



Guía de Coordinación del Proyecto



Rutas Adicionales Implementadas



Rutas de Verificación de Email (Samuel)

```
// Nuevas rutas añadidas al módulo de autenticación
POST /api/auth/verify-email           // Verificar email con token
POST /api/auth/resend-verification    // Reenviar email de verificación
PUT  /api/auth/update-email           // Actualizar email (requiere
verificación)
```



Rutas de Upload de Imágenes (Franklin)

```
// Nuevas rutas para manejo de archivos
POST /api/upload/avatar               // Subir avatar de usuario
POST /api/upload/document             // Subir documentos del proyecto
DELETE /api/upload/:filename          // Eliminar archivo
GET  /api/upload/:filename            // Obtener archivo
```



Flujo de Integración

Fase 1: Configuración Base (Semana 1-2)

Responsable: Samuel

- [] Configurar repositorio Git
- [] Estructura inicial del proyecto
- [] Configuración de base de datos
- [] Sistema de autenticación básico
- [] Middleware de seguridad

Fase 2: Módulos Independientes (Semana 3-5)

Paralelo entre todos:

- **Samuel:** Completar usuarios y roles
- **Franklin:** Categorías, estados y uploads
- **Mariana:** Iniciar proyectos (mockear dependencias)
- **Mariam:** Iniciar tareas (mockear dependencias)

Fase 3: Integración (Semana 6-7)

Todos juntos:

- [] Integrar todos los módulos
- [] Resolver conflictos de dependencias
- [] Testing de integración
- [] Documentación de API

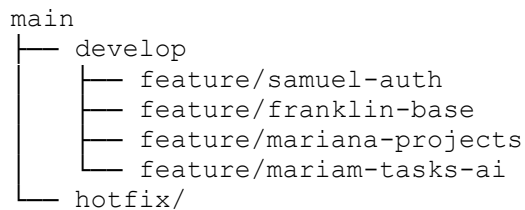
Fase 4: IA y Optimización (Semana 8-9)

Mariam con apoyo de todos:

- [] Implementar funcionalidades de IA
- [] Optimizar consultas de base de datos
- [] Testing completo
- [] Documentación final



Sistema de Branches



Reglas de Git:

1. **No push directo a main**
2. **Pull requests obligatorios**
3. **Code review por otro desarrollador**
4. **Tests deben pasar antes del merge**



Estrategia de Testing

Responsabilidades de Testing:

- **Samuel:** Tests de autenticación y usuarios
- **Franklin:** Tests de categorías y uploads
- **Mariana:** Tests de proyectos
- **Mariam:** Tests de tareas e IA

Tipos de Tests:

- **Unit Tests:** Cada desarrollador para sus módulos
- **Integration Tests:** En conjunto durante fase de integración
- **E2E Tests:** Al final del proyecto

Documentación Compartida

Colección de Postman (Compartida):

```
{
  "info": {
    "name": "Project Management API",
    "description": "API completa para gestión de proyectos y tareas"
  },
  "auth": {
    "type": "bearer",
    "bearer": [{"key": "token", "value": "{{jwt_token}}"}]
  }
}
```

Variables de Entorno Postman:

- `base_url`: `http://localhost:3000/api`
- `jwt_token`: (se actualiza automáticamente)
- `project_id`: (para tests de proyectos)
- `task_id`: (para tests de tareas)

Convenciones de Código

Nomenclatura:

- **Archivos**: camelCase.js
- **Rutas**: kebab-case
- **Variables**: camelCase
- **Constantes**: UPPER_SNAKE_CASE
- **Modelos**: PascalCase

Estructura de Respuestas:

```
// Respuesta exitosa
{
  "success": true,
  "message": "Operación exitosa",
  "data": {},
  "pagination": {} // si aplica
}

// Respuesta de error
{
  "success": false,
  "message": "Mensaje de error",
  "errors": [], // detalles específicos
  "code": "ERROR_CODE"
}
```

Utilidades Compartidas

Middleware de Validación (Franklin):

```
// validation.middleware.js
const validateRequest = (validations) => {
  return async (req, res, next) => {
    // Lógica de validación compartida
  };
};
```

Utilidad de Respuesta (Franklin):

```
// response.util.js
const sendResponse = (res, statusCode, success, message, data = null) => {
  {
    return res.status(statusCode).json({
      success,
      message,
      data,
      timestamp: new Date().toISOString()
    });
  }
};
```

Checklist Final de Cada Desarrollador

Antes de hacer Pull Request:

- ☐ Código comentado y limpio
- ☐ Tests unitarios escritos y pasando
- ☐ Validaciones implementadas
- ☐ Manejo de errores implementado
- ☐ Documentación de endpoints actualizada
- ☐ Variables de entorno documentadas
- ☐ Middleware de autorización implementado

Antes del Deploy:

- ☐ Todos los tests pasando
- ☐ Base de datos con datos de prueba
- ☐ Documentación completa
- ☐ Variables de producción configuradas
- ☐ Rate limiting configurado
- ☐ Logs implementados

Métricas de Calidad

Objetivos:

- **Cobertura de tests:** > 80%
- **Tiempo de respuesta:** < 200ms promedio
- **Disponibilidad:** > 99%
- **Seguridad:** Sin vulnerabilidades críticas

Herramientas de Monitoreo:

- **Performance:** New Relic / DataDog
- **Logs:** Winston + ELK Stack
- **Errores:** Sentry
- **API Docs:** Swagger/OpenAPI



Contactos del Equipo

- **Samuel Gomez:** samuel@email.com (Auth & Users)
- **Franklin Peña:** franklin@email.com (Base & States)
- **Mariana Gomez:** mariana@email.com (Projects)
- **Mariam Pizza:** mariam@email.com (Tasks & AI)

Horarios de Desarrollo:

- **Daily Standup:** 9:00 AM
- **Code Review:** 2:00 PM - 4:00 PM
- **Integration Testing:** Viernes 3:00 PM