



# Informe de Cierre: Sprint 1

**Proyecto:** Qualifast Buildings – Estación de Policía Inteligente

**Período:** 2 Semanas

**Estado:** Completado (100% de los objetivos alcanzados)

**Equipo:** 4 Integrantes

**Dedicación total:** ~160 horas (40h/persona)

---

## Objetivo del Sprint

El objetivo principal de este primer Sprint fue establecer los cimientos del proyecto de software:

- Definir la arquitectura del sistema:** Implementar el patrón Modelo-Vista-Controlador (MVC).
  - Persistencia de datos:** Configurar el sistema de archivos JSON para simular una base de datos.
  - Seguridad y Acceso:** Implementar el sistema de autenticación (Login) y la gestión de sesiones.
  - Navegación:** Crear el enrutador principal de la interfaz gráfica con Flet.
- 

## Resumen de Esfuerzo y Equipo

Según la metodología Scrum, el equipo ha trabajado de forma coordinada para cerrar las Historias de Usuario (HU) planificadas.

Rol	Miembro	Responsabilidad Principal en Sprint 1
Scrum Master	Nicolás Ramírez	Configuración del entorno,

		repositorio y estructura de carpetas.
<b>Product Owner</b>	Álvaro López	Diseño de UI de Login, lógica de controladores y navegación.
<b>Dev IoT / Backend</b>	Fernando Fernández	Implementación de modelos de datos (JSON) y funciones CRUD.
<b>QA / Dev</b>	Daniel Vicente	Lógica de validación de usuarios y estructura de seguridad.



## Backlog del Sprint y Tareas Realizadas

A continuación, se detallan las tareas completadas, extraídas directamente del tablero de gestión (Trello), agrupadas por Historia de Usuario.



### HU-01: Estructura de Proyecto e Infraestructura

**Objetivo:** Crear un entorno de desarrollo colaborativo y limpio.

Estado	Tarea	Responsable	Tiempo Estimado	Entregable
✓	Configurar el repositorio Git	Nicolás	4h	Repositorio con .gitignore y README inicial.
✓	Crear estructura de	Nicolás	3h	Carpetas modelo/, vista/,

	carpetas (MVC)			assets/.
✓	Definición de dependencias	Nicolás	2h	Archivo requirements.txt con flet.

## ● HU-02: Creación de Modelos de Datos (Backend)

**Objetivo:** Gestionar la persistencia de datos sin base de datos SQL, utilizando JSON.

Estado	Tarea	Responsable	Tiempo Estimado	Entregable
✓	Crear archivos JSON base	Fernando	4h	usuarios.json, presos.json, actuadores.json.
✓	Funciones auxiliares del modelo	Fernando	5h	Funciones _leer_json y _escribir_json.
✓	Implementar manejador de datos	Fernando	8h	Módulo manejador_datos.py con lógica CRUD.
✓	Simulación de carga de datos IoT	Fernando	6h	Script simulador_iot.py para generar datos dummy.

## ● HU-03: Modelo y Vista de Login

**Objetivo:** Crear la primera interfaz visual que permita el acceso seguro.

Estado	Tarea	Responsable	Tiempo Estimado	Entregable
✓	Implementar lógica de autenticación	Daniel	5h	Función validar_usuario en el modelo.
✓	Diseñar interfaz gráfica de Login	Álvaro	8h	vista_login.py con campos de texto y estilos.

## 🟡 HU-04: Controlador de Login y Navegación

**Objetivo:** Conectar la vista con el modelo y permitir el cambio de pantallas.

Estado	Tarea	Responsable	Tiempo Estimado	Entregable
✓	Implementar controlador de Login	Álvaro	6h	Función on_login_click en main.py.
✓	Implementar enrutador principal	Álvaro	10h	Función route_change para gestión de vistas.
✓	Implementar cierre de sesión	Álvaro	2h	Lógica de limpieza de sesión page.session.c

				lear().
--	--	--	--	---------

---



## Métricas del Sprint

- **Total de Tareas Planificadas:** 10
- **Total de Tareas Completadas:** 10
- **Tasa de éxito:** 100%
- **Incidencias bloqueantes:** 0
- **Horas totales (Desarrollo + Gestión + Learning):** ~160 Horas

*Nota: El tiempo restante hasta completar las 40h por persona se dedicó a reuniones de Daily Scrum, aprendizaje del framework Flet, diseño de la arquitectura y documentación.*

---



## Evidencias del Incremento (Demo)

Al finalizar el Sprint 1, el software es funcional y cuenta con las siguientes capacidades demostrables:

1. **Inicio del Sistema:** El archivo main.py lanza la aplicación correctamente.
  2. **Validación:** El sistema distingue entre usuarios válidos e inválidos leyendo desde usuarios.json.
  3. **Roles:** El sistema detecta el rol (comisario, inspector, policía) y lo guarda en la sesión.
  4. **Routing:** Tras un login exitoso, el sistema redirige automáticamente al Dashboard (base).
  5. **Persistencia:** Los archivos JSON se generan automáticamente si no existen.
- 



## Retrospectiva del Sprint

¿Qué funcionó bien?

- La división de carpetas MVC (modelo, vista, main) ha facilitado que trabajemos en

paralelo sin generar conflictos de código.

- La elección de **Flet** ha permitido crear la interfaz de usuario mucho más rápido de lo esperado.
- El uso de Trello con etiquetas de colores por Historia de Usuario nos ayudó a visualizar el progreso claramente.

## ¿Qué podemos mejorar?

- Para el Sprint 2, debemos definir mejor los permisos específicos de cada botón en la vista antes de empezar a programar.
  - Mejorar la seguridad de las contraseñas (actualmente en texto plano en usuarios.json) para futuros sprints.
- 

## Aprobación

Product Owner: Álvaro López

Scrum Master: Nicolás Ramírez

Fecha de Cierre: 14 de Noviembre de 2025