



# PROYECTO FINAL DAW

## Guía Estratégica Completa

*Todo lo que necesitas saber sobre el proyecto, los exámenes y la defensa oral*

2º DAW • Digitech Málaga • Febrero 2026

*Documento basado en las instrucciones del profesor (transcripciones de clase)*

## 1. QUÉ ES REALMENTE EL PROYECTO

Este proyecto sustituye el examen tradicional del segundo trimestre. No es un ejercicio académico más: es un proyecto web real, desplegado en producción, con SEO técnico aplicado, que servirá como herramienta profesional para buscar empleo.

### Lo que el profesor quiere transmitir

"Yo quiero algo ganador para ustedes. Quiero que cuando vayan a una empresa, puedan decir: mira, aquí lo tengo, está desplegado, está en un subdominio y puedes ver cómo está indexando. Es un proyecto de principio a fin."

En esencia: cada alumno desarrolla un sitio web individual (blog, inmobiliaria, tienda, portafolio, etc.) que debe cumplir unos requisitos técnicos mínimos y, además, diferenciarse por el valor añadido que aporte.

### Naturaleza del proyecto:

- **Desplegado:** Web individual desplegada en un subdominio real (no localhost).
- **Modular:** Frontend desacoplado: HTML por un lado, estilos por otro, JS por otro.
- **SEO real:** Sitemap funcional, robots.txt, slugs amigables, SEO técnico real.
- **Dinámico:** Filtrado dinámico de contenido (por categorías, precio, zona, etc.).
- **Medible:** Análisis de métricas reales de indexación, performance y crecimiento.




## 2. QUÉ EXIGE EL PROFESOR (REQUISITOS MÍNIMOS)



### IMPORTANTE: Mínimo ≠ Nota alta

Cumplir el checklist no garantiza un 10. Garantiza el aprobado (5). La nota sube con el criterio de aplicación, la calidad del código, la experiencia de usuario y el valor añadido que aportes.

### Checklist de requisitos obligatorios:

	Requisito	Descripción
1	Subdominio	Desplegado en un subdominio real, accesible online para el profesor
2	Sitemap funcional	Todas las páginas incluidas. Investigar cómo construirlo.
3	Robots.txt	Configurado correctamente para permitir/bloquear indexación
4	Slugs amigables	URLs tipo dominio/reparacion-movil, NO dominio/pagina3
5	Frontend desacoplado	HTML, CSS y JS separados. Código modular y organizado.
6	Filtrado dinámico	Bloques de contenido que cambien según categorías/filtros
7	SEO técnico	Estructura semántica, metadatos, URLs coherentes
8	Métricas	Análisis de indexación, performance, crecimiento orgánico
9	Defensa individual	Explicar cada línea de código. El profesor puede preguntar.
10	Exposición oral	Presentar el proyecto y defender las decisiones técnicas



### VALOR AÑADIDO: La clave para subir nota

El profesor busca que vayas más allá de lo pedido. Ejemplos: aplicar un framework por tu cuenta (autodidacta), implementar buenas prácticas avanzadas, optimizar la experiencia de usuario, trabajar con datos reales escalables. Cita textual del profesor: "Si lo ha aplicado, claramente lo ha estudiado por fuera. Autodidacta. Eso convence."

### 📄 3. EXAMEN DE JAVASCRIPT — Detalles completos

El examen de JavaScript se realizará en clase, en las horas de examen. Durará aproximadamente 2 horas. Usarás tu propio proyecto (o una API genérica si no lo tienes listo) y deberás demostrar que dominas la lógica desde cero.

#### ✅ LO QUE SÍ PUEDES TRAER PREPARADO:

- ✅ **Datos mock:** Datos mockeados (JSON con tus productos/artículos/pisos)
- ✅ **Tailwind:** Tailwind configurado y listo para usar
- ✅ **Reset CSS:** Reset CSS aplicado
- ✅ **Tipografía:** Sistema tipográfico definido (H1, H2, H3, párrafos, etc.)
- ✅ **HTML base:** Root base HTML vacío (estructura mínima)
- ✅ **Carpetas:** Estructura de carpetas organizada
- ✅ **JS enlazado:** Archivo JS enlazado (vacío de lógica)
- ✅ **Estilos:** Estilos preparados (variables, paleta cromática)
- ✅ **Formulario HTML:** Formulario HTML (solo maqueta, sin funcionalidad)

#### ❌ LO QUE NO PUEDES TRAER:

- ❌ **Lógica JS:** Lógica de JavaScript implementada
- ❌ **Filtrado:** Filtrado ya construido
- ❌ **Render:** Render dinámico ya hecho
- ❌ **Estado:** Gestión de estado preparada
- ❌ **Formulario funcional:** Formulario funcional con JS
- ❌ **Inteligencia Artificial:** USO DE IA (suspensión automática)

#### 🔧 LO QUE CONSTRUIRÁS EN EL EXAMEN:

Concepto	Qué debes demostrar
Grid responsiva	Montar una cuadrícula de productos/tarjetas que se adapte a pantalla
Renderizado dinámico	Pintar datos del JSON en el DOM mediante JavaScript
Métodos de Array	.map(), .filter(), .find(), .reduce(), .forEach() aplicados con criterio
Desestructuración	Extraer propiedades de objetos/arrays de forma limpia
Filtrado funcional	Botones/selectores que filtren contenido en tiempo real
Gestión de estado	Variables/objetos que controlen qué se muestra y qué no
Formulario + normalización	Capturar datos, limpiarlos (trim, toLowerCase, etc.) y añadirlos al estado
Actualización del DOM	Que cada acción del usuario se refleje instantáneamente en pantalla

💣 **ATENCIÓN: Uso de IA = Suspenso automático**

El profesor lo ha dejado claro: si detecta uso de inteligencia artificial durante el examen, la nota es automáticamente un suspenso. No hay segunda oportunidad en este punto.



## 4. EXAMEN DE INTERFACES — Detalles completos

El examen de Interfaces evalúa tu capacidad de maquetar con criterio visual, adaptabilidad y coherencia. Se realizará en la misma sesión que el de JavaScript (2 horas combinadas).

### ✓ LO QUE SÍ PUEDES TRAER:

- ✓ **Reset:** Reset CSS aplicado
- ✓ **Tipografía:** Sistema tipográfico con variables
- ✓ **Variables:** Variables CSS (:root con colores, espaciados, etc.)
- ✓ **Paleta:** Paleta cromática definida
- ✓ **HTML base:** Root base HTML vacío
- ✓ **Tailwind:** Tailwind configurado
- ✓ **Layout:** Layout base vacío (estructura sin contenido)

### ✗ LO QUE NO PUEDES TRAER:

- ✗ **Solución completa:** Solución completa ya maquetada
- ✗ **Maquetación:** Maquetación final ya hecha
- ✗ **HTML resuelto:** HTML completo resuelto

### LO QUE CONSTRUIRÁS EN EL EXAMEN:

- **Maquetación real:** Maquetación real desde cero con criterio de diseño
- **Grid aplicada:** CSS Grid aplicada con sentido (no solo flexbox)
- **Responsividad:** Adaptabilidad a distintos tamaños de pantalla
- **Criterio visual:** Criterio visual: colores coherentes, espaciado, jerarquía



## 5-6. RESUMEN VISUAL: QUÉ SE PUEDE Y QUÉ NO

Elemento	Examen JS	Examen Interfaces
Datos mockeados (JSON)	✓ Sí	—
Tailwind configurado	✓ Sí	✓ Sí
Reset CSS	✓ Sí	✓ Sí
Sistema tipográfico	✓ Sí	✓ Sí
Root HTML vacío	✓ Sí	✓ Sí
Estructura carpetas	✓ Sí	—
Archivo JS enlazado	✓ Sí	—
Estilos/variables CSS	✓ Sí	✓ Sí
Paleta cromática	✓ Sí	✓ Sí
Layout base vacío	—	✓ Sí
Lógica JS implementada	✗ No	—
Filtrado hecho	✗ No	—
Render dinámico hecho	✗ No	—
Gestión estado hecha	✗ No	—
Formulario funcional	✗ No	—
Maquetación final hecha	—	✗ No
HTML completo resuelto	—	✗ No
Uso de IA	✗ SUSPENSO	✗ SUSPENSO



## 7. QUÉ SE CONSTRUYE EN CLASE (EL DÍA DEL EXAMEN)

El día del examen, en aproximadamente 2 horas, deberás construir desde cero sobre tu base preparada:

- 1. Grid responsiva:** Montar una grid responsiva de tarjetas/productos
- 2. Render dinámico:** Renderizar datos dinámicamente desde tu JSON mock al DOM
- 3. Métodos array:** Aplicar métodos de array con criterio (.map, .filter, .find, etc.)
- 4. Desestructuración:** Usar desestructuración para acceder a propiedades
- 5. Filtrado:** Implementar filtrado funcional con botones/controles
- 6. Estado:** Gestionar el estado de la aplicación
- 7. Formulario:** Crear formulario con normalización de datos (trim, etc.)
- 8. DOM reactivo:** Que cada acción actualice el DOM instantáneamente
- 9. Maquetación:** Maquetar con CSS Grid, responsividad y criterio visual







## 8. CÓMO SERÁ LA DEFENSA ORAL

La defensa es individual. El profesor evaluará que realmente dominas tu código y que no lo ha hecho otra persona (ni una IA). Cita del profesor: "Quiero ver los nombres de clases, los nombres de variables, las funciones. Te voy a pedir que me expliques esta línea de aquí, y tú me la explicas."

### Qué esperar en la defensa:

- **Mostrar producto terminado:** Presentar tu proyecto desplegado en el subdominio
- **Justificar decisiones:** Explicar decisiones técnicas: ¿por qué esta estructura? ¿por qué este filtro?
- **Explicar código línea a línea:** El profesor puede señalar cualquier línea y pedirte que la expliques
- **Defender métricas:** Mostrar métricas: indexación, performance, crecimiento orgánico
- **Diferenciarte:** Explicar el valor añadido que diferencia tu proyecto



### **RIESGO REAL: El profesor compara códigos**

Si detecta funciones muy iguales entre alumnos, o que solo cambiaste la paleta cromática, investigará más a fondo. Cada proyecto debe tener nombres de clases, variables y funciones propias y coherentes. No copies: aprende y adapta.



## 9. MÉTRICAS QUE SE EVALUARÁN

El profesor no solo mira si "funciona". Evaluará con criterio profesional estas áreas:

Área	Qué se evalúa	Cómo destacar
SEO Técnico	Sitemap, robots.txt, slugs, estructura semántica	Amplitud semántica correcta, keywords coherentes
Indexación	Si Google indexa tus páginas correctamente	Verificar en Search Console / herramientas SEO
Performance	Velocidad de carga, optimización	Lighthouse >90, imágenes optimizadas, carga lazy
Crecimiento	Evolución del tráfico orgánico	Documentar métricas semana a semana
Experiencia usuario	Usabilidad, diseño, navegación fluida	Criterio visual profesional, accesibilidad
Código escalable	Modularidad, mantenibilidad, buenas prácticas	Separación de responsabilidades, código limpio
Valor añadido	Lo que aportaste más allá de lo pedido	Framework autodidacta, features extra, innovación



### Nota sobre la dificultad del nicho

El profesor entiende que un nicho con keywords difíciles (ej: seguridad informática) tardará más en posicionar que uno fácil (ej: blog de recetas). No penaliza por eso; al contrario, valorará positivamente que hayas elegido un reto mayor si el trabajo técnico está bien hecho.

## 10. ARGUMENTACIÓN ESTRATÉGICA: 60% PROYECTO / 30% EXAMEN / 10% RESTO

El profesor ha propuesto cambiar los porcentajes de evaluación. El objetivo es que el proyecto tenga el mayor peso posible para incentivar dedicarle tiempo real.

### Distribución propuesta:

Componente	Porcentaje	Justificación
Proyecto desplegado	60%	Es el producto real, la herramienta profesional, lo que enseñas en entrevistas
Examen práctico (JS + Interfaces)	30%	Demuestra que sabes hacerlo tú, sin IA, bajo presión
Resto evaluación continua	10%	Participación, asistencia, trabajo en clase



### ACCIÓN REQUERIDA: Documento formal de petición

El profesor ha pedido que le enviemos un escrito firmado por todos los alumnos, solicitando el cambio de porcentajes (de 50/50 a 60/30/10 o similar). Debe enviarse por correo con todas las firmas. Sin este paso, los porcentajes no cambian oficialmente.

### Por qué nos conviene el 60/30:

- Podemos dedicar más horas al proyecto sin miedo a perder peso en el examen.
- Un proyecto bien hecho = portafolio profesional real para prácticas y empleo.
- El examen práctico del 30% sigue existiendo para demostrar competencia individual.
- Es la forma de diferenciarnos de otros alumnos de DAW a nivel nacional.

## ⚠️ 11. RIESGOS SI NO SE HACE CON CRITERIO

- ⚠️ Copiar código de un compañero o usar código idéntico con cambios cosméticos → El profesor compara y lo detecta.
- ⚠️ Usar IA en el examen → Suspenso automático, sin negociación.
- ⚠️ No entender tu propio código en la defensa → El profesor puede bajarte la nota dramáticamente.
- ⚠️ Entregar sin desplegar → No se evalúa. El profesor necesita verlo online.
- ⚠️ No tener métricas documentadas → Pierdes toda la parte de SEO y crecimiento.
- ⚠️ Trabajar solo sin compartir investigación → Pierdes eficiencia (sitemap, robots.txt, etc. son iguales para todos).
- ⚠️ Dedicar todo el tiempo al proyecto e ignorar el examen → El 30% del examen puede hundirte.
- ⚠️ No presentar el documento de petición firmado → Los porcentajes se quedan en 50/50 o peor.

### 💡 Frase clave del profesor sobre la rentabilidad

"Hay alumnos que por ahí los veo un medio flojo y me queda la duda: ¿lo sabe o lo está haciendo con él [IA]? Si no llega, no es rentable. Esto a mano ya no es rentable... pero con criterios." — El profesor valora que uses herramientas, pero que demuestres que entiendes lo que haces.



## 12. PLAN DE ACCIÓN EN 2 SEMANAS

### SEMANA 1: Investigación + Estructura + Despliegue

Día	Tarea	Responsable/Equipo
L-M	Definir nicho de cada proyecto. Investigar sitemap, robots.txt, slugs amigables.	Todo el equipo (compartir hallazgos)
M-J	Crear estructura de carpetas, HTML base, CSS variables, Tailwind. Montar repo Git.	Individual (cada uno su proyecto)
J-V	Contratar/configurar hosting con subdominio. Primer despliegue (aunque sea esqueleto).	Coordinar costes en equipo
S-D	Crear datos mock (JSON). Diseñar estructura de filtrado. Primeros estilos.	Individual

### SEMANA 2: Desarrollo + SEO + Preparación examen

Día	Tarea	Responsable/Equipo
L-M	Implementar filtrado dinámico, render de contenido, formulario. Iterar sobre diseño.	Individual
M-J	Implementar SEO: sitemap, robots.txt, slugs. Configurar Search Console. Primeras métricas.	Individual (compartir cómo hacerlo)
J-V	Pulir proyecto. Preparar defensa oral. PRACTICAR examen JS (grid + render + filtro + formulario sin IA).	Individual
S-D	Ensayo general: hacer el examen completo cronometrado (2h). Corregir fallos.	Individual (recomendado en grupo)



#### DEADLINE CRÍTICO

El proyecto debe estar desplegado y presentable en la semana de exámenes. El que quiera entregar antes, puede. Pero el día del examen, el profesor tiene que ver el producto terminado online.



## 13. DISTRIBUCIÓN DE RESPONSABILIDADES



### Filosofía clave del profesor

"Cada uno trabaja en su proyecto individual, PERO si uno descubre cómo hacer el sitemap, ya lo han descubierto todos. Si trabajáis en equipo, ganáis todos." — La investigación se comparte; la implementación es individual.

### Tareas que SE COMPARTEN (investigación):

- ✓ Cómo crear un sitemap funcional
- ✓ Cómo configurar robots.txt correctamente
- ✓ Cómo crear slugs dinámicos amigables
- ✓ Cómo desplegar en subdominio (costes, opciones de hosting)
- ✓ Cómo configurar Search Console y medir métricas
- ✓ Patrones de filtrado dinámico en JavaScript

### Tareas que son INDIVIDUALES:

- Elección de nicho/temática del proyecto
- Diseño visual, paleta cromática, UX
- Implementación del código (nombres de clases, variables, funciones PROPIAS)
- Contenido y datos del proyecto
- Defensa oral (explicar cada línea)
- Valor añadido y diferenciación



## 14. VALOR PROFESIONAL PARA PRÁCTICAS Y EMPLEO

Este proyecto no es solo una nota: es tu carta de presentación profesional. El profesor lo ha dejado claro múltiples veces en clase.

### Lo que este proyecto demuestra a un empleador

Cuando llegas a una entrevista y dices "mira, aquí lo tengo desplegado, está en un subdominio, puedes ver cómo indexa, estas son las métricas"... eso no lo tiene el 95% de los alumnos de DAW. Eso es lo que te diferencia.

### Lo que un recruiter ve en este proyecto:

- **Producción real:** Sabe trabajar en producciones escalables (no solo ejercicios de clase)
- **Visión completa:** Entiende SEO técnico, no solo "hacer páginas bonitas"
- **Autodidacta:** Es autodidacta: ha aplicado tecnologías por fuera del temario
- **Ciclo completo:** Sabe desplegar, mantener y medir un producto web real
- **Comunicación:** Puede defender técnicamente lo que ha construido



### Para tus prácticas FCT

Si llegas a tu empresa de prácticas con este proyecto terminado, desplegado y con métricas reales, tu tutor de empresa verá inmediatamente que no eres "un becario más". Esto te posiciona para que te ofrezcan quedarte después de las prácticas.

## ? DUDAS QUE DEBEMOS CONFIRMAR CON EL PROFESOR

De las transcripciones de clase se extraen varios puntos ambiguos que necesitan confirmación explícita antes de avanzar:

- 1. Fecha y duración exacta de los exámenes:** El examen de JavaScript y el de Interfaces ¿se hacen el mismo día en una sesión de 2 horas? ¿O son días separados? La transcripción sugiere que se combinan, pero no queda 100% claro.
- 2. Hosting y subdominio:** El profesor mencionó contratar un servicio web ("Web3" / hosting). ¿Es obligatorio usar un hosting de pago? ¿Se acepta GitHub Pages, Vercel, Netlify o similares gratuitos? ¿Quién asume el coste si es de pago?
- 3. APIs y datos:** ¿Podemos usar API genéricas públicas o los datos tienen que ser 100% mockeados/proprios? ¿Se permite consumir APIs externas (ej: API de inmobiliaria, noticias)?
- 4. Uso de IA en el desarrollo del proyecto:** "No se puede usar IA en el examen" está claro. Pero ¿durante el desarrollo del proyecto se puede usar IA como herramienta de aprendizaje? ¿Dónde está el límite exacto?
- 5. Porcentajes definitivos:** ¿Los porcentajes definitivos son 60/30/10? ¿O siguen en 50/50 hasta que enviemos el documento firmado? ¿Hay deadline para enviarlo?
- 6. Fecha límite de entrega:** ¿Cuándo es exactamente la fecha límite de entrega del proyecto? ¿Coincide con el día de los exámenes o se puede entregar antes?
- 7. Repositorio Git:** El profesor mencionó que mirará el repositorio. ¿Es obligatorio usar Git/GitHub? ¿Se evalúa el historial de commits?
- 8. Costes compartidos:** Si un alumno no puede asumir costes de hosting, ¿pierde opción a hacer el proyecto? ¿Hay alternativa gratuita aceptada?
- 9. API genérica para examen:** El profesor mencionó la API propia del proyecto vs. una API genérica. Si alguien no tiene el proyecto listo, ¿qué API genérica se acepta en el examen?

*Documento elaborado a partir de las transcripciones de clase del profesor.*

2º DAW • Digitech Málaga • Febrero 2026

¡A por el 10! 💪