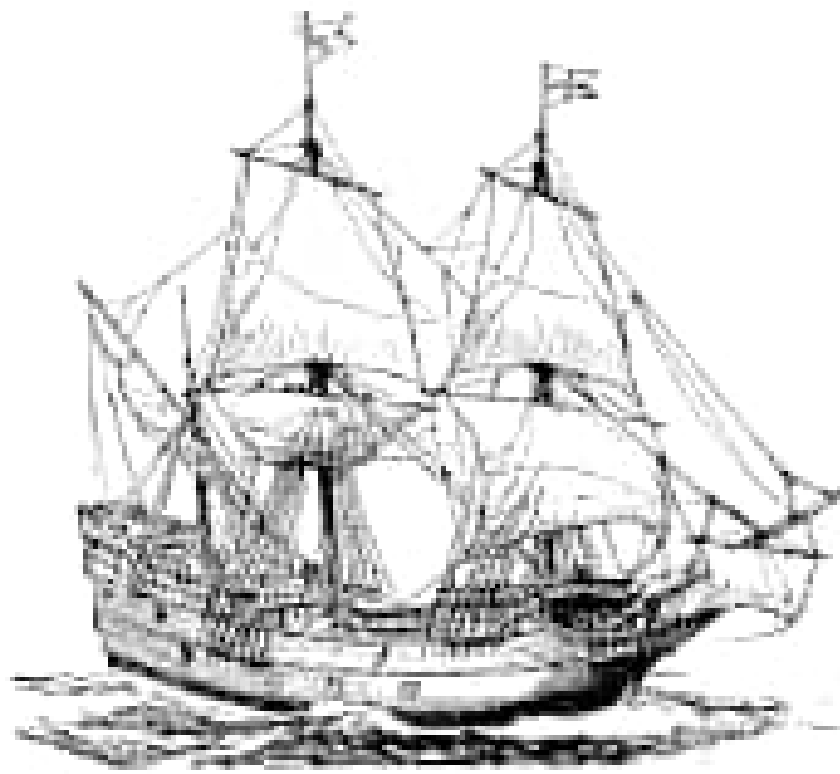


universidad Politecnica de Tecamac
programacion visual

BATALLA-NAVAL



Emmanuel Torres Servin

157134344 Lourdes Herrerias Aguirre Sanchez

157134375 Rocio Alvarado Yanez Ines

157134345 Sanchez Ariza Edric Yami

¿que es un mockup?

Un mockup, traducido del inglés como boceto, es un fotomontaje a través del cual los diseñadores gráficos pueden presentar sus propuestas a los clientes. Los mockups cuentan con muchos usos. No sólo sirven para presentar las propuestas del diseño de logos. Se utilizan también con mucha frecuencia en el diseño web y el diseño de apps. Por ejemplo, podemos encontrar mockups para hacer diseños de camisetas o incluso calcetines en tiendas online o e-commerce que aún no han llegado a fabricar este producto.

importancia

Los mockups pueden abarcar desde logotipos, tarjetas, folletos, webs... en general, elementos de diseño dentro del proceso creativo. Su elaboración permite el ahorro en materiales, impresión y montajes al plasmar la idea en formatos tan variados como el papel, vinilos, interfaces, rótulos... vamos, que hay mock-ups para todos los gustos, formas y tamaños. El uso de mockups constituye algo típico para las agencias por la serie de ventajas que conllevan: aportan valor añadido a los diseños, materializan las ideas haciéndolos reales, proporcionan detalles y generan empatía al poder experimentar algo tangible. Fuede que el cliente tenga una idea inicial, pero todo el procedimiento de adaptarlo a las tendencias y al diseño es algo particular, único. Y aunque pareciera sencillo, no está al alcance de cualquiera el poder transmitir lo que se busca.

componentes visuales

Un componente es visual cuando tiene una representación gráfica en tiempo de diseño y ejecución (botones, barras de scroll, cuadros de edición, etc.), y se dice no visual en caso contrario (temporalizadores, cuadros de diálogo —no visibles en la fase de diseño—, etc).

componentes visuales

Los componentes no visuales se pueden colocar en los formularios de la misma manera que los controles, aunque en este caso su posición es irrelevante.



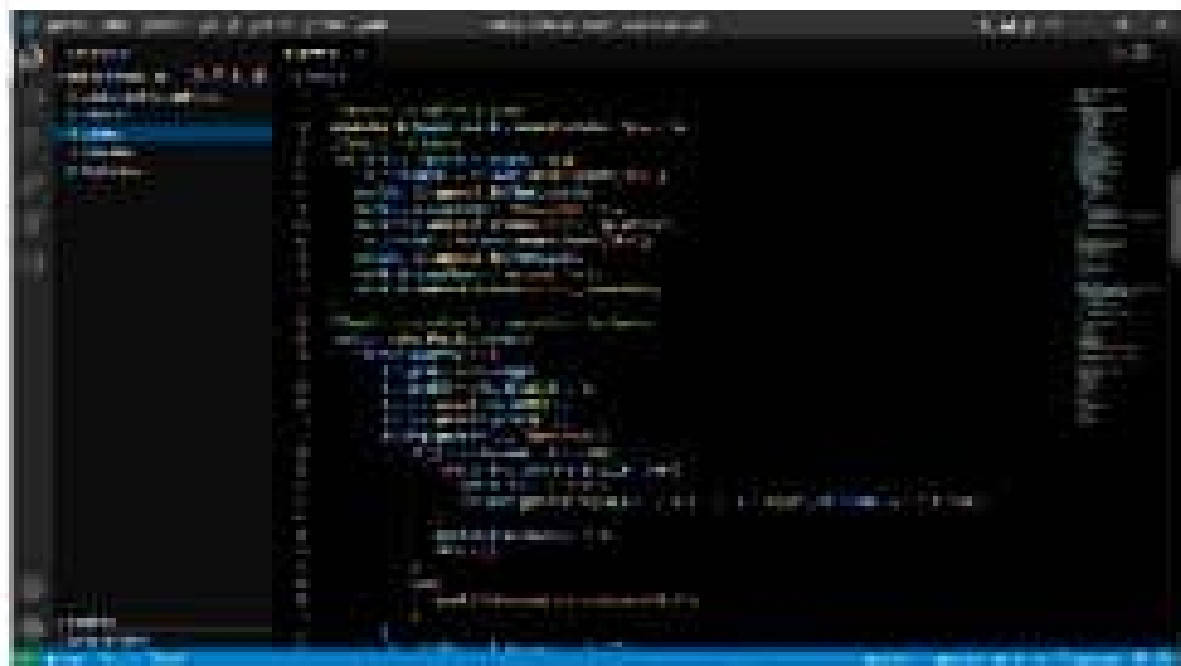
```
1 // C++ program to implement a linked list
2
3 #include <iostream>
4 using namespace std;
5
6 // Node structure
7 struct Node {
8     int data;
9     Node* next;
10 };
11
12 // Function to create a new node
13 Node* createNode(int data) {
14     Node* newNode = new Node;
15     newNode->data = data;
16     newNode->next = NULL;
17     return newNode;
18 }
19
20 // Function to insert a new node at the end
21 void insertAtEnd(Node* head, int data) {
22     if (head == NULL) {
23         head = createNode(data);
24         return;
25     }
26     Node* temp = head;
27     while (temp->next != NULL) {
28         temp = temp->next;
29     }
30     temp->next = createNode(data);
31 }
32
33 // Function to print the list
34 void printList(Node* head) {
35     if (head == NULL) {
36         return;
37     }
38     Node* temp = head;
39     while (temp != NULL) {
40         cout << temp->data << " ";
41         temp = temp->next;
42     }
43     cout << endl;
44 }
45
46 // Driver code
47 int main() {
48     Node* head = NULL;
49     insertAtEnd(head, 1);
50     insertAtEnd(head, 2);
51     insertAtEnd(head, 3);
52     printList(head);
53     return 0;
54 }
```

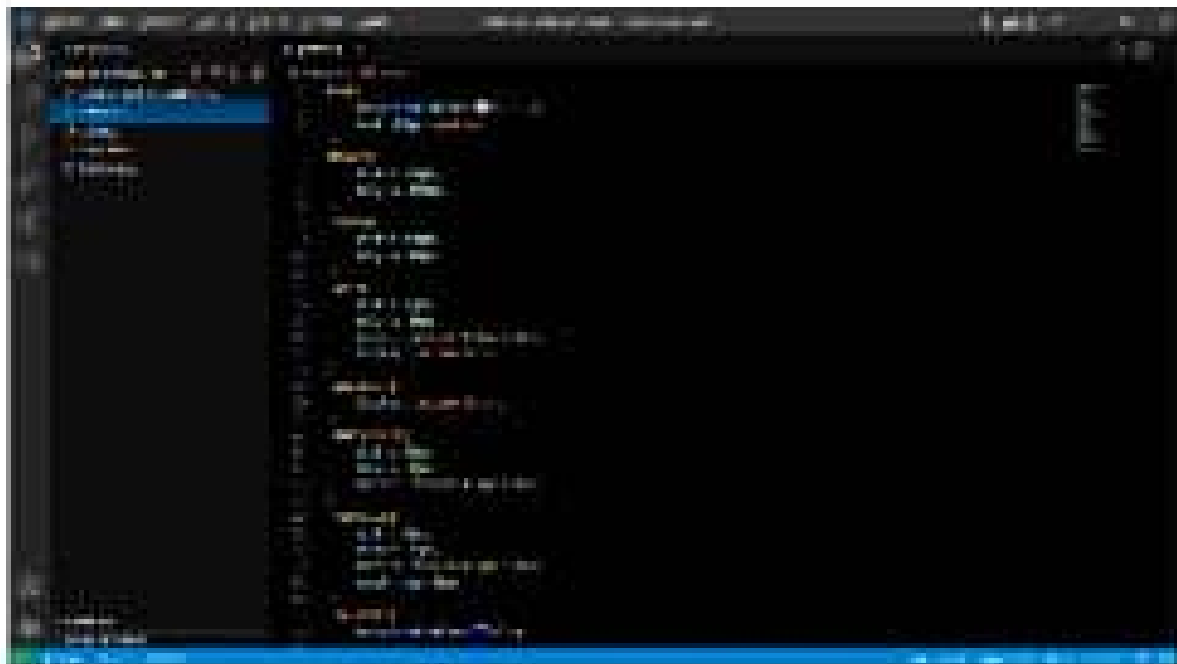
```
1 // C++ program to implement a linked list
2
3 #include <iostream>
4 using namespace std;
5
6 // Node structure
7 struct Node {
8     int data;
9     Node* next;
10 };
11
12 // Function to create a new node
13 Node* createNode(int data) {
14     Node* newNode = new Node;
15     newNode->data = data;
16     newNode->next = NULL;
17     return newNode;
18 }
19
20 // Function to insert a new node at the end
21 void insertAtEnd(Node* head, int data) {
22     if (head == NULL) {
23         head = createNode(data);
24         return;
25     }
26     Node* temp = head;
27     while (temp->next != NULL) {
28         temp = temp->next;
29     }
30     temp->next = createNode(data);
31 }
32
33 // Function to print the list
34 void printList(Node* head) {
35     if (head == NULL) {
36         return;
37     }
38     Node* temp = head;
39     while (temp != NULL) {
40         cout << temp->data << " ";
41         temp = temp->next;
42     }
43     cout << endl;
44 }
45
46 // Driver code
47 int main() {
48     Node* head = NULL;
49     insertAtEnd(head, 1);
50     insertAtEnd(head, 2);
51     insertAtEnd(head, 3);
52     printList(head);
53     return 0;
54 }
```

index



battle.js





battle.css

