

Introducció

Objectiu tema

Familiarització amb l'Entorn de Desenvolupament: Ser capaç de configurar l'entorn de desenvolupament Visual Studio Code, incloent la instal·lació i ús de plugins i snippets rellevants per agilitzar el seu flux de treball.

Configuració de l'entorn

Per a poder realitzar el curs serà necessari tenir instal·lat una sèrie de components.

- **Node:** Entorn en temps d'execució JavaScript. Versió 21 o superior. [enllaç](#)
- **Yarn:** Gestor de paquets yarn [enllaç](#)
- **VSCode:** (recomanat) Editor de codi [enllaç](#). Es recomana l'ús d'aquest editor encara que cada un és lliure de fer-ho amb aquell que més li agradi.
- **Docker:** (opcional) Gestor de contenidors [enllaç](#). Encara que queda fora del curs es faciliten els DockerFiles per executar-ho amb aquest entorn.

VSCode

Visual Studio Code aka vscode és un editor de codi multiplataforma creat per Microsoft.

Vscode és flexible i àgil, cosa que ens facilita l'edició de codi, a més, de permetre instal·lar plugins per adaptar-lo a les nostres necessitats.

Encara que no és obligatori usar aquest editor és l'opció recomanada.

Plugins

Els plugins o extensions són complements addicionals que amplien la funcionalitat de vscode per a adaptar-ho a les nostres necessitats.

A continuació us llistam una sèrie de complements a tenir en compte:

- [Vue Extension Box](#)
 - [Vue - Official](#): Suport per a VUE
 - [Path Intellisense](#): autocompleta nom de fitxers
 - [Auto Close Tag](#): Tanca automàticament les etiquetes
 - [Auto Rename Tag](#): Canvia el nom a la parella d'etiquetes a l'editar-ne una.
 - [ESLint](#): Analitzador de codi
 - [Sass](#): Formatejador de codi SASS
 - [SCSS Formatter](#): formatejador de SCSS
 - [Prettier](#): Formatejador de codi
- [Vue 3 Support - All In One](#): Complementos per a VUE3
- [Vue VSCode Snippets](#): Snippets per a VUE3
- [CodeMetrics](#) (opcional): Calcula la complexitat del codi indicant quan convé millorar-ho.
- [SonarLint](#) (opcional): Recomanacions de millora de codi

- **Material Icon Theme** (opcional): Canvi d'icones, especialment útil per a la identificació dels fitxers.

Settings

La configuració del workspace permet establir una configuració per a un projecte en concret.

Per a poder establir la configuració s'ha d'editar (o crear) el fitxer `<arrel projecte>`
`</arrel>/.vscode/settings.json`.

A continuació la configuració recomanada per poder personalitzar al vostre gust:

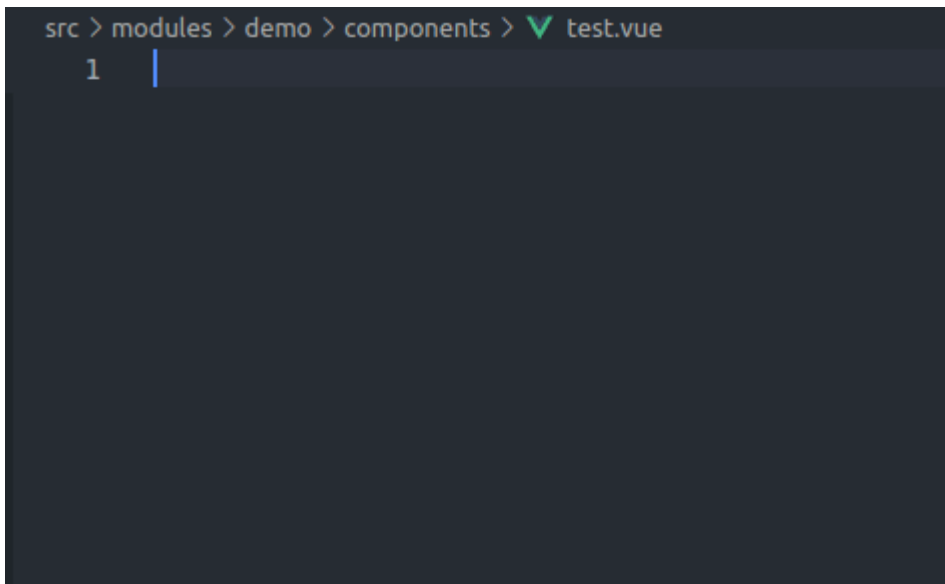
```
{
  "eslint.alwaysShowStatus": true,
  "eslint.format.enable": true,
  "editor.bracketPairColorization.enabled": true,
  "editor.guides.bracketPairs": true,
  "editor.formatOnSave": true,
  "editor.defaultFormatter": "esbenp.prettier-vscode",
  "editor.codeActionsOnSave": {
    "source.fixAll.eslint": true,
    "source.organizeImports": true
  },
  "eslint.validate": ["javascript", "javascriptreact", "typescript",
"vue"],
  "typescript.tsdk": "node_modules/typescript/lib",
  "editor.tabSize": 2,
  "i18n-ally.localesPaths": ["src/locales", "src/core/i18n"]
}
```

Snippets

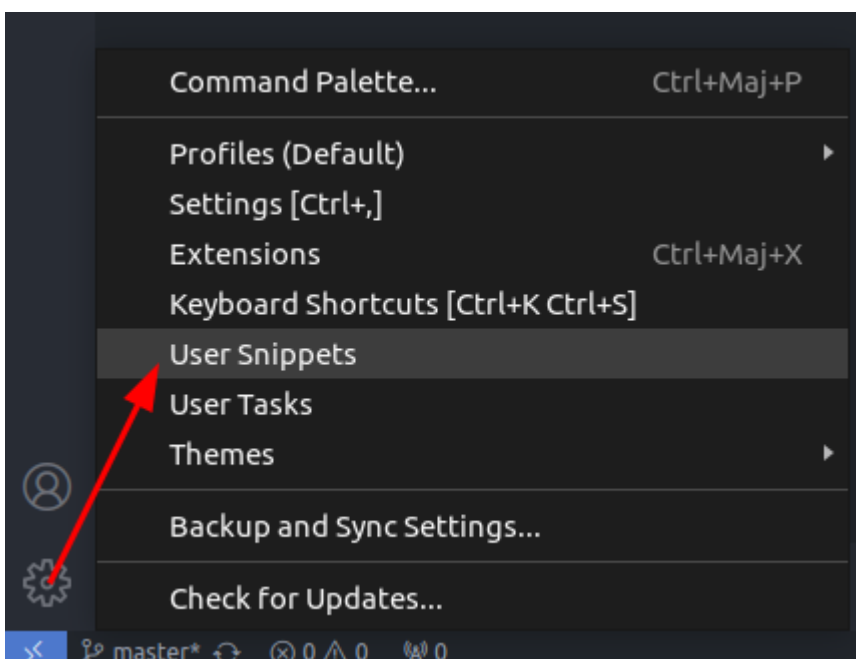
Els snippets són plantilles de fragments de codi que vscode ens permet cridar per a insertar aquest codi al fitxer que estam editant.

Una de les extensions que hem instal·lat ja incorpora una sèrie de snippets útils com, per exemple: `vbase-3-ts-setup` o abreujat `vb3tss`.

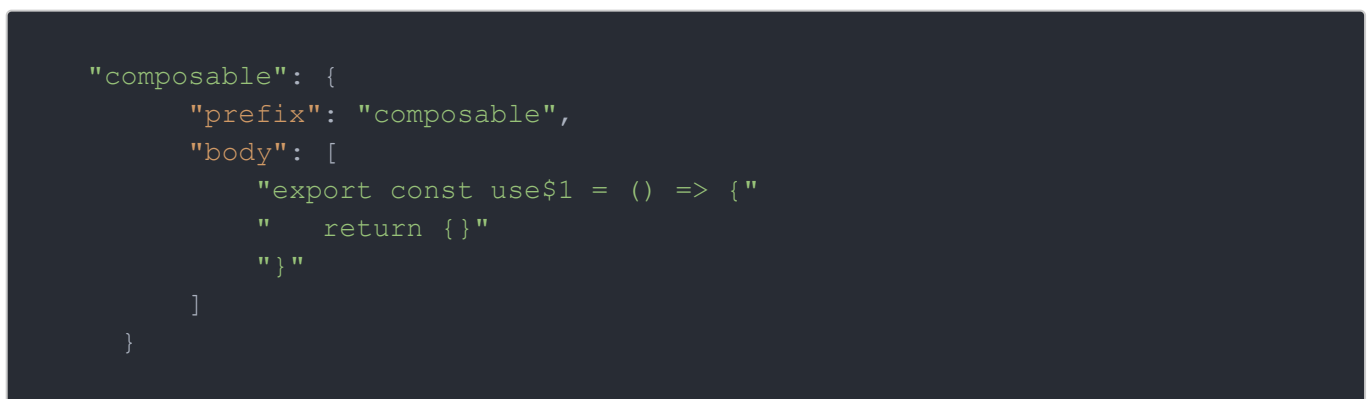
Com podeu veure basta començar a escriure per a que es mostrin les opcions i acceptar, en sortir, la que desitjam



A més dels snippets que ens proporcionen les extensions també podem crear els nostres.



Una vegada seleccionada aquesta opció haurem de marcar el llenguatge en concret i ens apareixerà l'editor. Aquí podem anar afegint un element per a cada snippet que volgем:



El **prefix** és el nom a través del qual cercarem i el **body** el contingut que es pintarà. Destacar que es poden posar variables amb \$1, \$2... Una vegada s'inserti el codi el cursor es posarà al lloc de les variables per a

poder completar.

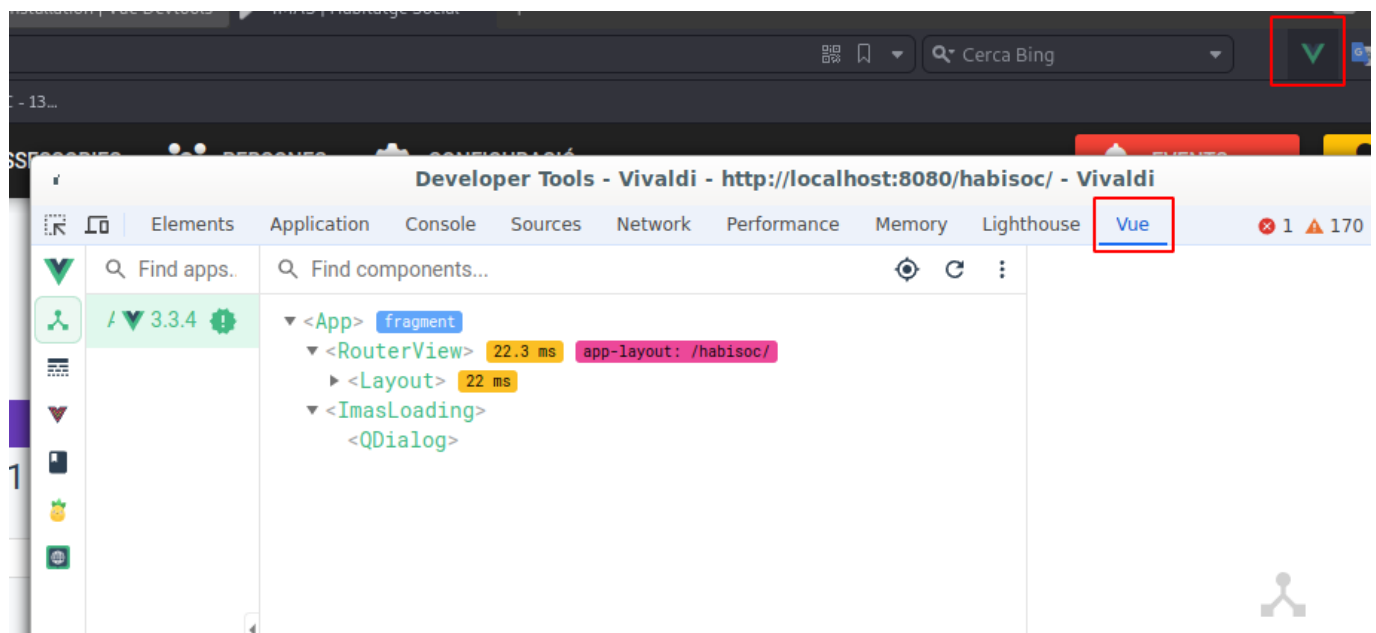
[Més informació](#)

Vue Devtools

Una altre eina important és l'extensió de navegador **Vue Devtools** que facilita el debbug d'aplicacions Vue.js.

La podeu instal·lar desde la web oficial: <https://devtools.vuejs.org>

Una vegada accediu a una web amb Vue l'icona de l'extensió s'activarà i si accediu al developer tools del navegador tindreu una nova pipella anomenada Vue amb distintes eines que s'aniràn explicant al llarg del curs.

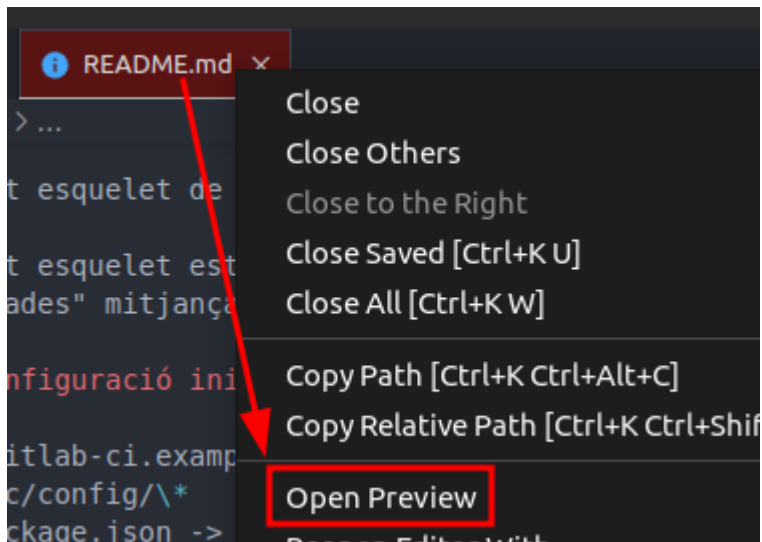


Esquelet

Descarregarem l'esquelet de Vue 3 des de l'enllaç adjunt al curs en format ZIP i el descomprimirem.

Una vegada dins hem de llegir i seguir les instruccions del README.md

Amb vscode podem previsualitzar el fitxer fent clic amb el boto secundari damunt el fitxer:



Encara que les opcions del `.env` no siguin usades, sí que s'ha de crear el fitxer en base al `.env.pre`.

Docker

L'estudi de docker surt de l'abast d'aquest projecte. Així i tot, s'adjunta un model de configuració.

```
# docker-compose
version: "3.7"
services:
  frontend:
    build:
      context: ./
      dockerfile: Dockerfile
      args:
        # proxy: "http://proxycentral.imasmallorca.net:8080/"
    container_name: cursVue-frontend
    image: cursVue-frontend
    ports:
      - "8080:8080"
    volumes:
      - ./frontend/:/app
```

```
# Dockerfile
# Name the node stage "builder"
FROM node:21 AS builder

ARG proxy

ENV http_proxy $proxy
ENV https_proxy $proxy

# Set working directory
WORKDIR /app

# Copy all files from current directory to working dir in image
```

```
COPY . .
```

```
CMD ["yarn", "dev-docker"]
```