**Sprint 1: Wireframes**

Fecha de entrega: **28/10/2021**

Última revisión: **13.2.2022** (Juan Pablo Badino)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SPRINT 1 – Wireframes** | | | | |
| **1) Definición de la temática del marketplace** Readme con una breve descripción de los productos y/o servicios ofrecidos por su sitio. Descripción del público al que apunta el sitio. | **2) Inspiración** Listado en el Readme de, al menos, 5 sitios de referencia fundamentando la elección. | **3) Boceto del sitio** Wireframe digital o analógico que incluya todas las secciones (home, detalle, del producto, carrito de compras, formularios de registro y login. | **4. (Opcional)**  Boceto / Diseño con colores del sitio, logo y tipografías | **Porcentaje aprobado** |
| **100% (APROBADO)** | **100% (APROBADO)** | **100% (APROBADO)** | **-** | **100% (APROBADO)** |

Cumple deadline: **Sí**

Porcentaje de aprobación:

**100% (APROBADO)**

**Sprint 2: HTML & CSS**

Fecha de entrega: **16/11/2021**

Última revisión: **13.2.2022** (Juan Pablo Badino)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **SPRINT 2 -** | | | | | |
| **Vista: carrito de compras** Detalle y posibilidad de modificación o eliminación de los productos del carrito. En un archivo productCart.html junto con todos los recursos necesarios, estilos CSS e imágenes | **Página de registro**  Estructura y definición de los campos del formulario en un archivo register.html junto con todos los recursos necesarios, estilos CSS e imágenes | **Formulario de login** Estructura y definición de los campos del formulario. | **Coherencia visual** Integrar la estructura y los estilos de las páginas hechas en un sitio unificado que mantenga el mismo diseño y estilo en todas sus secciones. | **Sitio responsive** Diseño adaptable a celular, tablet y desktop. |
| **100% (APROBADO)** | **100% (APROBADO)** | **100% (APROBADO)** | **100% (APROBADO)** | **100% (APROBADO)** |
| **Estructura de archivos** Creación de la misma utilizando Node + Express + sendFile() | **Estructura HTML** Definición de la estructura utilizando tags semánticos. | **Estilo CSS** Definición de estilos optimizada. Sin repetir reglas. | **Vista: home del sitio** Con la oferta de productos y serviciosen un archivo index.html con todos los recursos, imágenes y estilos de CSS | **Vista: detalle de producto** En un archivo porduct.Detail.html junto con todos los recursos necesarios, estilos CSS e imágenes |
| **100% (APROBADO)** | **100% (APROBADO)** | **100% (APROBADO)** | **100% (APROBADO)** | **100% (APROBADO)** |

Cumple deadline: **Sí |** Porcentaje de aprobación: **95% (APROBADO)**

**Sprint 3: Motores de plantillas (Template Engines)**

Fecha de entrega: **30/11/2021**

Última revisión: **13.2.2022** (Juan Pablo Badino)

|  |  |  |  |
| --- | --- | --- | --- |
| **SPRINT 3 – Motores de plantillas (Template Engines)** | | | |
| **Template engine I**mplementar el motor de templates | **Template engine** Separar los componentes repetidos (head, navbar, footer, etc) en archivos parciales | **Página: formulario de carga de producto** Página de administración en un archivo productAdd.html junto con todos los recursos, estilos CSS e imágenes necesarios. | **Porcentaje** |
| **100% (APROBADO)** | **100% (APROBADO)** | **100% (APROBADO)** | **100% (APROBADO)** |

Cumple deadline: **Sí**

Porcentaje de aprobación:

**100% (APROBADO)**

**Sprint 4: JSON & Métodos HTTP**

Fecha de entrega: **16/12/2021**

Última revisión: **13.2.2022** (Juan Pablo Badino)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **SPRINT 4 – JSON y Métodos HTTP** | | | | | |
| **JSON DB** Definir los campos necesarios para los productos y generar archivo JSON | **JSON DB** Definir los campos necesarios para los usuarios y generar archivo JSON | **Listado** ABM | **Alta** ABM | **Baja** ABM | **Modificación** ABM |
| **100% (APROBADO)** | **100% (APROBADO)** | **100% (APROBADO)** | **100% (APROBADO)** | **100% (APROBADO)** | **100% (APROBADO)** |

Cumple deadline: **Sí**

Porcentaje de aprobación:

**100% (APROBADO)**

**Sprint 5:** Middlewares & Authentication

Fecha de entrega: **18/1/2022**

Última revisión: **10.2.2022** (Juan Pablo Badino)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sprint 5 – Requerimientos** | | | | |
| Entidad Usuarios | Formulario Registro | Formulario log-in | Función de recordar al usuario (opcional) | Rutas de huéspedes y usuarios |
| **100% (aprobado)** | **100% (aprobado)** | **100% (aprobado)** | **100% (aprobado)** | **100% (aprobado)** |
| *Implementar la entidad de usuarios.* | *Implementar el formulario de registro de usuarios.* | *Implementar el log-in de usuarios.* | *Implementar la función de recordar al usuario.* | *Implementar rutas de guest/huéspedes y de usuarios.* |

Cumple deadline: **Sí**

Porcentaje de aprobación:

**100% (aprobado)**

**Sprint 6:** Bases de datos

Fecha de entrega: **18/1/2022**

Última revisión: **15.2.2022** (Juan Pablo Badino)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **SPRINT 6** | | | | | |
| **Diagrama de base de datos** Armar el Diagrama de Entidades y Relaciones (DER). | **Script de creación de estructura de base de datos** Escribir las sentencias de SQL que crearán las tablas y sus relaciones. Archivo structure.sql | **(Opcional) Script de población de base de datos** Archivo con extensión data.sql que permita poblar la base con datos. | **Creación de carpeta Sequelize y archivos de modelos** Carpeta database que incluya los archivos de configuración y archivos de modelos junto con sus relaciones. | **BREAD** Rutas, controladores y vistas necesarias para que suceda lo detallado previamente utilizando Sequelize para trabajar con la base de datos. | **Porcentaje** |
| **100% (aprobado)** | **100% (aprobado)** | **100% (aprobado)** | **100% (aprobado)** | **100% (aprobado)** | **100%** |

Cumple deadline: **Sí**

Porcentaje de aprobación:

**100% (aprobado)**

**Sprint 7:** Validaciones

Fecha de entrega: **18/1/2022**

Última revisión: **10.2.2022** (Juan Pablo Badino)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **SPRINT 7** | | | | | |
| **Diagrama de base de datos** Armar el Diagrama de Entidades y Relaciones (DER). | **Script de creación de estructura de base de datos** Escribir las sentencias de SQL que crearán las tablas y sus relaciones. Archivo structure.sql | **(Opcional) Script de población de base de datos** Archivo con extensión data.sql que permita poblar la base con datos. | **Creación de carpeta Sequelize y archivos de modelos** Carpeta database que incluya los archivos de configuración y archivos de modelos junto con sus relaciones. | **BREAD** Rutas, controladores y vistas necesarias para que suceda lo detallado previamente utilizando Sequelize para trabajar con la base de datos. | **Porcentaje** |
| **% ()** | **% ()** | **% ()** | **% ()** | **% ()** |  |

Cumple deadline: **Sí**

Porcentaje de aprobación:

**% ()**

**Sprint 8:** API’s & React

Fecha de entrega: **18/1/2022**

Última revisión: **10.2.2022** (Juan Pablo Badino)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **SPRINT 8 -** | | | | | |
| **Diagrama de base de datos** Armar el Diagrama de Entidades y Relaciones (DER). | **Script de creación de estructura de base de datos** Escribir las sentencias de SQL que crearán las tablas y sus relaciones. Archivo structure.sql | **(Opcional) Script de población de base de datos** Archivo con extensión data.sql que permita poblar la base con datos. | **Creación de carpeta Sequelize y archivos de modelos** Carpeta database que incluya los archivos de configuración y archivos de modelos junto con sus relaciones. | **BREAD** Rutas, controladores y vistas necesarias para que suceda lo detallado previamente utilizando Sequelize para trabajar con la base de datos. | **Porcentaje** |
| **% ()** | **% ()** | **% ()** | **% ()** | **% ()** |  |

Cumple deadline: **Sí**

Porcentaje de aprobación:

**% ()**

|  |
| --- |
| **Repositorio grupal** |
| **Repositorio grupal** Url del repositorio donde irán subiendo sus avances en el Proyecto |
|  |
|  |
|  |