio de respuesta del proveedor

- **Escala:**

- < 1 hora: 10 puntos
- < 2 horas: 9 puntos
- < 4 horas: 7 puntos
- < 8 horas: 5 puntos
- < 24 horas: 3 puntos
- 24+ horas: 1 punto
- **Peso:** 10% del score total
- **Boost:** x1.5 para trabajos urgentes



Uso del Sistema

API Endpoints

1. Obtener Recomendaciones

```
POST /api/providers/recommend
Content-Type: application/json
```

```
{
  "buildingId": "clxxx...",
  "tipo": "plomeria",
  "prioridad": "alta",
  "presupuestoMax": 5000,
  "fechaRequerida": "2024-12-10T00:00:00Z",
  "limit": 5
}
```

Respuesta:

```
{
  "success": true,
  "recommendations": [
    {
      "provider": {
        "id": "clxxx...",
        "nombre": "Plomería Rápida SL",
        "tipo": "Plomería",
        "telefono": "+34 600 123 456",
        "email": "contacto@plomeria.com",
        "rating": 4.8
      },
      "score": {
        "total": 92.5,
        "breakdown": {
          "rating": 24.0,
          "availability": 20.0,
          "specialization": 15.0,
          "workload": 15.0,
          "performance": 13.5,
          "responseTime": 5.0
        }
      },
      "reasoning": [
        "Excelente rating: 4.8/5.0",
        "Disponible inmediatamente",
        "Especializado en Plomería",
        "Sin trabajos pendientes",
        "Excelente puntualidad (95% a tiempo)",
        "Respuesta dentro del día"
      ],
      "recommendation": "Altamente Recomendado"
    }
  ],
  "metadata": {
    "timestamp": "2024-12-04T10:30:00Z",
    "criteria": {
      "tipo": "plomeria",
      "prioridad": "alta"
    }
  }
}
```

2. Métricas de Rendimiento

```
GET /api/providers/performance/{providerId}?days=90
```

Respuesta:

```
{
  "success": true,
  "providerId": "clxxx...",
  "period": {
    "days": 90,
    "from": "2024-09-04T00:00:00Z",
    "to": "2024-12-04T00:00:00Z"
  },
  "metrics": {
    "completion": {
      "avgTime": 3.5,
      "rate": 100.0,
      "onTimeRate": 95.0
    },
    "workload": {
      "completed": 25,
      "pending": 2
    },
    "quality": {
      "avgRating": 4.8,
      "trend": "improving"
    },
    "responsiveness": {
      "avgResponseTime": 4.5
    }
  },
  "analysis": [
    "Excelente historial de puntualidad en entregas",
    "Calidad de trabajo excepcional según clientes",
    "Tendencia positiva: mejora continua en el rendimiento",
    "Carga de trabajo ligera - buena disponibilidad",
    "Proveedor con buena experiencia"
  ]
}
```

3. Estadísticas Generales

`GET /api/providers/stats`

Respuesta:

```
{
  "success": true,
  "stats": {
    "totalProviders": 15,
    "totalWorkOrders": 142,
    "completedOrders": 128,
    "completionRate": 90.14,
    "avgProviderRating": 4.5
  }
}
```

Componentes React

1. ProviderRecommendations

Componente para mostrar recomendaciones en un formulario de creación de orden:

```
import ProviderRecommendations from '@/app/components/ProviderRecommendations';

function CreateWorkOrder() {
  const [selectedProviderId, setSelectedProviderId] = useState<string>(null);

  return (
    <div>
      {/* Otros campos del formulario */}

      <ProviderRecommendations
        buildingId="clxxx..."
        tipo="plomeria"
        prioridad="alta"
        presupuestoMax={5000}
        fechaRequerida={new Date()}
        onSelectProvider={(id, provider) => {
          setSelectedProviderId(id);
          // Hacer algo con el proveedor seleccionado
        }}
        selectedProviderId={selectedProviderId}
      />
    </div>
  );
}
```

2. ProviderAssignmentDashboard

Dashboard de estadísticas:

```
import ProviderAssignmentDashboard from '@/app/components/ProviderAssignmentDashboard';

function AdminDashboard() {
  return (
    <div>
      <ProviderAssignmentDashboard />
      {/* Otros componentes */}
    </div>
  );
}
```

Hooks Personalizados

useProviderRecommendations

```
import { useProviderRecommendations } from '@/app/hooks/useProviderRecommendations';

function MyComponent() {
  const { recommendations, loading, error, fetchRecommendations } = useProviderRecommendations();

  const handleGetRecommendations = async () => {
    try {
      const recs = await fetchRecommendations({
        buildingId: 'clxxx...',
        tipo: 'electricidad',
        prioridad: 'urgente',
        limit: 5,
      });
      console.log('Recomendaciones:', recs);
    } catch (err) {
      console.error('Error:', err);
    }
  };

  return (
    <div>
      <button onClick={handleGetRecommendations}>Obtener Recomendaciones</button>
      {loading && <p>Cargando...</p>}
      {error && <p>Error: {error}</p>}
      {recommendations.map((rec) => (
        <div key={rec.provider.id}>
          {rec.provider.nombre} - Score: {rec.score.total}
        </div>
      ))}
    </div>
  );
}
```

useProviderPerformance

```
import { useProviderPerformance } from '@/app/hooks/useProviderRecommendations';

function ProviderDetail({ providerId }: { providerId: string }) {
  const { performance, loading, fetchPerformance } = useProviderPerformance();

  useEffect(() => {
    fetchPerformance(providerId, 90);
  }, [providerId]);

  if (loading) return <div>Cargando métricas...</div>;
  if (!performance) return null;

  return (
    <div>
      <h3>Métricas de Rendimiento</h3>
      <p>Rating Promedio: {performance.metrics.quality.avgRating}</p>
      <p>Tasa de Puntualidad: {performance.metrics.completion.onTimeRate}%</p>
      <p>Trabajos Completados: {performance.metrics.workload.completed}</p>
      <p>Tendencia: {performance.metrics.quality.trend}</p>
    </div>
  );
}
```

Interpretación de Resultados

Niveles de Recomendación

Score	Nivel	Descripción
85-100	Altamente Recomendado	Proveedor excepcional, ideal para el trabajo
70-84	Recomendado	Buen candidato, confiable y competente
55-69	Aceptable	Puede realizar el trabajo, monitoreo recomendado
40-54	Considerar con Cautela	Factores de riesgo presentes, evaluar alternativas
0-39	No Recomendado	No cumple criterios mínimos, buscar otros proveedores

Análisis de Razonamiento

Cada recomendación incluye un array de `reasoning` con explicaciones:

-  **Verde:** Aspectos positivos
-  **Amarillo:** Aspectos aceptables con margen de mejora

- **Rojo**: Aspectos negativos o de riesgo
 - **Verde**: Tendencia de mejora
 - **Rojo**: Tendencia de declive
 - **Azul**: Respuesta muy rápida
 - **Amarillo**: Rating destacado
 - **Rojo**: Urgencia detectada
-

Configuración Avanzada

Ajustar Pesos de Factores

Para modificar los pesos de los factores, edita `lib/provider-assignment-service.ts`:

```
// Ejemplo: dar más peso al rating
breakdown.rating = (providerRating / 5) * 30; // cambiar de 25 a 30

// Ajustar otros factores proporcionalmente
breakdown.availability *= 0.8; // reducir peso
```

Personalizar lógica de prioridad

```
if (criteria.prioridad === 'urgente') {
    // Personalizar boosts
    breakdown.availability *= 2.0; // duplicar peso
    breakdown.responseTime *= 2.0;
}
```

Añadir nuevos factores

1. Añadir campo en `ProviderScore.breakdown`
 2. Implementar lógica de cálculo en `calculateProviderScore`
 3. Actualizar API response y componentes
-

Métricas de Rendimiento

El sistema calcula automáticamente:

Métricas de Completado

- **avgCompletionTime**: Tiempo promedio de finalización (días)
- **completionRate**: Porcentaje de trabajos completados
- **onTimeRate**: Porcentaje de trabajos finalizados a tiempo

Métricas de Carga

- **totalJobsCompleted**: Total de trabajos completados
- **totalJobsPending**: Total de trabajos pendientes

Métricas de Calidad

- **avgCustomerRating:** Rating promedio de clientes
- **recentTrend:** Tendencia reciente (improving/stable/declining)

Métricas de Respuesta

- **avgResponseTime:** Tiempo promedio de respuesta (horas)



Mantenimiento

Logs del Sistema

Todos los eventos importantes se registran:

```
logger.info('Recomendaciones generadas', {
    total: recommendations.length,
    topScore: recommendations[0]?.totalScore,
});
```

Optimizaciones

1. **Caché:** Considera implementar caché para proveedores frecuentes
2. **Batch Processing:** Para múltiples recomendaciones simultáneas
3. **ML Training:** Registrar asignaciones exitosas para mejorar el algoritmo



Roadmap Futuro

- [] Machine Learning para predicciones más precisas
- [] Análisis de proximidad geográfica
- [] Integración con calendario de proveedores
- [] Recomendaciones basadas en historial de colaboraciones
- [] Sistema de notificaciones automáticas
- [] A/B testing de diferentes algoritmos
- [] Dashboard de BI avanzado

? FAQ

P: ¿Cómo maneja el sistema trabajos sin proveedores especializados?

R: El sistema primero busca proveedores especializados. Si no encuentra ninguno, busca en todos los proveedores activos y penaliza el score de especialización.

P: ¿Se pueden ignorar ciertos factores?

R: Sí, puedes ajustar los pesos en el servicio o crear una versión personalizada del algoritmo.

P: ¿Cómo se calculan las tendencias?

R: Comparando el rendimiento de los últimos 30 días vs los 60 días anteriores.

P: ¿El sistema aprende de las asignaciones pasadas?

R: Actualmente no, pero hay un placeholder para implementar ML en futuras versiones.

Soporte

Para reportar bugs o sugerir mejoras:

- Crear un issue en el repositorio
 - Contactar al equipo de desarrollo
 - Revisar la documentación adicional en [/docs](#)
-

Última actualización: Diciembre 2024

Versión: 1.0.0