



## ENTREGA 6

### PROYECTO SEMESTRAL 1-2016

El proyecto de este semestre consistirá en un catálogo de productos online, de 2 familias de productos y promociones.

El proyecto se desarrollará en 5 entregas parciales más la entrega final:

~~E3. Viernes 8 de abril – Maquetas en HTML + CSS~~

~~E4. Martes 3 de mayo – Autenticación y manejo de usuarios~~

~~E5. Martes 17 de mayo – Catálogos y administrador~~

**E6. Lunes 30 de mayo – Mejoras varias JS**

E7. Viernes 10 de junio – APIs REST

EF. Viernes 24 de junio – Mejoras varias y detalles

## REPOSITORIO Y ENTREGAS

Esta entrega se realizará en los mismos grupos registrados desde la entrega 3, al igual que el resto del proyecto semestral. El grupo no podrá cambiar durante el transcurso del proyecto.

### ENTREGA: REPOSITORIO Y SERVIDOR

Esta entrega se realizará igual que las entregas anteriores, mediante el repositorio privado registrado. Recordar que las entregas son evaluadas de acuerdo a **la última copia presente en el repositorio del grupo al momento de la entrega**.

También deben actualizar su aplicación **en producción (Heroku o equivalente)**.

### ABANDONO DE UN GRUPO

En caso de que **un alumno no continúe con el proyecto**, será evaluado con la nota mínima en todas las entregas siguientes (incluyendo la entrega y presentación final, reprobando automáticamente el curso de acuerdo a las reglas publicadas en el programa).

## Entrega 6

Para **el integrante restante**, que deberá continuar sólo con el proyecto, se aplicarán reglas especiales en cada entrega (aunque probablemente esto no evitará que la carga individual sea mayor).

**Estos casos excepcionales deben ser avisados a la brevedad al profesor.**

## ENTREGA 6

En esta entrega el foco será en mejorar funcionalidades existentes con el uso de JavaScript.

### CATÁLOGOS V2

Hasta este momento la vista de un catálogo podría ser vista como sólo un listado simple de productos. Esto cambiará en este punto:

#### PAGINACIÓN

Lo primero es que deberán incorporar paginación en la vista del catálogo. Es decir, al momento de mostrar los productos, deberán obtener y mostrar a lo más los primeros K elementos; además deberán incluir una barra con las páginas disponibles, de modo que al hacer clic en la página siguiente, se muestren los siguientes K elementos, y así hasta la última página.

Por ejemplo, si el catálogo se compone de 72 productos, habrán 4 páginas de 20 productos: producto 1 al 20, 21 al 40, 41 al 60 y 61 al 72.

Salvo por la página inicial, deben cargar todas las páginas usando JavaScript vía AJAX.

#### VISTA PRELIMINAR DE UN PRODUCTO

En la vista de un catálogo deberán incorporar algún tipo de vista previa que use un cuadro "modal" construido en JavaScript. Algunas ideas:

- Al hacer clic en el producto, ver un cuadro flotante (modal) con más información, que incluya un link a ir al producto y otro para agregarlo directamente al carro de compras.
- Al hacer clic en la imagen, ver en un cuadro modal una imagen de mayor tamaño.
- Al hacer clic en un link "más opciones", ver un cuadro con variantes del producto (por ejemplo, en un auto, otros colores o configuraciones disponibles)

El único requisito en este punto es que la información adicional se muestre en un cuadro modal que se construya con información adicional a la mostrada (puede ser en atributos data- o obtenido vía AJAX).

### ADMINISTRACIÓN: VALIDACIONES

En la sección de administración deberán agregar validaciones usando JavaScript (incluyendo mensajes apropiados). La idea es que, para todos los formularios:

- No puede ser enviado si no se han ingresado todos los datos obligatorios.
- No puede ser enviado con datos del tipo erróneo (por ejemplo no permitir algo como "hola" en un campo que debe ser exclusivamente numérico).

## Entrega 6

- Todos los mensajes de error estén en español – es decir, no es suficiente con validaciones HTML5, cuyos mensajes no son personalizables ni localizables.

### LOCALIZACIÓN

Como en la entrega anterior, todo el trabajo debe estar localizado para español.

### RESTRICCIONES

A priori, no pueden usar ninguna librería / gema, salvo por:

- Gemas que vienen con Rails
- Bcrypt
- Gemas para conexión con base de datos o instalación en heroku o equivalente
- Otras gemas que sean autorizadas

### PLAZO Y FORMA DE ENTREGA

El plazo máximo de entrega es el lunes 30 de mayo, a las 23:59 hrs. La forma de entrega consistirá en:

- Tener la aplicación corriendo, disponible para la corrección
- Un repositorio, según se explica anteriormente. Para la corrección se usará una copia del código tras el último commit anterior al plazo máximo de entrega.

### AYUDANTÍA DE AYUDA

El día viernes 27 de mayo habrá una ayudantía para la resolución de dudas y apoyo de trabajo, en el módulo 5 (15:30 a 16:50) en la sala B16.

### EVALUACIÓN

La evaluación de esta entrega responderá a la siguiente rúbrica:

Ptje ítem	Ítem / aspecto	Excelente (5)	Bueno (4)	Deficiente (2)	No logrado (0)
4	a) Funcionalidad: paginación vía AJAX	Hay paginación vía AJAX, que carga en cada página sólo los productos de la página. Además, la lógica de la paginación está donde debe (según M-V-C)	Hay paginación que opera vía AJAX, la que carga en cada página sólo los productos de la página.	Hay paginación vía AJAX, pero o bien carga cosas innecesarias cada vez o tiene otros problemas.	No hay paginación o esta no funciona por AJAX.

## Entrega 6

Ptje ítem	Ítem / aspecto	Excelente (5)	Bueno (4)	Deficiente (2)	No logrado (0)
3	b) Funcionalidad: vista previa de productos	Hay una vista previa de productos, cuya información es construida a partir atributos data-X o obtenida vía AJAX.	N/A	Hay una vista previa de productos, pero el JavaScript no la construye sólo se encarga de que el modal se vea.	No hay vista previa de productos.
3	c) Funcionalidad: validación de formularios en la administración	Todos los formularios en la administración poseen alguna validación vía JavaScript de los campos obligatorios y de los tipos de cada campo, entregando mensajes adecuados en español.	Al menos 3 formularios poseen la validación requerida.	Al menos 1 formulario posee validación vía JavaScript tanto de campos requeridos como de tipos, con mensajes en español.	Ni un solo formulario cumple con la validación solicitada.
5	d) Calidad del código: limpieza del HTML (unobstrusive JavaScript)	No hay ninguna forma de JavaScript en el HTML, ni en tags <script>, ni en atributos href. Tampoco hay atributos que ejecutan JavaScript como los on-algo (onload, onclick, etc). Todo el JavaScript está en archivos .js cargados en el <head> del HTML.	No hay ninguna forma de JavaScript en el HTML, pero los archivos .js se cargan al final, justo antes del cierre del <body>.	Hay a lo más una inclusión de JavaScript en el HTML o un atributo on-algo que ejecuta JavaScript.	Hay más de una inclusión de JavaScript en el HTML.
2	e) Layout y diseño: este punto no evalúa que el diseño sea "bonito", pero se busca valorar el esfuerzo de lograr un catálogo "razonable" del punto de vista visual.	El catálogo de productos y la vista de un producto es razonable y podría usarse en un catálogo real.	N/A	Se aprecia un esfuerzo en lograr vistar visualmente razonables, pero hay detalles evidentes.	Las vistas no son razonables para un catálogo o productos reales.