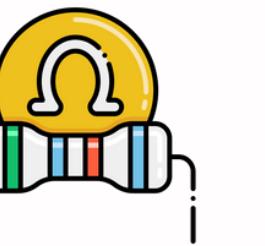


OMEGA



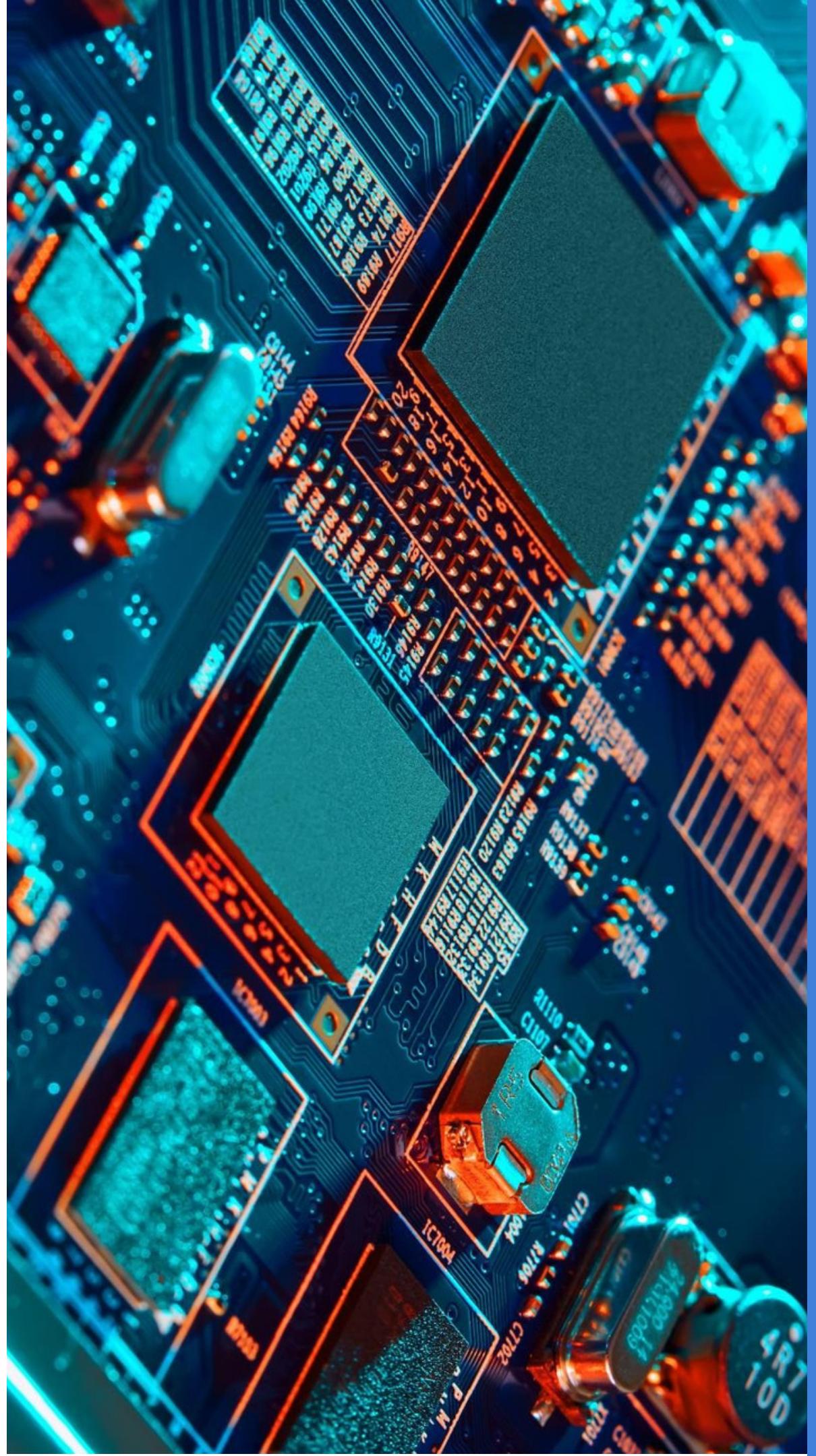
Davide Salvetti
Matteo Verzeroli

Firmware & Hardware Engineers

15 Via A. Einstein
Dalmine, BG 24044

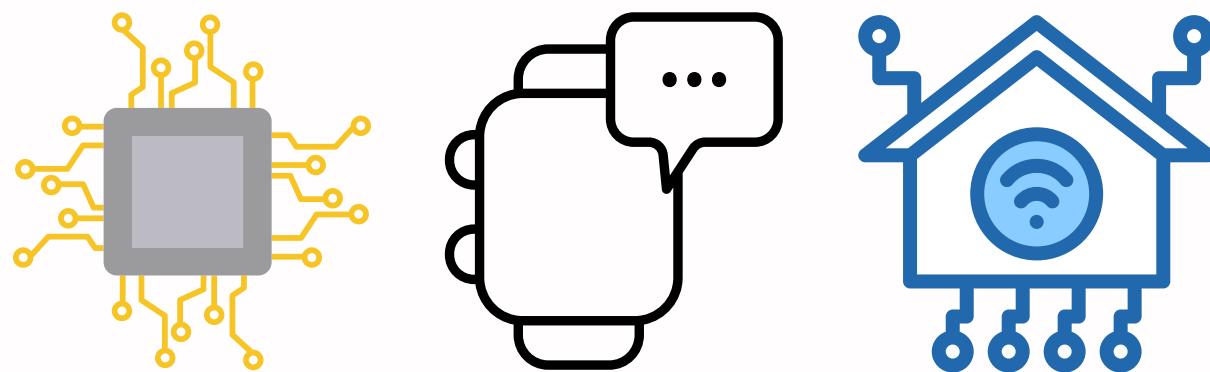
info@omega.com

349-123-4567



CHI SIAMO

OMEGA è un'azienda giovane che opera nel settore dell'elettronica, con esperienza nello sviluppo di prodotti in ambito IoT, wearable e smart home.



Cosa facciamo:

- Sviluppo Hardware
- Simulazioni di circuiti elettronici con LTSpice
- Sviluppo Firmware
- Analisi dei dati
- Sviluppo di applicazioni Desktop



PROBLEMA

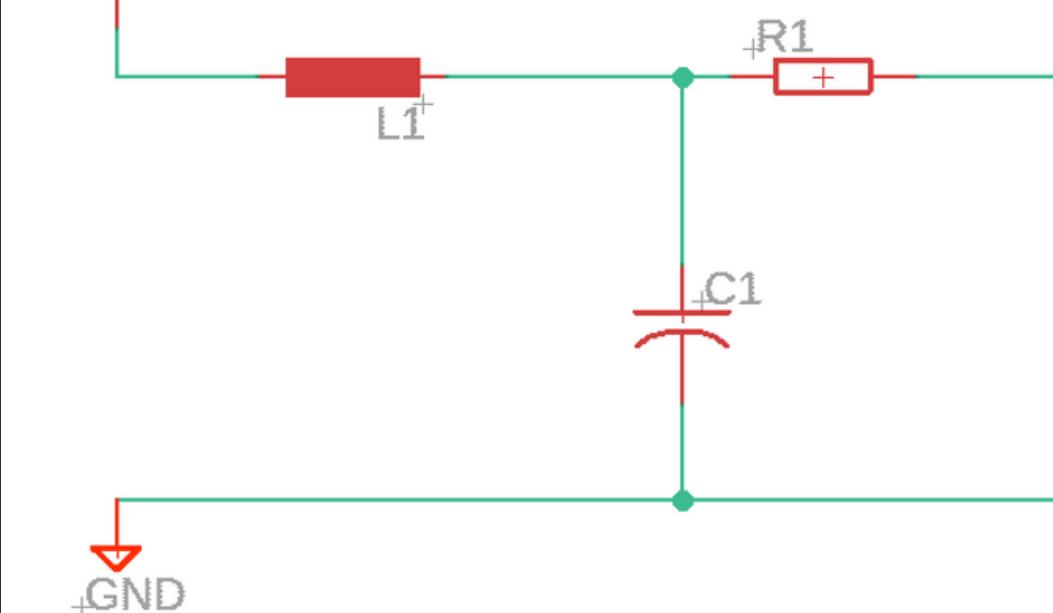
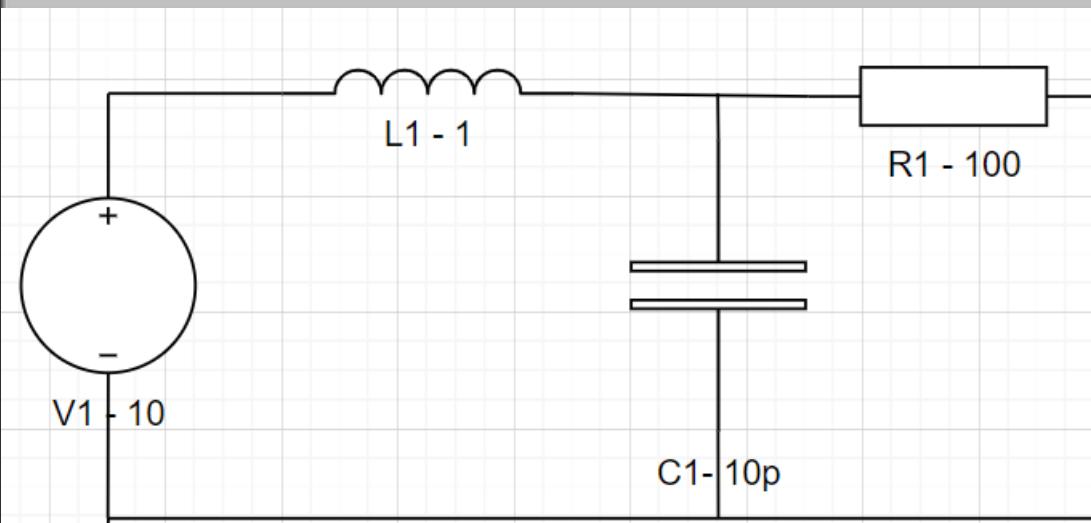
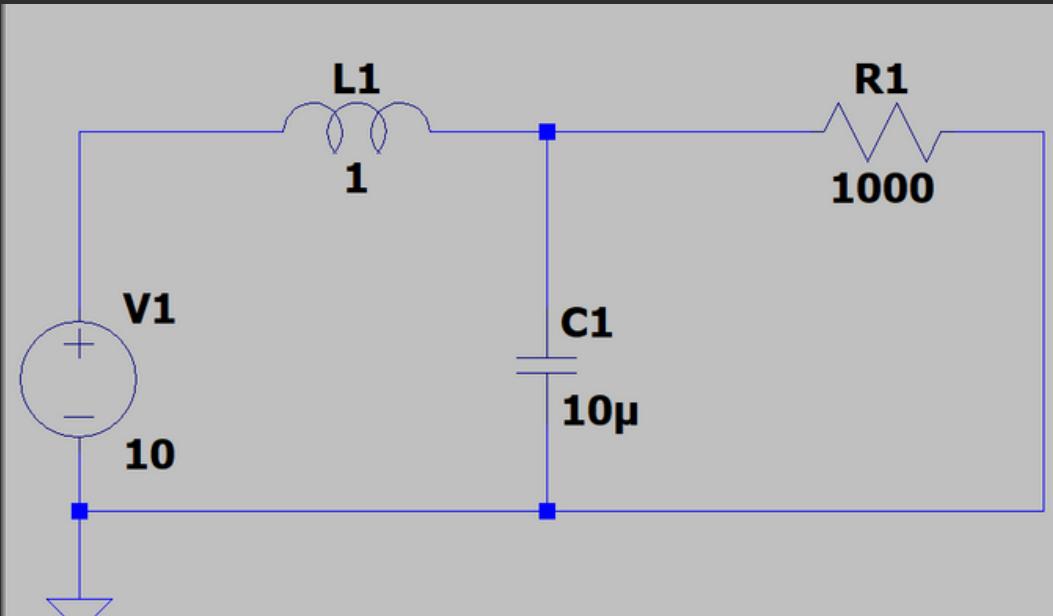
I circuiti elettronici riportati nelle documentazioni dei progetti devono seguire delle regole ben precise imposte dai clienti.

La rappresentazione circuitale può essere ricavata dai diversi software e tool usati in azienda (LTSpice, Eagle e Draw.io) che seguono convenzioni diverse.

Per seguire uno standard traduciamo manualmente i circuiti in latex (CircuiTikz).

Questo processo:

1. è dispendioso in termini di tempo (e soldi!);
2. può portare ad errori nella trascrizione;
3. il cambiamento di una convenzione richiede la modifica manuale della rappresentazione





Value Proposition

Il nostro prodotto offre diversi vantaggi:

01

Una semplice interfaccia grafica utilizzata da tutti i membri dell'azienda per la generazione di circuiti da inserire nella documentazione.

02

Automatizzazione del processo di trascrizione dei circuiti con conseguente diminuzione di tempo impiegato e di errori.

03

Possibilità di rendere il software open source in modo che la community possa migliorarlo e che sia più facile seguire i cambiamenti di regole per soddisfare le esigenze dei singoli clienti.

Ecco la soluzione:

LTSpice2CircuiTikz

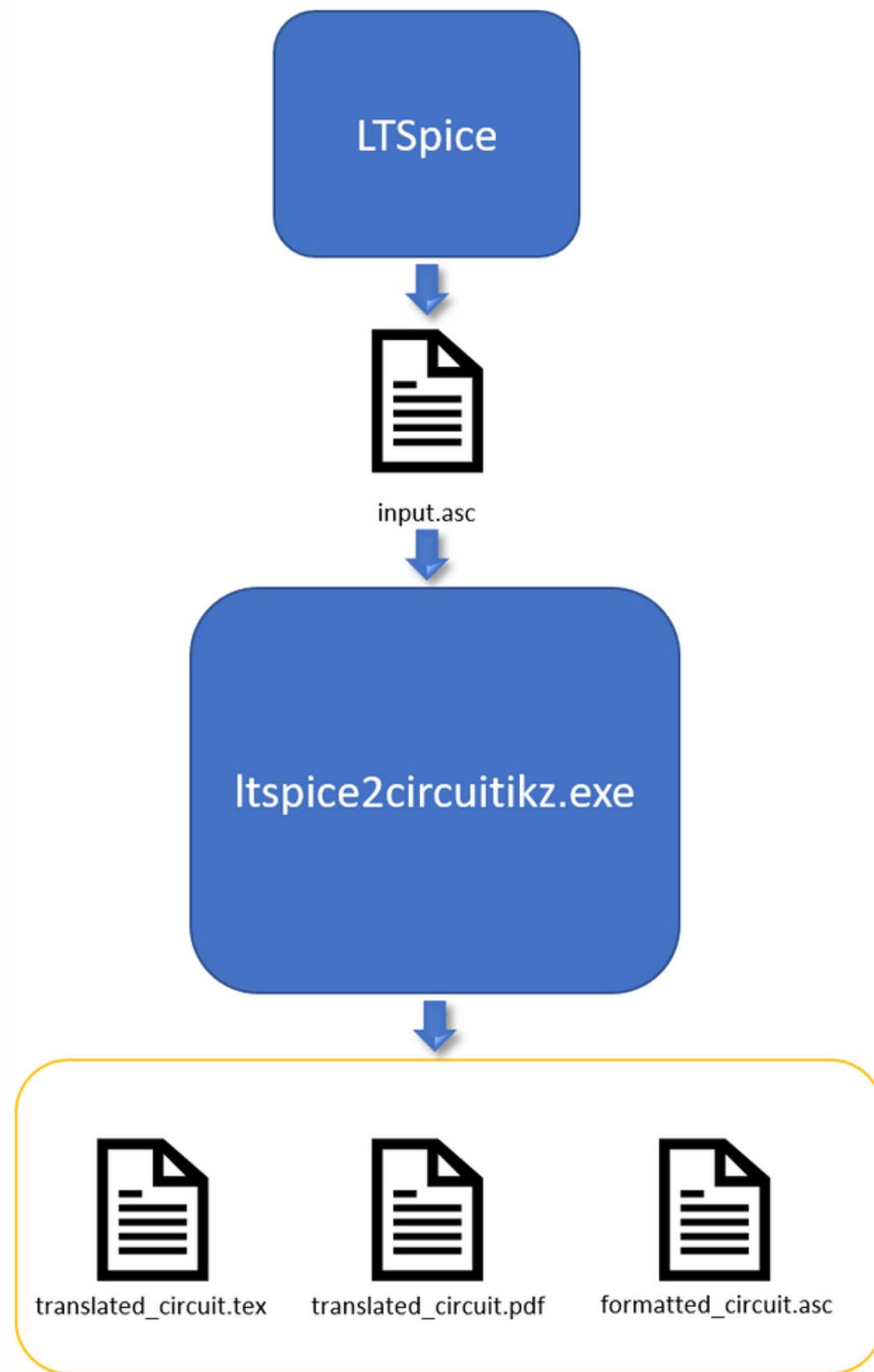


The image shows three screenshots of the LTSpice2CircuiTikz software interface, illustrating the process of generating TikZ code from LTSpice circuit simulations.

- Top Screenshot:** Shows the software's main window with tabs for "ASC File" and "Latex Output". The "Latex Output" tab displays the generated TikZ code for a circuit diagram.
- Middle Screenshot:** Shows the software's main window with tabs for "ASC File" and "Latex Output". The "Latex Output" tab displays the generated TikZ code for a circuit diagram, with a green message indicating "Latex output produced correctly!"
- Bottom Screenshot:** Shows the software's main window with tabs for "ASC File" and "Latex Output". The "Latex Output" tab displays the generated TikZ code for a circuit diagram, with a green message indicating "Latex output produced correctly!"

The generated TikZ code includes commands for document class, package imports, and circuit drawing, such as \documentclass{article}, \usepackage{circuitikz}, \usepackage[4paper, total={6in, 8in}]{geometry}, \begin{document}, \begin{center}, \begin{circuitikz}, \draw, \tikzshorten, \tikzshort, and \tikzbox.

Funzionamento



Utilizzare l'applicazione è semplicissimo!

- 01 Create un circuito con LTSpice
- 02 Lanciarare l'applicazione ltspice2circuitikz
- 03 Selezionare il file .asc generato da LTSpice
- 04 Cliccare il pulsante Run sull'applicazione
- 05 L'applicazione genera il testo latex da inserire nella documentazione

Tecnologie utilizzate

ANTLRWorks 1.5.2

Tool per la generazione del parser e del lexer in Java

MikTek (pdflatex)

Compilatore Latex per la generazione di file pdf a partire da un file .tex

Java e antlr-complete

Libreria con la definizione delle classi necessarie all'utilizzo del parser e lexer generato

Eclipse IDE e Github

IDE per la creazione del progetto Java e versioning del codice

CircuiTikz

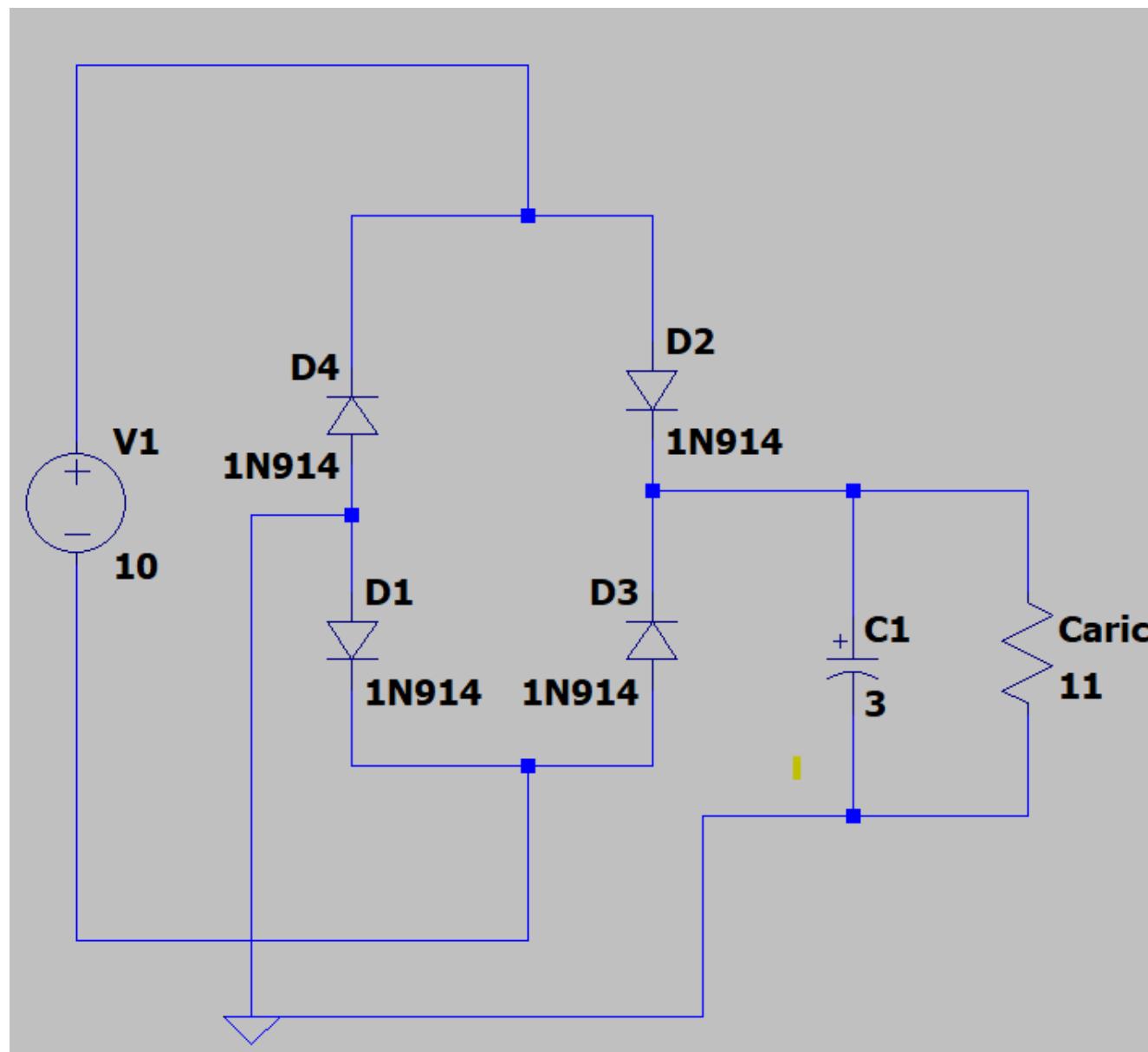
Pacchetto Latex per la rappresentazione di circuiti elettrici

Framework Qt

Creazione dell'app client

CONFRONTO RAPPRESENTAZIONI

01



02

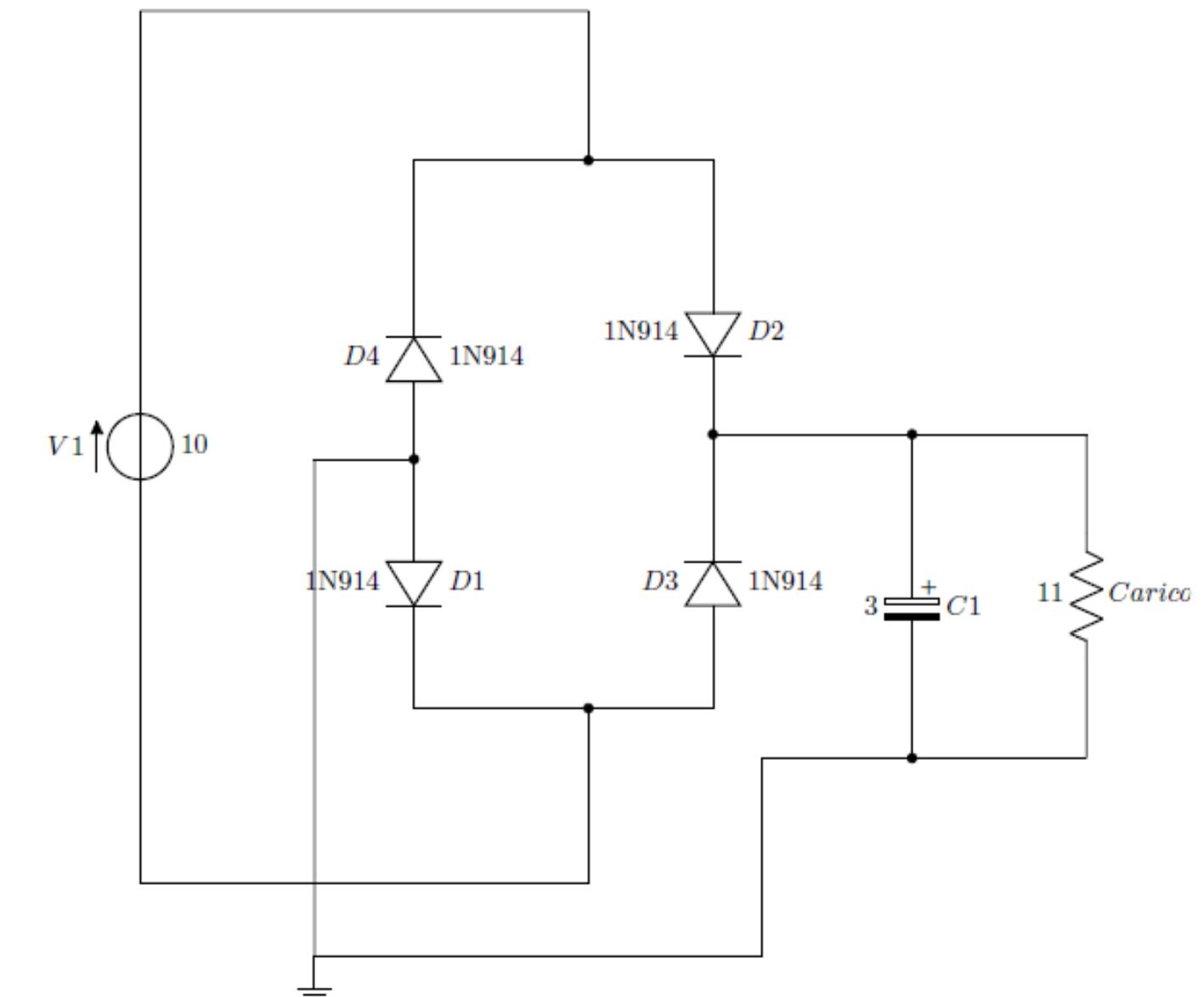
```
Version 4
SHEET 1 880 680
WIRE -32 -304 -320 -304
WIRE -32 -208 -32 -304
WIRE -32 -208 -144 -208
WIRE 48 -208 -32 -208
WIRE 48 -128 48 -208
WIRE -144 -112 -144 -208
WIRE -320 -64 -320 -304
WIRE 48 -32 48 -64
WIRE 176 -32 48 -32
WIRE 288 -32 176 -32
WIRE -144 -16 -144 -48
WIRE -144 -16 -208 -16
WIRE -144 32 -144 -16
WIRE 48 32 48 -32
WIRE 288 32 288 -32
WIRE 176 48 176 -32
WIRE -144 144 -144 96
WIRE -32 144 -144 144
WIRE 48 144 48 96
WIRE 48 144 -32 144
WIRE 176 176 176 112
WIRE 176 176 80 176
WIRE 288 176 288 112
WIRE 288 176 176 176
WIRE -320 256 -320 16
WIRE -32 256 -32 144
WIRE -32 256 -320 256
WIRE -208 304 -208 -16
WIRE 80 304 80 176
WIRE 80 304 -208 304
FLAG -208 304 0
SYMBOL diode -160 32 R0
SYMATTR InstName D1
SYMATTR Value 1N914
SYMBOL diode 32 -128 R0
SYMATTR InstName D2
SYMATTR Value 1N914
SYMBOL diode 64 96 R180
WINDOW 0 24 64 Left 2
WINDOW 3 24 0 Left 2
SYMATTR InstName D3
SYMATTR Value 1N914
SYMBOL diode -128 -48 R180
WINDOW 0 24 64 Left 2
WINDOW 3 24 0 Left 2
SYMATTR InstName D4
SYMATTR Value 1N914
SYMBOL voltage -320 -80 R0
WINDOW 123 0 0 Left 0
WINDOW 39 24 44 Left 2
SYMATTR InstName V1
SYMATTR Value 10
SYMBOL res 272 16 R0
SYMATTR InstName Carico
SYMATTR Value 11
SYMATTR SpiceLine tol=1 pwr=0.1
SYMBOL polcap 160 48 R0
WINDOW 3 24 56 Left 2
SYMATTR Value 3
SYMATTR InstName C1
SYMATTR Description Capacitor
SYMATTR Type cap
SYMATTR SpiceLine V=2.7 Irms=0 Rser=0.0453121 Lser=25n mfg="Würth Elektronik" pn="850617030001 WCAP-STSC 8x20" type="Sup
```

CONFRONTO RAPPRESENTAZIONI

03

```
\documentclass{article}
\usepackage[circuitikz]
\usepackage[a4paper, total={6in, 8in}]{geometry}
\begin{document}
\begin{center}
\begin{circuitikz}
\draw (3.52,5.44) to[Do=$D1$, a={1N914}] (3.52,4.16);
\draw (7.36,8.64) to[Do=$D2$, a={1N914}] (7.36,7.36);
\draw (7.36,4.16) to[Do=$D3$, a={1N914}] (7.36,5.44);
\draw (3.52,7.04) to[Do=$D4$, a={1N914}] (3.52,8.32);
\draw (0.00,5.76) to[vsource=$V1$, a={10}] (0.00,7.36);
\draw (12.16,5.44) to[R=$Carico$, a={11}] (12.16,3.84);
\draw (9.92,5.12) to[eC=$C1$, a={3}] (9.92,3.84);
\draw (5.76,12.16) to[short, l={$\cdot$}] (0.00,12.16);
\draw (5.76,10.24) to[short, l={$\cdot$}] (5.76,12.16);
\draw (5.76,10.24) to[short, l={$\cdot$}] (3.52,10.24);
\draw (7.36,10.24) to[short, l={$\cdot$}] (5.76,10.24);
\draw (7.36,8.64) to[short, l={$\cdot$}] (7.36,10.24);
\draw (3.52,8.32) to[short, l={$\cdot$}] (3.52,10.24);
\draw (0.00,7.36) to[short, l={$\cdot$}] (0.00,12.16);
\draw (7.36,6.72) to[short, l={$\cdot$}] (7.36,7.36);
\draw (9.92,6.72) to[short, l={$\cdot$}] (7.36,6.72);
\draw (12.16,6.72) to[short, l={$\cdot$}] (9.92,6.72);
\draw (3.52,6.40) to[short, l={$\cdot$}] (3.52,7.04);
\draw (3.52,6.40) to[short, l={$\cdot$}] (2.24,6.40);
\draw (3.52,5.44) to[short, l={$\cdot$}] (3.52,6.40);
\draw (7.36,5.44) to[short, l={$\cdot$}] (7.36,6.72);
\draw (12.16,5.44) to[short, l={$\cdot$}] (12.16,6.72);
\draw (9.92,5.12) to[short, l={$\cdot$}] (9.92,6.72);
\draw (3.52,3.20) to[short, l={$\cdot$}] (3.52,4.16);
\draw (5.76,3.20) to[short, l={$\cdot$}] (3.52,3.20);
\draw (7.36,3.20) to[short, l={$\cdot$}] (7.36,4.16);
\draw (7.36,3.20) to[short, l={$\cdot$}] (5.76,3.20);
\draw (9.92,2.56) to[short, l={$\cdot$}] (9.92,3.84);
\draw (9.92,2.56) to[short, l={$\cdot$}] (8.00,2.56);
\draw (12.16,2.56) to[short, l={$\cdot$}] (12.16,3.84);
\draw (12.16,2.56) to[short, l={$\cdot$}] (9.92,2.56);
\draw (0.00,0.96) to[short, l={$\cdot$}] (0.00,5.76);
\draw (5.76,0.96) to[short, l={$\cdot$}] (5.76,3.20);
\draw (5.76,0.96) to[short, l={$\cdot$}] (0.00,0.96);
\draw (2.24,0.00) to[short, l={$\cdot$}] (2.24,6.40);
\draw (8.00,0.00) to[short, l={$\cdot$}] (8.00,2.56);
\draw (8.00,0.00) to[short, l={$\cdot$}] (2.24,0.00);
\draw (2.240000,0.000000) to (2.240000,0.000000) node[ground]();
\draw (9.92,2.56) to[short, -*] (9.92,2.56);
\draw (5.76,10.24) to[short, -*] (5.76,10.24);
\draw (7.36,6.72) to[short, -*] (7.36,6.72);
\draw (9.92,6.72) to[short, -*] (9.92,6.72);
\draw (5.76,3.2) to[short, -*] (5.76,3.2);
\draw (3.52,6.4) to[short, -*] (3.52,6.4);
\end{circuitikz}
\end{center}
\end{document}
```

04



Stima guadagno in ore

Manualmente:

4 ore/progetto

Tempo necessario per la trascrizione
dei circuiti per un progetto,
considerando errori e manutenzione.

Automaticamente:

10 min/progetto

Tempo necessario a lanciare
l'applicazione più eventuale
manutenzione.

Costi per lo sviluppo

Persone

2

Ore/persona

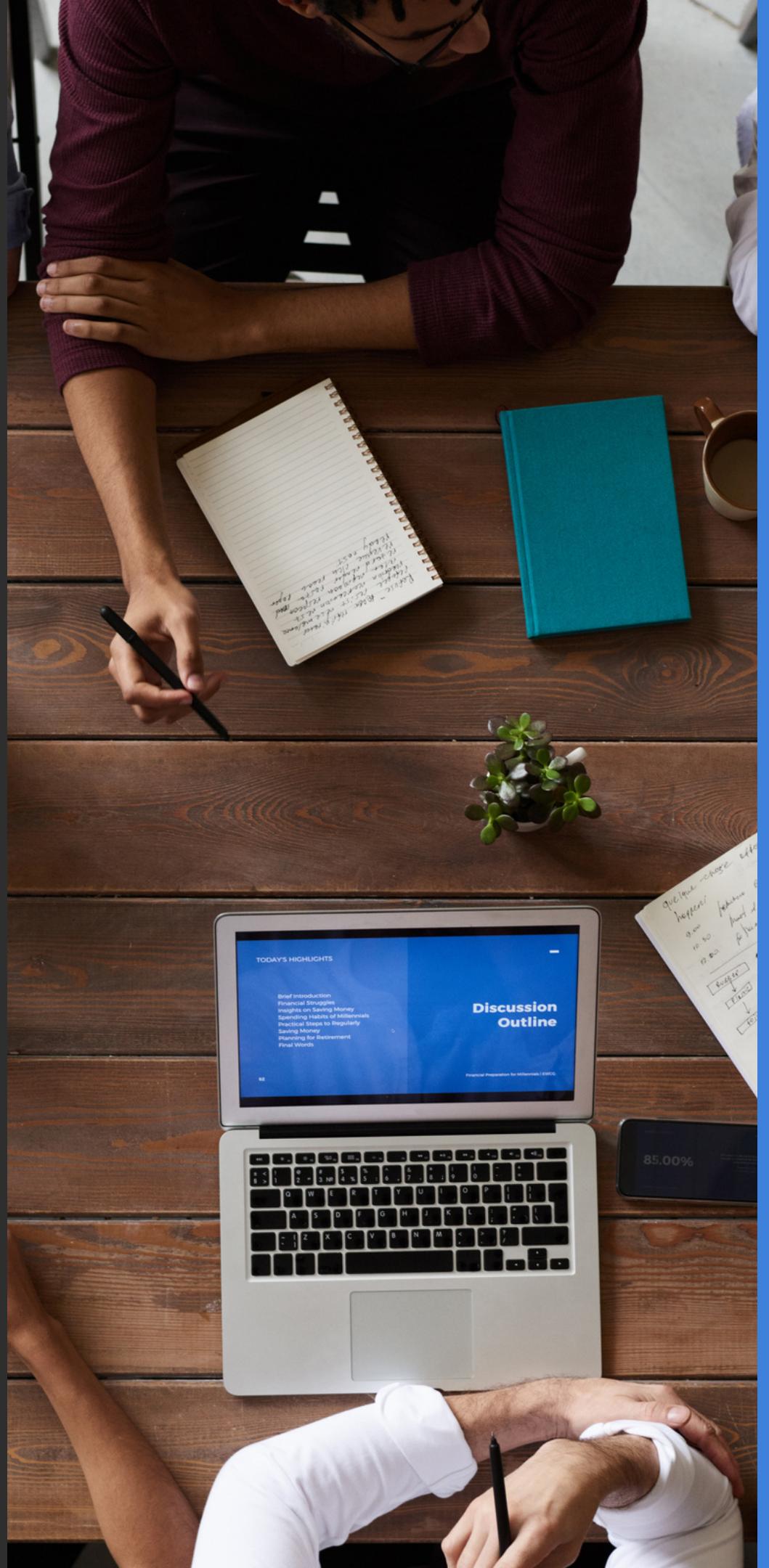
80

Costo all'ora

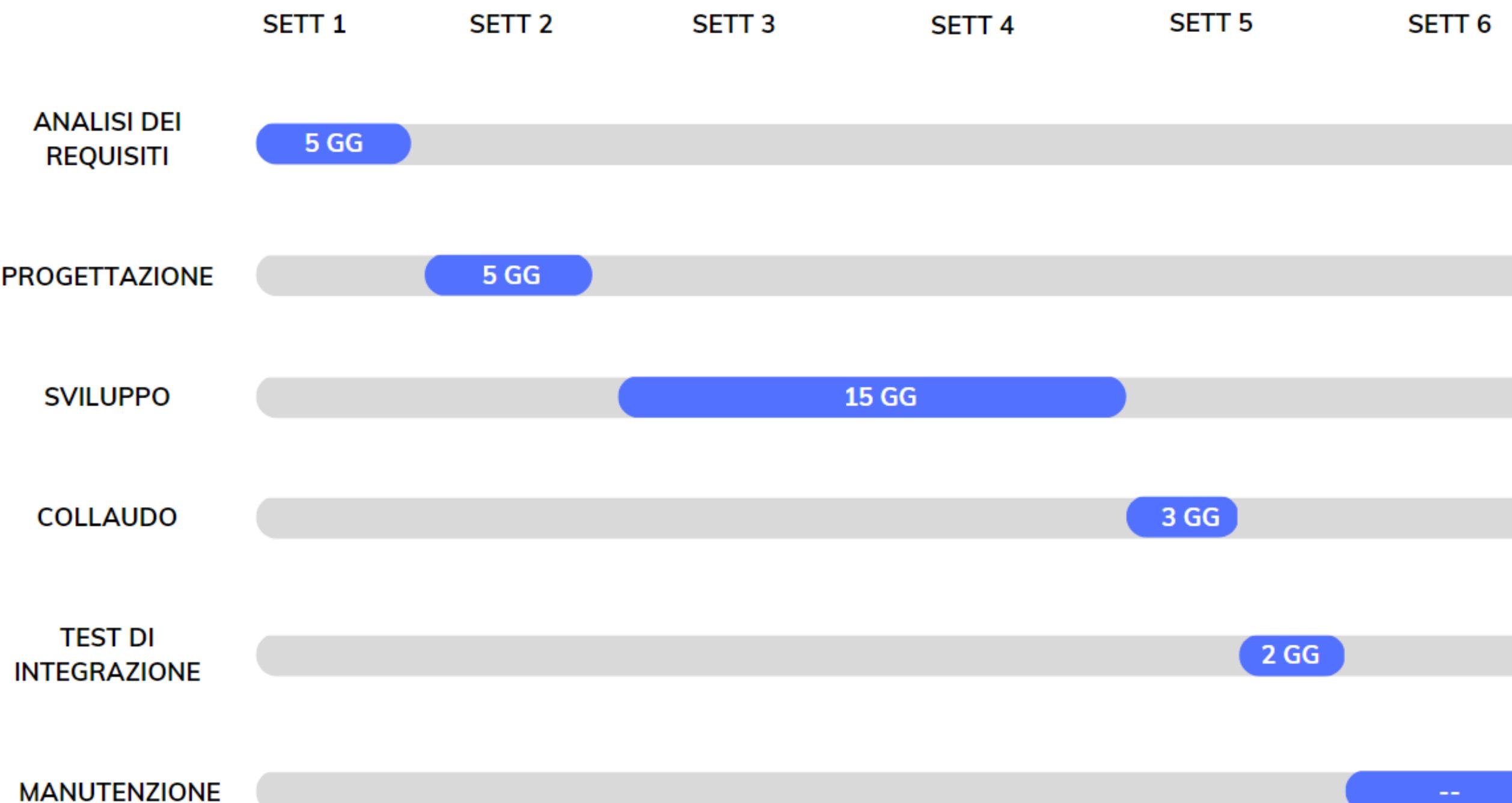
€45

Costo Finale

~€7200



GANTT CHART



Per aumentare il guadagno dallo sviluppo di tale applicazione, si può valutare anche la commercializzazione del prodotto.

Sul mercato non sono presenti soluzioni concorrenti.

Si può utilizzare un modello ad abbonamento mensile:

9.99 euro/mese



Management Team



Stella Ornelas
Software Engineer



Bryan Cook
Developer

Grazie per l'attenzione!