



Sprint 2

Estudiosos:

Santiago Castro González

Anderson Yecid Hurtado Puentes

Victor Alfonso Alarcón Ospina

ARQUITECTURA DEL SOFTWARE

Ingeniería en software

Universidad Manuela Beltrán

13/2/2025

1. Revisión del Sprint Anterior

Estrategias que Funcionaron Bien

- División del trabajo por módulos permitió avanzar de manera paralela (ej. RRHH, Proveedores, Clientes).
- Uso de GitHub facilitó el control de versiones y la colaboración en tiempo real.
- Reuniones cortas y frecuentes (dailys) ayudaron a mantener sincronización entre los miembros.
- Documentación previa con arc42 evitó dudas al estructurar los componentes clave del sistema.
- Implementación de Django + React + PostgreSQL permitió desarrollo modular y pruebas rápidas.

Aspectos a Mejorar

A nivel grupal:

- Coordinación en la integración de los módulos fue tardía (hubo conflictos al unir RRHH).
- Falta de pruebas automatizadas en los formularios de proveedores y servicios.
- Tareas mal estimadas: algunas historias tomaron más tiempo del previsto.

A nivel individual:

- Algunos miembros enfrentaron dificultades técnicas con Django y PostgreSQL.
- Hubo retrasos por desconocimiento de cómo configurar base de datos.

Cuellos de Botella o Problemas

- Problemas de conexión entre frameworks cuando se probaba el despliegue local.
- Retrasos en el desarrollo del módulo de Clientes debido a validaciones incompletas en los formularios.
- Problemas de comunicación al inicio con respecto a nomenclatura y estructura de carpetas del backend.

2. Planificación del Sprint 2

Objetivo del Sprint 2 (Sprint Goal)

Desarrollar y consolidar funcionalidades críticas del ERP Logístico en los módulos de **Proveedores** y **Clientes**, incluyendo la conexión con la base de datos, validaciones,

reportes y pruebas. Además, integrar parcialmente los módulos de Contabilidad e Inventory para habilitar flujos transversales como órdenes de compra.

Historias de Usuario

ID	Historia de Usuario	Rol	Necesidad	Resultado Esperado
HU-05	Como administrador quiero registrar un proveedor para gestionar órdenes de compra.	Admin	Registrar proveedor con datos básicos y de contacto.	Proveedor guardado y visible.
HU-06	Como administrador quiero ver la lista de proveedores y editar su información.	Admin	Consultar proveedores y actualizarlos.	Lista editable funcional.
HU-07	Como cliente quiero registrarme para poder solicitar servicios.	Cliente	Registrar mis datos para futuros pedidos.	Cliente creado con validación.
HU-08	Como administrador quiero generar una orden de compra con un proveedor activo.	Admin	Asociar proveedor a productos e iniciar orden.	Orden creada y enviada.
HU-09	Como operador de inventario quiero registrar entradas/salidas de productos.	Operador	Mantener actualizado el stock.	Movimiento almacenado y visible.

Sprint Backlog (Historias, Tareas, Criterios de Aceptación)

Ejemplo: HU-05 - Registro de Proveedores

HU	Tarea	Responsable	Inicio	Fin	Criterio de Aceptación
HU-05	Crear modelo Proveedor en Django	Santiago	06/05	06/05	El modelo contiene campos validados correctamente.
HU-05	Diseñar formulario web de registro	Anderson	06/05	07/05	El formulario muestra todos los campos requeridos.
HU-05	Conectar formulario con API REST	Víctor	07/05	08/05	Se crea el proveedor y muestra mensaje de éxito.
HU-05	Prueba funcional del registro	Todos	08/05	09/05	Se realizan 3 pruebas positivas y 2 negativas.
HU-06	Crear vista de lista de proveedores	Víctor	06/05	06/05	La vista permite visualizar todos los registros.
HU-06	Agregar funcionalidad de edición de proveedor	Anderson	07/05	08/05	La edición actualiza la base de datos sin errores.
HU-07	Crear formulario de cliente	Santiago	06/05	07/05	El formulario crea un cliente nuevo con validación.

HU-08	Diseñar módulo de orden de compra	Anderson	07/05	08/05	Orden vinculada a proveedor y guardada correctamente.
HU-09	Registrar movimientos de inventario	Víctor	06/05	07/05	Entradas/salidas registradas y visibles en reportes.

3. Atributos de Calidad, Escenarios y Tácticas

A continuación, te presento una tabla con 4 atributos de calidad relevantes para el ERP Logístico de Eventos, cada uno con un escenario y una táctica aplicada:

Atributo de Calidad	Escenario	Táctica Aplicada
Seguridad	Cuando un usuario inicia sesión, el sistema debe validar credenciales cifradas y proteger contra ataques (XSS, CSRF, inyección SQL).	Autenticación basada en roles, hashing PBKDF2, middleware CSRF.
Rendimiento	Al consultar una orden o registrar un proveedor, la respuesta del sistema debe ser menor a 2 segundos bajo carga normal.	Optimización de consultas SQL, uso de índices, caché de consultas.
Mantenibilidad	Al modificar el módulo de proveedores, los demás módulos no deben verse afectados.	Separación de módulos en apps Django, principios SOLID, pruebas unitarias.
Disponibilidad	En caso de caída parcial del sistema, el usuario debe poder acceder a módulos no afectados.	Despliegue distribuido en AWS con balanceador de carga y monitoreo.

4. Planificación de Pruebas Funcionales

Durante este segundo Sprint se definieron cinco Historias de Usuario prioritarias (HU-05 a HU-09), enfocadas en funcionalidades claves del sistema ERP Logístico. Para garantizar su correcta implementación y calidad, se diseñó un conjunto de pruebas funcionales por cada HU.

Objetivo de las Pruebas

Validar que cada funcionalidad desarrollada cumpla con los criterios de aceptación definidos, respondiendo correctamente tanto a entradas válidas como a casos de error esperados.

Casos de Prueba por Historia de Usuario

HU-05: Registro de Proveedores

Caso de Prueba	Descripción	Resultado Esperado
CP-05-01	Ingresar todos los campos válidos y guardar	Proveedor creado exitosamente
CP-05-02	Omitir campo obligatorio (ej: NIT)	Error: campo requerido
CP-05-03	Ingresar NIT con formato incorrecto	Rechazo por formato inválido
CP-05-04	Duplicar un NIT ya registrado	Mensaje de error por duplicado
CP-05-05	Enviar formulario sin datos	Bloqueo del envío

HU-06: Edición de Proveedores

Caso de Prueba	Descripción	Resultado Esperado
CP-06-01	Editar nombre y guardar	Datos actualizados correctamente
CP-06-02	Eliminar campo obligatorio y guardar	Error de validación
CP-06-03	Intentar editar proveedor inexistente	Error de recurso no encontrado
CP-06-04	Cerrar sin guardar	Cambios descartados
CP-06-05	Introducir caracteres inválidos	Error de validación

HU-07: Registro de Clientes

Caso de Prueba	Descripción	Resultado Esperado
CP-07-01	Registrar cliente con todos los datos válidos	Cliente registrado correctamente
CP-07-02	Email mal escrito	Error de formato
CP-07-03	Campo de contacto vacío	Mensaje de error obligatorio
CP-07-04	Cliente duplicado	Sistema rechaza duplicación
CP-07-05	Acceso sin autenticación	Acceso restringido

HU-08: Generar Orden de Compra

Caso de Prueba	Descripción	Resultado Esperado
CP-08-01	Seleccionar proveedor y crear orden	Orden registrada con éxito
CP-08-02	Omitir proveedor	Error por datos incompletos
CP-08-03	Producto fuera de stock	Advertencia al usuario
CP-08-04	Cancelar creación	Operación abortada
CP-08-05	Usuario no autenticado	Denegación de acceso

HU-09: Movimientos de Inventario

Caso de Prueba	Descripción	Resultado Esperado
CP-09-01	Registrar entrada de producto	Stock actualizado correctamente
CP-09-02	Salida sin stock suficiente	Rechazo de operación
CP-09-03	Cantidad negativa	Error de validación
CP-09-04	Producto no seleccionado	Formulario inválido
CP-09-05	Verificar movimiento en reporte	Registro visible en historial

Cobertura

- Se han definido 25 casos de prueba funcionales, con foco en entradas válidas, errores controlados, validaciones y comportamiento de flujos de usuario.
- Cada prueba será documentada con evidencia (captura o log) y su ejecución se registrará en una hoja de resultados.

Clientes List Create

OPTIONS GET

```
GET /api/clientes/

HTTP 200 OK
Allow: GET, POST, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

[
    {
        "id": 2,
        "nombre": "Sandy",
        "contacto": "sandy@hotmail.com",
        "preferencias": "Carnaval",
        "created_at": "2025-04-27T11:31:23.957510-05:00"
    },
    {
        "id": 5,
        "nombre": "victor",
        "contacto": "victor@hotmail.com",
        "preferencias": "Reunion",
        "created_at": "2025-04-27T11:35:37.931897-05:00"
    },
    {
        "id": 6,
        "nombre": "Jair",
        "contacto": "jair@gmail.com",
        "preferencias": "Cumpleanos",
        "created_at": "2025-04-27T11:45:16.344198-05:00"
    },
    {
        "id": 7,
        ...
    }
]
```

Proveedores List Create

OPTIONS GET

```
GET /api/proveedores/

HTTP 200 OK
Allow: GET, POST, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

[
    {
        "id": 1,
        "nombre": "Leonardo Alarcón",
        "contacto": "leo@gmail.com",
        "direccion": "calle 6 b h",
        "telefono": "3214569874",
        "email": "angle2014@gmail.com",
        "tipo_proveedor": "madera",
        "created_at": "2025-04-27T11:08:00.529961-05:00"
    },
    {
        "id": 5,
        "nombre": "alfonso.s.a.s",
        "contacto": "andres",
        "direccion": "calle 8 #9-54",
        "telefono": "3125467896",
        "email": "alfonso@hotmail.com",
        "tipo_proveedor": "transporte",
        "created_at": "2025-05-01T17:56:16.969148-05:00"
    },
    ...
]
```

Recursos Humanos List Create

OPTIONS GET ▾

GET /api/recursos_humanos/

```
HTTP 200 OK
Allow: GET, POST, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

[
    {
        "id": 1,
        "nombre": "Leonardo Alarcón",
        "cargo": "auxiliar",
        "salario": "5000000.00",
        "area": "almacen",
        "telefono": "3156889794",
        "correo": "leonardo@gmail.com",
        "contrato": "temporal",
        "created_at": "2025-05-06T13:11:16.228430-05:00"
    },
    {
        "id": 2,
        "nombre": "Leonardo Alarcón",
        "cargo": "auxiliar",
        "salario": "5000000.00",
        "area": "almacen",
        "telefono": "3156889794",
        "correo": "leonardo@gmail.com",
        "contrato": "temporal",
        "created_at": "2025-05-06T13:11:16.228430-05:00"
    }
]
```

Page not found (404)

Request Method: GET

Request URL: http://127.0.0.1:8000/api/Contacts/

Using the URLconf defined in `erp.urls`, Django tried these URL patterns, in this order:

1. [name='index']
2. `modulo/ [name='modulo']`
3. `login/ [name='login']`
4. `api/contacts/list/ [name='list_contacts']`
5. `api/auth/status/ [name='auth_status']`
6. `api/auth/register/ [name='register']`
7. `api/auth/login/ [name='login']`
8. `api/auth/save-contact/ [name='save_contact']`
9. `api/contacts/ [name='contact_list_create']`
10. `api/proveedores/ [name='proveedores_list_create']`
11. `api/proveedores/int:pk/ [name='proveedor_detail']`
12. `api/clientes/ [name='clientes_list_create']`
13. `api/clientes/int:pk/ [name='cliente_detail']`
14. `api/recursos_humanos/ [name='recursos_humanos_list_create']`
15. `api/recursos_humanos/int:pk/ [name='recursos_humanos_detail']`
16. `recursos_humanos/`
17. `recursos-humanos/`

The current path, `api/Contacts/`, didn't match any of these.

You're seeing this error because you have `DEBUG = True` in your Django settings file. Change that to `False`, and Django will display a standard 404 page.

The screenshot shows a code editor interface with several tabs and panels. On the left, there's a tree view of a Django project structure. The main area shows an open file named 'app.js'. In the bottom right corner, there's a terminal window displaying command-line output. The terminal output includes a stack trace and an error message about an integrity error related to a unique constraint on the 'correo' field of the 'recursos_humanos' model.

```
const CONFIG = {
    proveedores: {
        apiUrl: "/api/proveedores/",
        tableId: "tabla-proveedores",
        formId: "formulario-proveedor",
        fields: [
            { id: "nombre-proveedor", key: "nombre", required: true },
            { id: "contacto-proveedor", key: "contacto", required: true },
            { id: "direccion-proveedor", key: "direccion", required: false },
            { id: "telefono-proveedor", key: "telefono", required: false },
        ]
    }
}

File "C:\Users\USER\Desktop\ERP\venv\lib\site-packages\django\db\backends\utils.py", line 79, in execute
    return self._execute_with_wrappers(
        sql, params, many=False, executor=self._execute
        ^^^^^^^^^^
sql, params, many=False, executor=self._execute
        ^^^^^^^^^^
)
File "C:\Users\USER\Desktop\ERP\venv\lib\site-packages\django\db\backends\utils.py", line 92, in _execute_with_wrappers
    return executor(sql, params, many, context)
File "C:\Users\USER\Desktop\ERP\venv\lib\site-packages\django\db\backends\utils.py", line 100, in _execute
    with self.db.wrap_database_errors:
        ^^^^^^^^^^
File "C:\Users\USER\Desktop\ERP\venv\lib\site-packages\django\db\utils.py", line 91, in __exit__
    raise dj_exc_value.with_traceback(traceback) from exc_value
File "C:\Users\USER\Desktop\ERP\venv\lib\site-packages\django\db\backends\utils.py", line 103, in _execute
    return self.cursor.execute(sql)
        ^^^^^^
django.db.utils.IntegrityError: no se pudo crear el indice unico recursos_humanos_empleado.correo 814b2260 unique
DETAIL: La llave (correo)=(leonardo@gmail.com) esta duplicada.

PS C:\Users\USER\Desktop\ERP\erp> python manage.py showmigrations
```

Módulo de Recursos Humanos

Gestiona tus empleados, nómina y registros de ausentismos/horas extras.

Empleados	Cálculo de Nómina	Ausentismos y Horas Extras																																																					
Buscar empleado...		<button>Buscar</button>																																																					
<table><thead><tr><th>ID</th><th>Nombre</th><th>Cargo</th><th>Salario</th><th>Área</th><th>Teléfono</th><th>Correo</th><th>Contrato</th><th>Acciones</th></tr></thead><tbody><tr><td>1</td><td>Leonardo Alarcón</td><td>auxiliar</td><td>5000000.00</td><td>almacen</td><td>3156889794</td><td>leonardo@gmail.com</td><td>temporal</td><td><button>Editar</button> <button>Eliminar</button></td></tr><tr><td>2</td><td>Leonardo Alarcón</td><td>auxiliar</td><td>5000000.00</td><td>almacen</td><td>3156889794</td><td>leonardo@gmail.com</td><td>temporal</td><td><button>Editar</button> <button>Eliminar</button></td></tr><tr><td>3</td><td>Leonardo Alarcón</td><td>auxiliar</td><td>5000000.00</td><td>almacen</td><td>3156889794</td><td>leonardo@gmail.com</td><td>temporal</td><td><button>Editar</button> <button>Eliminar</button></td></tr><tr><td>4</td><td>Leonardo Alarcón</td><td>auxiliar</td><td>5000000.00</td><td>almacen</td><td>3156889794</td><td>leonardo@gmail.com</td><td>temporal</td><td><button>Editar</button> <button>Eliminar</button></td></tr><tr><td>5</td><td>Yinna</td><td>operario</td><td>1500000.00</td><td>administrativo</td><td>315648987</td><td>yina@gmail.com</td><td>indefinido</td><td><button>Editar</button> <button>Eliminar</button></td></tr></tbody></table>		ID	Nombre	Cargo	Salario	Área	Teléfono	Correo	Contrato	Acciones	1	Leonardo Alarcón	auxiliar	5000000.00	almacen	3156889794	leonardo@gmail.com	temporal	<button>Editar</button> <button>Eliminar</button>	2	Leonardo Alarcón	auxiliar	5000000.00	almacen	3156889794	leonardo@gmail.com	temporal	<button>Editar</button> <button>Eliminar</button>	3	Leonardo Alarcón	auxiliar	5000000.00	almacen	3156889794	leonardo@gmail.com	temporal	<button>Editar</button> <button>Eliminar</button>	4	Leonardo Alarcón	auxiliar	5000000.00	almacen	3156889794	leonardo@gmail.com	temporal	<button>Editar</button> <button>Eliminar</button>	5	Yinna	operario	1500000.00	administrativo	315648987	yina@gmail.com	indefinido	<button>Editar</button> <button>Eliminar</button>
ID	Nombre	Cargo	Salario	Área	Teléfono	Correo	Contrato	Acciones																																															
1	Leonardo Alarcón	auxiliar	5000000.00	almacen	3156889794	leonardo@gmail.com	temporal	<button>Editar</button> <button>Eliminar</button>																																															
2	Leonardo Alarcón	auxiliar	5000000.00	almacen	3156889794	leonardo@gmail.com	temporal	<button>Editar</button> <button>Eliminar</button>																																															
3	Leonardo Alarcón	auxiliar	5000000.00	almacen	3156889794	leonardo@gmail.com	temporal	<button>Editar</button> <button>Eliminar</button>																																															
4	Leonardo Alarcón	auxiliar	5000000.00	almacen	3156889794	leonardo@gmail.com	temporal	<button>Editar</button> <button>Eliminar</button>																																															
5	Yinna	operario	1500000.00	administrativo	315648987	yina@gmail.com	indefinido	<button>Editar</button> <button>Eliminar</button>																																															
<button>Agregar Empleado</button>																																																							

Herramienta Utilizada

Las pruebas se documentarán en una hoja de cálculo y serán ejecutadas sobre el entorno de pruebas del ERP, utilizando navegadores modernos y el cliente web del sistema.

5. Planificación de Pruebas QA y WhiteBox

Objetivo General

Verificar que el sistema ERP Logístico no solo cumple con los requisitos funcionales, sino que también satisface los atributos de calidad previamente definidos (seguridad, rendimiento, mantenibilidad y disponibilidad), aplicando pruebas orientadas a escenarios y tácticas arquitectónicas. Además, se definen pruebas de caja blanca para garantizar cobertura estructural del código.

A. Pruebas de Atributos de Calidad (QA)

Atributo de Calidad	Escenario de Prueba	Táctica Aplicada	Criterio de Éxito
Seguridad	Ingreso con credenciales correctas e incorrectas. Intento de inyección SQL en formulario.	Autenticación por roles, middleware CSRF y validación de entrada	El sistema impide acceso no autorizado y bloquea entradas maliciosas.
Rendimiento	Consulta masiva de proveedores y órdenes. Registro de 100 movimientos de inventario.	Indexación de base de datos, caché, reducción de llamadas redundantes	Respuestas inferiores a 2 segundos. Sin degradación crítica con carga media.

Mantenibilidad	Modificación del módulo de clientes sin alterar otros módulos.	Modularización y bajo acoplamiento	El sistema compila y opera sin errores tras cambios localizados.
Disponibilidad	Simulación de caída de servicio de base de datos. Verificación de respuesta parcial del sistema.	Infraestructura distribuida, manejo de errores y redundancia	El sistema mantiene operación parcial o informa adecuadamente al usuario.

B. Pruebas de Caja Blanca (WhiteBox)

Las siguientes pruebas se enfocan en revisar la estructura lógica del código fuente:

Historia de Usuario	Prueba WhiteBox	Técnica Aplicada	Resultado Esperado
HU-05	Validar condicionales en creación de proveedor	Cobertura de decisiones	Todos los flujos (válido, inválido, duplicado) son ejecutados
HU-06	Editar proveedor con diferentes combinaciones de campos	Prueba de condiciones múltiples	El sistema permite actualizar si y solo si hay consistencia
HU-07	Verificar función de validación de email	Ánálisis de flujo de datos	No hay fallos al evaluar expresiones regulares
HU-08	Probar creación de orden y cálculo de totales	Cobertura de rutas	El cálculo se ejecuta correctamente con distintos datos
HU-09	Verificar actualización de stock (positiva/negativa)	Prueba de bucles y entradas límite	El sistema actualiza solo si se cumplen las reglas del dominio

Herramientas de Prueba Sugeridas

- Postman: pruebas a la API REST.
- Pytest / Unittest (Python): pruebas unitarias y de flujo interno.
- Apache JMeter o Locust: simulación de carga para evaluar rendimiento.
- Coverage.py: verificar cobertura de código Python.
- Jira / Excel / TestLink: documentación de resultados.

Resultado Esperado

Estas pruebas asegurarán que el ERP Logístico:

- Se comporta correctamente bajo condiciones normales y anómalas.
- Tiene una arquitectura robusta, con componentes desacoplados y manejables.
- Cumple con requisitos no funcionales claves, garantizando calidad a largo plazo.

Actividad 6: Proponer un Cronograma de Trabajo para el Sprint 2

Historia	Tarea	Responsable	Inicio	Fin	Entregable
HU-05	Crear modelo Proveedor	Santiago	06/05	06/05	Modelo con validaciones base
HU-05	Formulario de registro Proveedor	Anderson	06/05	07/05	Formulario funcional
HU-05	Conexión con API REST	Víctor	07/05	08/05	Integración backend completa
HU-06	Lista y edición de Proveedores	Anderson	07/05	08/05	Vista con opción de edición activa
HU-07	Registro de Cliente	Santiago	06/05	07/05	Formulario y validaciones operativas
HU-08	Módulo de Orden de Compra	Anderson	07/05	08/05	Creación de orden asociada a proveedor
HU-09	Registro de movimiento inventario	Víctor	06/05	07/05	Entrada/salida almacenada correctamente
Todos	Pruebas funcionales y QA	Todos	08/05	09/05	Pruebas completadas con registro de resultados
Todos	Documentación y evidencias	Todos	09/05	10/05	Documentos entregables preparados